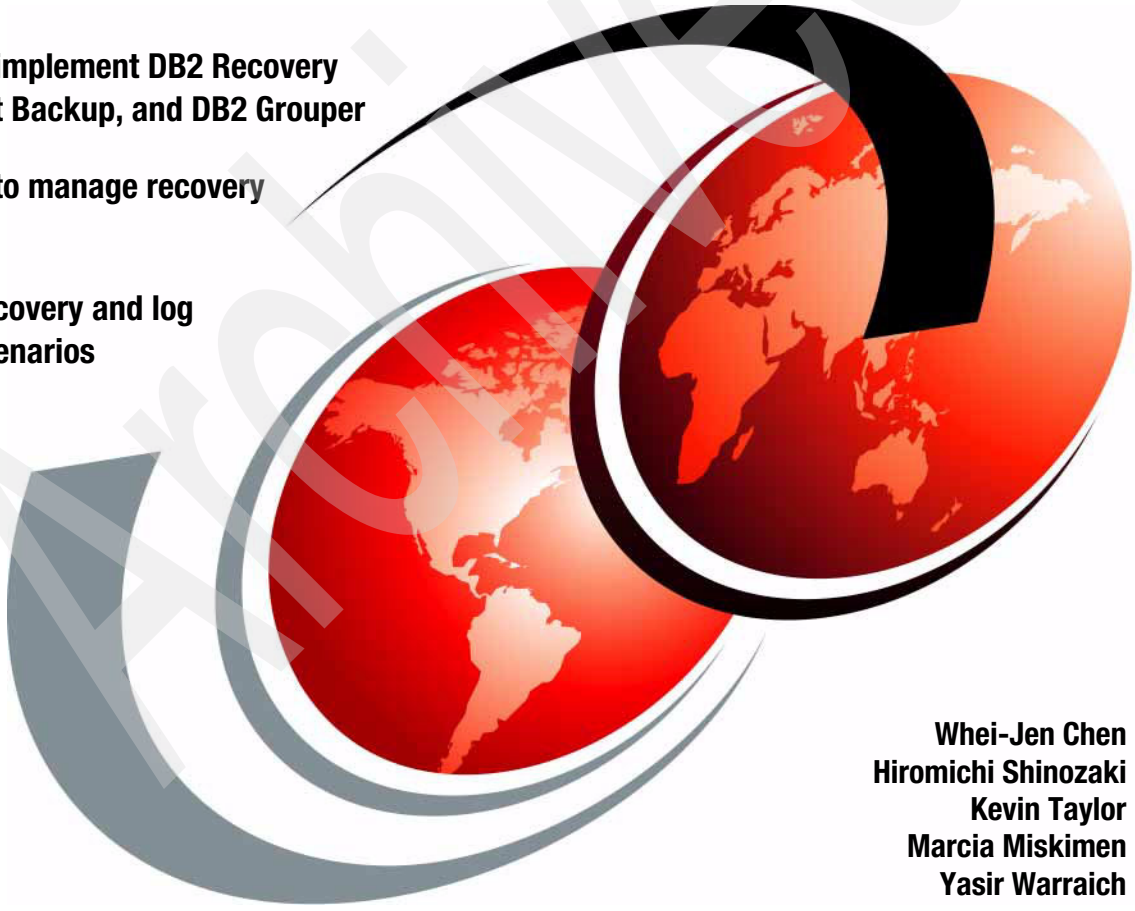
DB2 Information Management Software

IBM

IBM DB2 Recovery Expert for Multiplatforms V2 Usage Guide

- Install and implement DB2 Recovery Expert, Fast Backup, and DB2 Grouper
- Learn how to manage recovery assets
- Includes recovery and log analysis scenarios



Whei-Jen Chen
Hiromichi Shinozaki
Kevin Taylor
Marcia Miskimen
Yasir Warraich

Redbooks

ibm.com/redbooks

International Technical Support Organization

IBM DB2 Recovery Expert for Multiplatforms V2 Usage Guide

February 2006

SG24-7133-00

Note: Before using this information and the product it supports, read the information in “Notices” on page xi.

First Edition (February 2006)

This edition applies to IBM DB2 Recovery Expert V2, Fix Pack 1.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other 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 10504-1785 U.S.A.

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.

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.

This information contains 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 contains 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. 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 IBM's application programming interfaces.

Trademarks

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

℮server®	DRDA®	RACF®
℮server®	Enterprise Storage Server®	System Storage™
Redbooks (logo) ™	FlashCopy®	Tivoli®
pSeries®	IBM®	WebSphere®
AIX®	QMF™	
DB2®	Redbooks™	

The following terms are trademarks of other companies:

Java, Java Naming and Directory Interface, JDBC, JVM, Solaris, Sun, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Preface

This redbook describes how to recover database objects safely, precisely, and quickly by using DB2® Recovery Expert for Multiplatforms. It includes the following chapters:

- **Chapter 1, "Introduction to DB2 Recovery Expert V2"**

  Introduces DB2 Recovery Expert V2 for Multiplatforms and provides a high-level discussion of its capabilities.

- **Chapter 2, "Planning for recovery"**

  Introduces DB2 Recovery Expert V2 for Multiplatforms and provides a high level discussion of its capabilities.

- **Chapter 3, "Installing and configuring DB2 Recovery Expert"**

  Contains information about installing DB2 Recovery Expert and describes the configuration options for DB2 Recover Expert. It also includes tips on troubleshooting.

- **Chapter 4, "Creating and managing recovery assets"**

  Provides details of DB2 Recovery Expert assets and how to maintain these assets.

- **Chapter 5, "Log analysis"**

  Discusses the Log Analysis concepts, Log Analysis usage scenarios, Remote Log Analysis, and considerations when using Log Analysis.

- **Chapter 6, "Backing out insert, update, and delete operations"**

  Describes how to use DB2 Recovery Expert to back out the insert, update, and delete operations on table, table space, and database to a point-in-time. It also shows the details of using DB2 Recover Expert to perform redirected restore and recovering related objects using DB2 Recover Expert and Grouper.

- **Chapter 7, "Recovering dropped objects"**

  Describes the recovery procedures for dropped objects including table, table space, and database.

- **Chapter 8, "Cloning a database, table space, or table"**

  Describes how to clone a database and clone tables or table spaces from within a database or to another database using Object Translation command.

- **Chapter 9, “Fast Backup”**

  Introduces DB2 Recovery Expert Fast Backup concept and describes how to prepare and use DB2 Recover Expert Fast Backup features and functions.

- **Chapter 10, “Installing and using the DB2 Grouper component”**

  Contains information on installing the DB2 Grouper component on multiplatform environment, and using Grouper to create sets to be used later by Recovery Expert.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), San Jose Center.

Left to right: Hiromichi, Kevin, Whei-Jen, Marcia, and Yasir

Whei-Jen Chen is a Project Leader at the ITSO, San Jose Center. She has extensive experience in application development, database design and modeling, and DB2 system administration. Whei-Jen is an IBM® Certified Solutions Expert in Database Administration and Application Development as well as an IBM Certified IT Specialist.

Hiromichi Shinozaki is a development engineer working in Recovery Expert Fast Backup development in Yamato Software Lab, IBM Japan. Hiromichi has three years of experience in finance solution software development and 10 years of experience in product planning of IBM PC and device driver and microcode development of several PC option cards.

Kevin Taylor is a Project Manager in the Product Introduction Centre in Hursley, UK, where he runs beta programs for DB2 products on Linux®, UNIX®, and Windows®. He has 21 years of experience in IT in IBM, including 15 years working with DB2, specifically in the banking and insurance sectors. He is an IBM Certified IT Specialist, an IBM Certified DB2 UDB V8.1 Database Administrator for Linux UNIX and Windows, and an IBM Certified Solution Designer in DB2 Business Intelligence V8. He has a Bachelor of Science degree in Management Science from Manchester University.

Marcia Miskimen is a Software Engineer at IBM Software Group's Silicon Valley Lab, where she is currently a Data Management Tools Specialist. She has over 20 years of experience in the IT industry, including 10 years as an IT Specialist in IBM Global Services. Her areas of expertise and interest include the application development life cycle, software testing, and tools of all sorts. She has a Bachelor of Science in Business Administration from The Ohio State University.

Yasir Warraich has a computer engineering degree from McGill University, Montreal. Currently, he is working as a Staff software engineer at IBM Toronto Software Lab. He has more than seven years of experience working in Advanced technical support team for DB2 on distributed platforms. His areas of expertise include data recovery and database performance.

Acknowledgement

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

John Bowen
Joel Farber
Teresa Leamon
Nathan Church
Don Weil
Guey-ing Chang
Khoa Pham
IBM Silicon Valley Laboratory

Ken Buck
Rocket Software

Takashi Saitoh
IBM Japan

Emma Jacobs
Octavian Lascu
Sangam Racherla
International Technical Support Organization, San Jose Center

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. QXXE Building 80-E2
  650 Harry Road
  San Jose, California 95120-6099

1

Introduction to DB2 Recovery Expert V2

This chapter introduces DB2 Recovery Expert V2 for Multiplatforms and provides a high-level discussion of its capabilities. It includes the following sections:

1.1 Overview of DB2 Recovery Expert

DB2 Recovery Expert (DB2 RE) for Multiplatforms provides an enhanced data recovery solution that enables more precise recovery operations while reducing disruption during the recovery process. DB2 RE creates and maintains additional recovery assets which are then used for intelligent analysis of both DB2 and DB2 Recover Expert assets to find the most efficient recovery path. DB2 RE facilitates the process of rebuilding your database assets such as tables, indexes, and data, to a specified point-in-time, often without taking the database or the business operations offline.

When you use standard DB2 UDB recovery utilities, you can restore databases and table spaces and roll forward to a point-in-time. However, you cannot recover only a single table easily, recover a set of related tables automatically, or recover even a specific row of data from one table. With DB2 RE, you can do this type of granular recovery. You might need this type of recovery if one of the following occurs:

- You ran the weekly batch job instead of the monthly job.
- You dropped a production table instead of a test table.
- You ran a delete with the wrong date parameter and ended up deleting too many rows.

DB2 RE for Multiplatforms is designed for the following environments:

- IBM AIX®
- Linux
- Microsoft® Windows
- Sun™ Solaris™

1.2 Key features and enhancements with DB2 Recovery Expert V2

This section describes the key features and enhancements with DB2 Recovery Expert V2.

1.2.1 Base features

In this section, we describe the base features of DB2 Recovery Expert V2. These base features are available in DB2 Recovery Expert V1. Enhancements are available through the DB2 Recovery Expert V1 fix packs. We describe the types of analysis and recovery that you can do using DB2 RE in this section. However, we cover how to do these tasks in the other chapters in this book.

Log analysis

You can use DB2 RE to generate reports that detail changes to tables or table spaces over time by analyzing the DB2 transaction logs. You can identify who made the changes, which is useful for certain types of auditing and compliance requirements. You can also use the log analysis function to undo or redo SQL statements selectively that have already run. You can also run the log analysis using logs retrieved from a remote system.

Point-in-time object recovery

DB2 RE allows the recovery of an object (a database, table space, or table) to a specified point-in-time. DB2 RE determines the most efficient method of recovering the specified object to the specified point-in-time automatically and allows you to select a specific method (if multiple methods are available). Point-in-time recovery is useful if you have made many changes to an object and you want to reverse these changes. If you know the specific point-in-time at which the object was correct, you can recover to that point.

Dropped object restore

Using DB2 RE, you can restore objects that have been dropped accidentally, such as table space, table, stored procedures, and so forth. Unlike the native DB2 method for restoring dropped tables (the `RECOVER DROPPED TABLES` option on the **`CREATE TABLESPACE`** command), the recovery method that DB2 RE uses does not require a full restore and roll forward. Instead, it uses another DB2 RE recovery feature called the *Versioning Repository*, which contains information about your database objects. If you need to recover from a dropped object, DB2 RE restores from the Versioning Repository, then does a selective roll forward or redo of transactions that relate specifically to the object in question. Additionally,

DB2 RE recovers not only the table but also recovers the related objects and authorizations on the dropped object.

Minilogs

DB2 RE provides specialty utilities that allow you to create subsets of transaction logs for specific tables. These subsets are called *minilogs*. Minilogs can be useful if you need to recover a table that is updated infrequently but exists within the same table space as more volatile tables.

Object translation

DB2 RE provides a facility to copy or move tables between table spaces. DB2 RE uses this feature during its recovery process to extract data from a backup or to recover a table space. However, you can also use this feature as a stand-alone program to clone a database or table space.

1.2.2 New features in DB2 Recovery Expert V2

DB2 Recovery Expert V2 has the following new features:

- Direct exploitation of the hardware copying features

  DB2 RE's Fast Backup feature now allows DB2 RE to manage the backup process as well as the recovery process. Fast backup is available for specific types of disk hardware. Flash Copy and Snapshot allow nondisruptive backups in a fraction of the time of conventional backups. DB2 RE integrates these hardware copies into recoveries as necessary. Hardware from the major providers of SAN and NAS is included. This combined support extends the 24x7 characteristics of DB2 system availability.

- DB2 Grouper component

  The DB2 Grouper component provides DB2 RE with the advanced ability to exploit dependent relationships in recovery scenarios. This ability ensures that DB2-enforced referential integrity, application-enforced referential integrity, and other key relationships are presented accurately when you are recovering a database object.

1.3 Introduction to DB2 Recovery Expert assets

This section introduces the components of DB2 RE that make the new features of DB2 RE V2 possible.

Note: This book describes DB2 RE. It does not describe the DB2 recovery process in depth. We assume that you have a basic understanding of DB2 backup and recovery process and strategy.

In a good DB2 production environment, you should make regular backups of your database. You should also have database recovery logging enabled so that data activity is captured in the log. We call these backups and logs *recovery assets*. You use these assets in DB2 restore and roll forward activities.

DB2 RE introduces additional recovery assets that can aid in getting your data back to a healthy state. The following sections describe DB2 RE recovery assets.

1.3.1 Versioning Repository

The Versioning Repository is a collection of tables that store information that is captured from the DB2 system catalog. This information allows DB2 RE to recreate object and authorization DDL. The Versioning Repository includes DB2 system catalog information and DB2 RE object recovery information. Only DB2 system catalog information that is required to recreate object DDL is maintained in the Versioning Repository.

Important: The Versioning Repository is not a copy or backup of the DB2 system catalog. It is used by and is relevant to DB2 RE only.

Important: You should *never* use DB2 RE to attempt a recovery on any of the DB2 system catalog tables or table space.

For more information about using the Versioning Repository, see 4.2, “Versioning Repository” on page 56.

1.3.2 Minilogs

Minilogs are files that contain extracted portions of the DB2 log that pertain to database tables that you identify as being especially important. If minilogs are created during recovery, DB2 RE presents an additional recovery path that uses minilogs. Minilogs shift the I/O time that is spent in reading vast portions of the log that do not contain relevant data to non-critical times. For more information about minilogs, see 4.3, “Minilogs” on page 70.

1.3.3 Log analysis

You can use log analysis to retrieve information about database activity from the DB2 logs. Specifically, you can extract the INSERT, UPDATE, and DELETE operations and save information about those transactions. The data for the report is retrieved from the active log files and the archive log files. For more information about log analysis, see Chapter 5, “Log analysis” on page 83.

1.3.4 Object translation

When you need to recover a dropped table, DB2 RE can extract the data from a DB2 backup. You can also clone a table space or table to another location using object translation which will keep track of the internal DB2 object IDs so the resulting objects match the original. DB2 RE calls this function Object Translation. DB2 RE also uses this function internally during certain types of object recovery. For more information about object translation, see Chapter 8, “Cloning a database, table space, or table” on page 215.

1.3.5 Fast backup

DB2 RE introduces a new feature called *Fast Backup* to create backups of your databases and table spaces using hardware technology such as Shark Flash Copy. DB2 RE can use the backups that Fast Backup creates to recover your data. Backups that Fast Backup creates are separate and independent from DB2 backups. For more information about Fast Backup, see Chapter 9, “Fast Backup” on page 247.

1.3.6 DB2 Grouper component

The DB2 Grouper component is a component that is common in several DB2 Tools, including DB2 RE. DB2 Grouper collects information about DB2-enforced referential integrity and application enforced relationships. DB2 Grouper defines these relationships through an automatic discovery process, or you can define them manually. The relationship information is stored in *group sets* that you use in DB2 RE object recovery. For more information about how to use DB2 Grouper

group sets, see Chapter 10, “Installing and using the DB2 Grouper component” on page 305.

1.4 Introduction to the recovery path concept

DB2 RE used whatever recovery assets it finds available to determine the most efficient recovery path for a particular recovery scenario. Figure 1-1 illustrates the recovery path concept. If certain assets are not available or do not provide a valid recovery, they are not used. DB2 RE might also find several valid recovery paths, which it offers to you as choices. It is important to understand the creation and use of the different DB2 RE assets so that you have the best chances of recovering your data.

Because intervening events might occur in the database history, it is not always possible to recover a database object to some arbitrary time. Thus, DB2 RE might not be able to generate a valid recovery path. For example, it is not currently possible to recover an object by working forward or backward across a DB2 load operation.

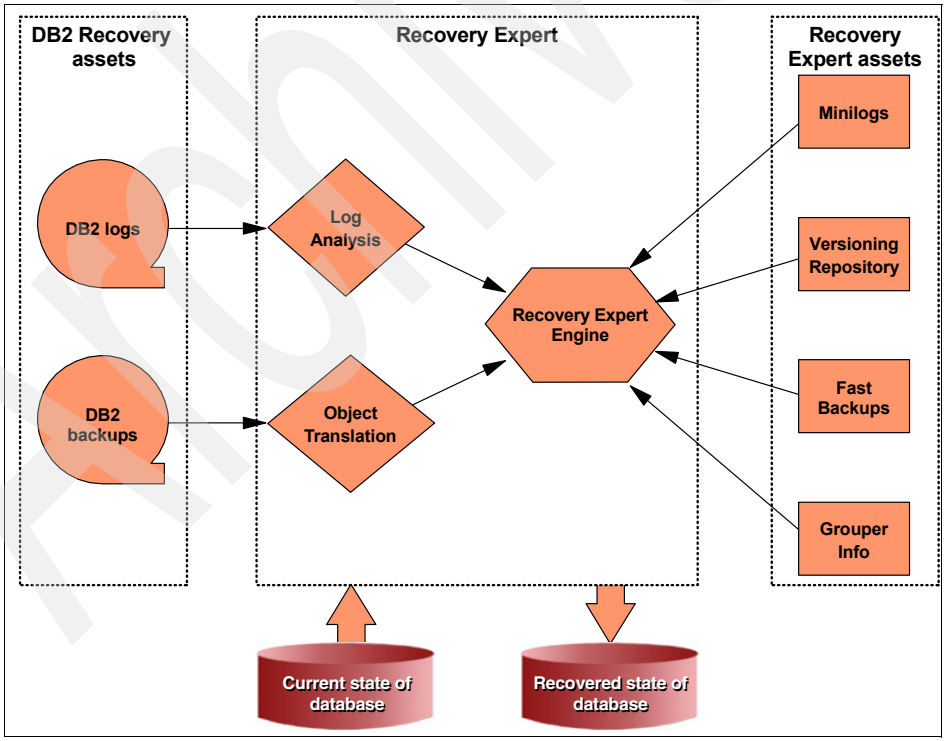


Figure 1-1 Recovery path concept

For more information about creating and using recovery assets, see Chapter 4, “Creating and managing recovery assets” on page 53. Although we discuss the recovery path and its concepts throughout this book, you can find more detailed information in Chapter 6, “Backing out insert, update, and delete operations” on page 123 and Chapter 7, “Recovering dropped objects” on page 187.

2

Planning for recovery

You can use this chapter as a guide to ensure that you have the required prerequisite assets available should you need to recover data using DB2 Recovery Expert. It discusses each of the recovery areas and lists the assets that are required for DB2 Recovery Expert to be able to recover data in these scenarios.

Note: This chapter lists the most common set of required assets for these scenarios. There might be cases that do not require all the assets.

This chapter includes the following sections:

2.1 Introduction to database recovery planning

In the scenarios in this chapter, we consider each type of object from which you might need to recover as well as the situation from which you are recovering. We then list the minimum assets that are required to recover.

You should include the following assets in your planning:

- DB2 backups
- DB2 log records
- Versioning Repository
- Archive Log Inventory Data
- Minilogs
- DB2 Grouper
- Fast Backup

You find more information about how to create and to manage these assets in Chapter 4, “Creating and managing recovery assets” on page 53.

Including DB2 Recovery Expert (DB2 RE) as an addition to your recovery strategy does not mean that you should change the frequency of backups. It is strongly recommended that you perform full, offline backups as part of the backup schedule.

Important: There are circumstances where DB2 RE is able to use online backups as well as incremental and delta backups, specifically when DB2 RE calls the DB2 **RESTORE** command. However, you cannot predict whether these backup will be sufficient for DB2 RE to use. Therefore, the most pragmatic approach is to perform regular, full, offline database backups.

Important: For DB2 RE to function correctly, your database should be enabled for archiving logging by setting the database configuration parameters `LOGRETAIN` or `LOGARCHMETH1`.

Tip: Although updating the Archive Log Inventory Data (ALID) is not always mandatory, if you keep it up-to-date, it will give the best choice of recovery paths.

2.2 Database recovery

This section lists the assets that are involved in database recovery.

2.2.1 Database point-in-time recovery

Table 2-1 lists the assets that are required for a database point-in-time recovery.

Table 2-1 Database point-in-time recovery assets

Recovery Asset	Comment
Database backup	Full offline database backup required. Backup as frequently as your current recovery strategy.
Database log records	Must be available if rolling forward beyond backup.
Fast Backup image	The database can be recovered using a volume restore or database object restore from a Fast Backup image.

2.2.2 Dropped database recovery

Table 2-2 lists the assets that are required for a dropped database recovery.

Table 2-2 Dropped database recovery assets

Recovery Asset	Comment
Database backup	Full offline database backup required. Backup as frequently as your current recovery strategy. Online backup is not suitable because DB2 RE does not allow roll forward in this situation.
Database log records	Must be available if rolling forward beyond backup.
Fast Backup image	If an object has been dropped, you *cannot* use Fast Backup image.

2.3 Table space recovery

This section lists the assets that are involved in table space recovery.

2.3.1 Table space point-in-time recovery

Table 2-3 lists the assets that could be required for a table space point-in-time recovery, depending on the assets that are available and the recovery path that is generated.

Table 2-3 Table space point-in-time recovery assets

Recovery Asset	Comment
Versioning Repository	Versioning Repository should have been run after object created. Versioning Repository must have been run after any DDL changes on tables in the table space. This includes authorizations.
Database or table space backup	Full offline or online database or table space backup. Take as frequently as your current recovery strategy dictates. Versioning Repository is up-to-date when the backup is taken.
Database log records	Must be available if rolling forward beyond backup, or from online backup.
ALID	An up to date ALID will save time during recovery. May also give extra recovery path options. If not current it will be updated automatically during log analysis phase of recovery.
Fast Backup image	Large table spaces can be recovered at the volume or object level.
Minilogs	Optional. You can create minilogs for low update tables if they save whole log records from being read. You can take minilogs for every table in the table space.
DB2 Grouper sets	Use DB2 Grouper to check whether other objects are involved in the recovery.

2.3.2 Dropped table space recovery

DB2 RE does not recover a dropped table space that has tables with indexes or LOB/Long data in a separate table space. Table 2-4 lists the assets that are required for a dropped table space recovery.

Table 2-4 Dropped table space recovery assets

Recovery Asset	Comment
Versioning Repository	Versioning Repository must have been run after object created *or* if any DDL changes on tables in the table space, including authorizations.
Database or table space backup	Full offline database or table space backup. Take as frequently as your current recovery strategy dictates. Versioning Repository is up-to-date when the backup is taken.
Database log records	Must be available if rolling forward beyond backup.
ALID	An up-to-date ALID saves time during recovery and might also provide extra recovery path options. If not current, it is updated automatically during the log analysis phase of recovery.
Fast Backup image	You *cannot* use the Fast Backup image if the table space has been dropped.
Minilogs	Optional You can create minilogs for low update tables if they save whole log records from being read.
DB2 Grouper sets	Use DB2 Grouper to check whether other objects are involved in the recovery.

2.4 Table level recovery

This section lists the assets that are involved in table recovery.

2.4.1 Table point-in-time recovery

Table 2-5 lists the assets that are required for a table point-in-time recovery.

Table 2-5 Table point-in-time recovery assets

Recovery Asset	Comment
Versioning Repository	You must run Versioning Repository after the object created *or* if there are any DDL changes on tables, including authorizations.
Database or table space backup	Full offline database or table space backup. Take as frequently as your current recovery strategy dictates. This is the starting point for point-in-time Redo/Undo SQL. Versioning Repository is up-to-date when the backup is taken.
Database log records	Must be available if rolling forward beyond backup. Used by Log Analysis to generate Undo/Redo SQL.
ALID	An up-to-date ALID saves time during recovery and might also give extra recovery path options. If not current, it is updated automatically during the log analysis phase of recovery.
Minilogs	Optional You can create minilogs for low update tables if they save whole log records from being read.
DB2 Grouper sets	Use DB2 Grouper to check whether other objects are involved in the recovery.

2.4.2 Dropped table recovery

Table 2-6 lists the assets that are required for a dropped table recovery.

Table 2-6 Dropped table recovery assets

Recovery Asset	Comment
Versioning Repository	You must run Versioning Repository after the object is created *or* if there are any DDL changes on tables, including authorizations.
Database or table space backup	Full offline database or table space backup. Take as frequently as your current recovery strategy dictates. Versioning Repository is up-to-date when the backup is taken.
Database log records	Must be available if rolling forward beyond backup. Used by Log Analysis to generate Undo/Redo SQL
ALID	An up-to-date ALID saves time during recovery and might also give extra recovery path options. If not current, it is updated automatically during log analysis phase of recovery.
Minilogs	Optional You can create minilogs for low update tables if they save whole log records from being read.
DB2 Grouper sets	Use DB2 Grouper to check whether other objects are involved in the recovery.

2.5 Row-level recovery

Row-level recovery is the most granular level of recovery and is considered as application or logical recovery. At the minimum level, row-level recovery involves running log analysis to determine the Undo/Redo SQL. However, if many rows are involved, it might be expedient to go back to a backup and re-run the SQL, which would involve the assets requirements that are listed in Table 2-5 on page 14.

Table 2-7 lists the minimum requirements, assuming that no DDL has been run against the table since its creation.

Table 2-7 Row -level recovery assets

Recovery Asset	Comment
Database or table space backup	Full offline or online database or table space backup. Take as frequently as your current recovery strategy dictates.
Database log records	Used by log analysis to generate Undo/Redo SQL.
Minilogs	Optional You can create minilogs for low update tables if they save whole log records from being read.
DB2 Grouper sets	Use DB2 Grouper to check whether other objects are involved in the recovery.

2.6 Other object recovery

This section lists the assets involved in recovering the following objects:

- Buffer pool
- Event monitor
- Partition group
- Schema
- Stored procedure
- User-defined type

Table 2-8 lists the assets that are required for other object recovery.

Table 2-8 Other objects recovery assets

Recovery Asset	Comment
Database or table space backup	Full offline database or table space backup. Take as frequently as your current recovery strategy, or update to date Versioning Repository.
Versioning Recovery	You should run Versioning Repository after the object is created. You must run Versioning Repository after any DDL changes on tables in the table space, including authorizations.
Database log records	Must be available if rolling forward beyond backup.

Recovery Asset	Comment
ALID	An up-to-date ALID saves time during recovery and might also give extra recovery path options. If not current, it is updated automatically during the log analysis phase of recovery.

2.7 Log mining

Log mining specifically refers to running DB2 RE log analysis, whether from the GUI or from the command line. You can run log analysis as required against DB2 log files. Therefore, the only asset upon which log mining is dependant is the DB2 log files.

If the logs have been archived to Tivoli® Storage Manager, or some other such library manager, then they need to be accessible physically to DB2 RE. If the logs were archived by DB2 using `USEREXIT` or a shared library that is defined in `LOGARCHMETH1` or `LOGARCHMETH2`, DB2 RE retrieves the archived logs automatically. If the logs were archived using a method outside of DB2, users must retrieve the log files manually.

DB2 RE works directly with the storage media server automatically. You can use a full online or offline backup as a starting point for reconstructing update SQL.

3

Installing and configuring DB2 Recovery Expert

This chapter contains information about installing DB2 Recovery Expert and describes the configuration options for DB2 Recovery Expert. It also includes tips on troubleshooting. This chapter discusses installation of the DB2 Recovery Expert main components only.

For information about how to install Fast Backup, see Chapter 9, “Fast Backup” on page 247. For information about how to install DB2 Grouper see Chapter 10, “Installing and using the DB2 Grouper component” on page 305.

This chapter includes the following sections:

3.1 Installation requirements for DB2 Recovery Expert

Support for specific levels of hardware and software can change with each release. Therefore, we do not describe these components in detail here in an effort to reduce possible confusion. For the most up-to-date hardware and software requirements, consult the *IBM DB2 Recovery Expert for Multiplatforms: User's Guide*, SC18-9564, the product readme file, and any documentation updates that are available on the DB2 Tools Publications site at:

http://www.ibm.com/software/data/db2imstools/db2tools-library.html

This chapter discusses the general requirements that you need to install DB2 Recovery Expert (DB2 RE) successfully. However, the requirements that we discuss here do not supersede the product documentation.

Note: Check to be sure that your DB2 UDB, operating system, and Java™ are at an appropriate level for DB2 RE to work properly. DB2 Recovery Expert V2 supports recovery in a 64-bit DB2 instance, although DB2 Recovery Expert is a 32-bit application.

3.2 Before you begin the installation process

DB2 RE must be installed on the same system where your database resides. DB2 RE needs access to the database, its log files, and its backups when it creates assets or attempts recovery. DB2 RE cannot perform these actions remotely.

Note: Having to have DB2 RE installed on the same system as your database might introduce challenges in your environment. Most of the DB2 RE functions require the use of the DB2 RE graphical user interface (the GUI), which means DB2 RE needs to run in a UNIX environment with some sort of X-windows display capability.

You need a root user ID to install DB2 RE and its companion products in UNIX environments.

You need an administrator user ID to install DB2 RE and its companion products in a Windows environment.

Typically, when you execute DB2 RE, you do so as the instance owner or as a user who has administrative rights on the database.

Note: If you are installing DB2 Recovery Expert V2 over an existing version of DB2 Recovery Expert, there are special steps to take and considerations to examine. These steps are documented in *IBM DB2 Recovery Expert for Multiplatforms: User's Guide*, SC18-956, which is available at:

`http://www.ibm.com/software/data/db2imstools/db2tools-library.html`

For this book, we start with DB2 Recovery Expert V2. Thus, we do not execute the special steps.

3.2.1 DB2 configuration requirements

For a database to be fully recoverable via DB2 RE, the database must be enabled for log archiving. The following information about how to set the logging and performance parameters is copied directly from the *IBM DB2 Recovery Expert for Multiplatforms: User's Guide*:

> These are the required settings in the DB2 database configuration parameters that ensure that Recovery Expert functions properly. If Recovery Expert is used to recover data from one database into another, the parameters must be set for both databases. In a DB2 multiple partition environment, you must apply these required settings to all partitions of the database. Refer to the *IBM DB2 UDB Data Recovery and High Availability Guide and Reference*, V8.2 SC09-4831 for more information about how to set these parameters.

Types of log files

There are different types of DB2 logging. Each type provides a different level of recovery capability. DB2 RE requires archive logging so that the log files are retained and can be analyzed to determine possible recovery options.

DB2 versions prior to DB2 UDB V8.2

For DB2 versions prior to DB2 UDB V8.2, archive logging is enabled for a database by turning on either the `LOGRETAIN` or `USEREXIT` configuration parameters, as follows:

- Setting `LOGRETAIN` to `YES`, `ON`, or `RECOVERY` causes DB2 to save archive log files in the directory that is specified by the database configuration newlogpath/path.

  As active logs fill, DB2 rolls filled logs into archive log files. These archive log files are used by DB2 RE during recovery analysis or log analysis. You should save the archived log files until they become obsolete, which typically occurs after a number of full backups have been done.

- Setting USEREXIT to YES causes DB2 to perform logging regardless of the LOGRETAIN setting. Setting the USEREXIT also indicates that a user exit program must be used to archive and retrieve log files. You must also ensure that the DB2 RE minilogs are updated before the user exit removes the logs from availability.

DB2 versions of DB2 UDB V8.2 or higher

In DB2 UDB V8.2 and higher, archive logging is enabled using the LOGARCHMETH1 configuration parameter. LOGARCHMETH1 accepts the following parameters:

OFF This setting is not recommended because log archiving will not take place.

LOGRETAIN Equivalent to the LOGRETAIN setting.

USEREXIT Equivalent to the USEREXIT setting.

DISK Writes the archive log to disk. This parameter must be followed by a colon (:) and then a fully qualified existing path name where the log files will be archived. For example, DISK:/u/dbuser/archived_logs.

TSM This parameter indicates that log files should be archived on the local Tivoli Server Management server using the default management class. If followed by a colon (:) and a Tivoli Server Management management class, the log files are archived using the specified management class.

VENDOR This parameter specifies that a vendor library is used to archive the log files. This parameter must be followed by a colon (:) and the name of the library. The APIs that are provided in the library must use the backup and restore APIs for vendor products.

Performance

The Application heap size (APPLHEAPSZ) database configuration parameter must be set to at least 128 to provide sufficient memory page space when DB2 RE is running.

Optional configuration settings

There are several other DB2 configuration settings that can affect how DB2 RE behaves. You do not need to set to any specific value. However, you should understand the effect of setting the following parameters:

- `DROPPED TABLE RECOVERY`

  Setting the `DROPPED TABLE RECOVERY` option on the table space allows you to use DB2 restore capabilities to recover a dropped table. However, this option also requires that you restore and roll forward the entire table space in order to recover the single table. DB2 RE offers a faster alternative to do this type of recover. Enabling `DROPPED TABLE RECOVERY` does not add any additional capabilities to DB2 RE, but because it is quite valuable for DB2, we recommend that you use it.

- `DATA CAPTURE CHANGES`

  Enabling the `DATA CAPTURE CHANGES` option for a table can have significant positive impact on DB2 RE performance. For information about this parameter, see 5.7.2, “Data capture changes” on page 114.

- `DB2_LOGGING_DETAIL`

  There is a DB2 registry variable called `DB2_LOGGING_DETAIL` that, when set, causes the authorization user ID to be included in each log record. This registry variable is undocumented at the time that we wrote this book (UDB V8 FP10 level). Thus, it might change in the future. However, this option is very useful for auditing data changes. For more information, see 5.1.1, “Prerequisites for log analysis for auditing” on page 84.

- Autonomic parameters

  The automatic maintenance features of UDB V8.2, such as automatic backups or automatic table space sizing, are not supported by DB2 RE. If you currently have automatic backups enabled and if you use the Versioning Repository, you must make a manual backup after you update the Versioning Repository, as described in 4.2, “Versioning Repository” on page 56.

- Log file inclusion in backup images

  DB2 RE can use a backup that is created using the `INCLUDE LOGS` parameter but ignores the log files that are included in the backup. DB2 RE cannot use the logs that are included in the backup for log analysis or roll forward.

- Compressed backups

  DB2 RE cannot process backups that are created using the `COMPRESS` parameter. This limitation is documented in the product readme file.

3.2.2 Space planning for DB2 Recovery Expert

This section introduces space planning topics for DB2 RE.

DB2 Recovery Expert software

The DB2 RE software and online help do not consume a large amount space — roughly 50 MB to 150 MB, depending on the platform. So, installing the software should not be a major space concern.

Note: When you install DB2 RE on a UNIX system, you must choose an install path. If there is not enough space there, the installer expands the file system. Thus, if you are short on space, you might need to do some pre-work on the file system. For more information, see 3.3.1, “Installing DB2 Recovery Expert on AIX in GUI mode” on page 26.

In some cases, DB2 RE might also install a Java Virtual Machine (JVM™) into the install path if it cannot find one. This installation consumes additional space. So, you might want to ensure that you have a supported Java JRE already installed on your system prior to installing DB2 RE. For details on the supported Java packages, see *IBM DB2 Recovery Expert for Multiplatforms: User’s Guide*, SC18-9564, which is available at:

http://www.ibm.com/software/data/db2imstools/db2tools-library.html

Recovery assets

As discussed in Chapter 1, “Introduction to DB2 Recovery Expert V2” on page 1, DB2 RE uses information from a variety of sources during recovery. Each of these recovery assets can consume space. Thus, you should think about these space requirements before implementing DB2 RE in your environment. These types of assets can consume space. However you do not need to do any special pre-work for these assets prior to installing DB2 RE. Consider the following assets before you install DB2 RE:

- DB2 RE recovery assets
  - Metadata that is captured from log analysis
  - Unloaded IXF files from recovery or object translation
  - Minilogs
  - Versioning Repository
  - DB2 Grouper sets
  - Fast Backup

- DB2 recovery assets
  - Logs that are retrieved from offline storage
  - Backups that might need to be retrieved from offline storage
  - Additional logs for remote log analysis

We describe the space planning details for these assets in 3.4, “Configuration tasks” on page 33. We describe how to create and manage these assets in Chapter 4, “Creating and managing recovery assets” on page 53.

3.3 Installing DB2 Recovery Expert

On all platforms, DB2 RE is installed using the InstallShield program. The steps are essentially the same on all platforms, so we do not show examples of each. The installation program name is different for each platform. Table 3-1 lists the installation program of each supported platforms.

Table 3-1 Installation program for each platform

Platform	Installation program
AIX	db2reaixsetup
Linux on x86/x86_64	db2rexlinuxsetup
Linux on pSeries®	db2replinuxsetup
Solaris	db2resolarissetup
Windows	db2rewin32setup.exe

3.3.1 Installing DB2 Recovery Expert on AIX in GUI mode

This section shows an example of installing DB2 RE on one of the ITSO AIX systems using the GUI. The windows that we show here look the same on any UNIX platform. The windows when you install on a Windows system are very similar, but we do not show them here.

To install DB2 RE on AIX in GUI mode, follow these steps:

1. As the root user, change to the directory where the install code is stored and start the InstallShield graphical installer by typing:

   ```
   ./db2reaixsetup
   ```

   This command launches the GUI as shown in Figure 3-1. Click **Next**.

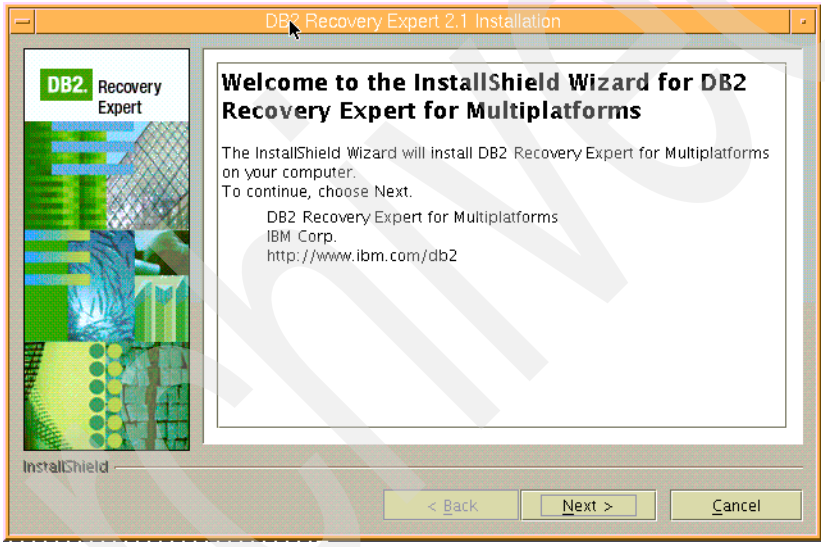


Figure 3-1 Welcome window

2. Agree to the license, as shown in Figure 3-2, and click **Next**.

Figure 3-2 The license agreement

3. Review the readme file, as shown in Figure 3-3, and click **Next**.

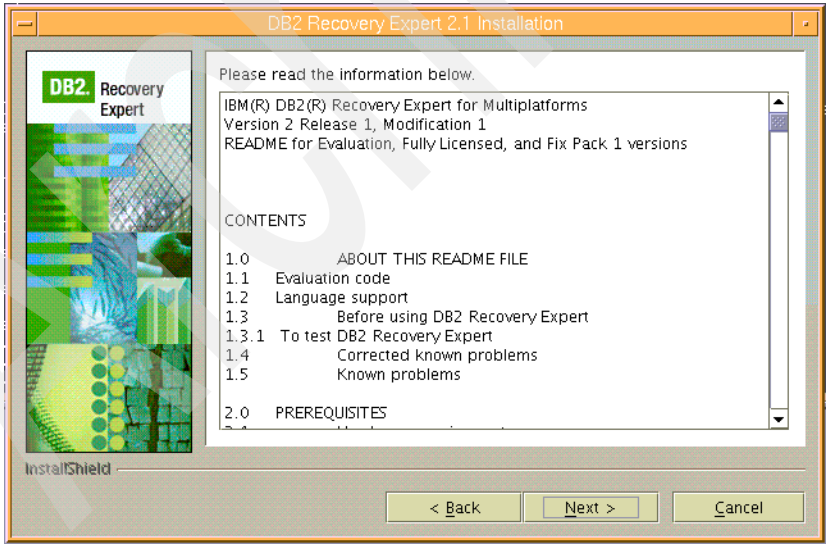


Figure 3-3 The readme file

Figure 3-4 shows the default installation path.

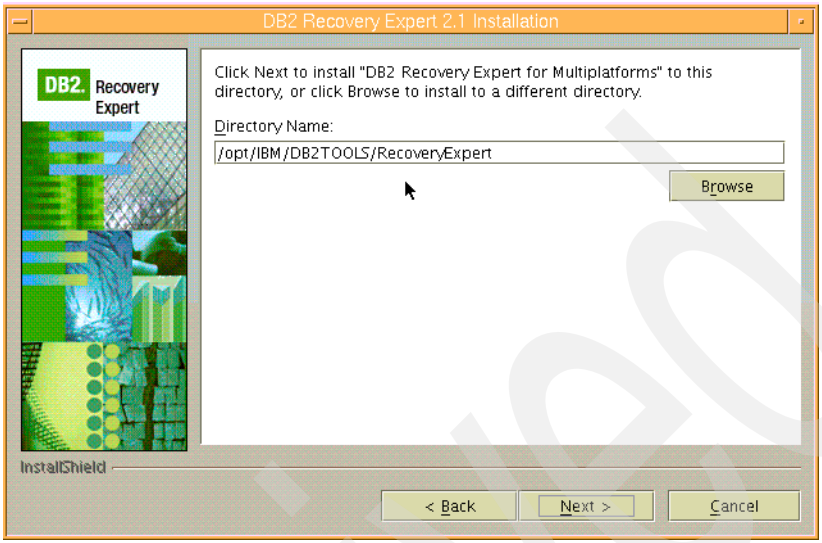


Figure 3-4 The default installation path

4. In this example, we change the installation path to be a bit shorter, as shown in Figure 3-5. Click **Next**.

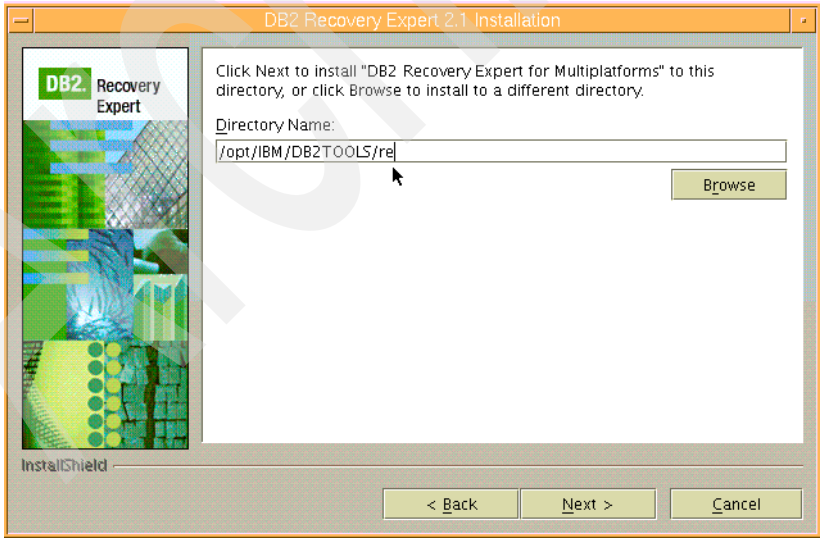


Figure 3-5 Using a different installation path

5. Choose the type of setup that you want, as shown in Figure 3-6. We choose a custom setup type for this example. If you plan to use Fast Backup (see Chapter 9, “Fast Backup” on page 247), you must use a custom setup. Otherwise, you can use the typical setup type. Click **Next**.

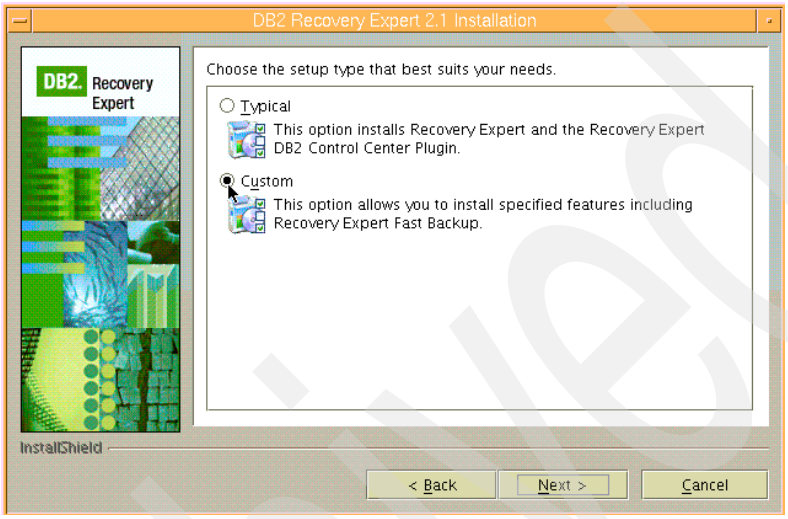


Figure 3-6 Selecting the setup type

6. Select **Recovery Expert** and **Recovery Expert DB2 Control Center Plugin** as shown in Figure 3-7. Click **Next**.

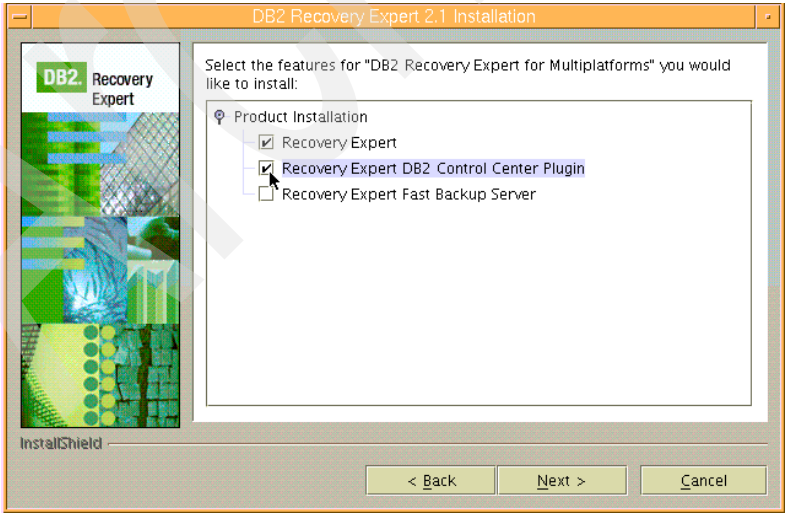


Figure 3-7 Custom installation options

7. When DB2 RE is ready to copy the files to the installation path, you must confirm the installation, as shown in Figure 3-8.

   In our example, notice that the /opt file system will be expanded. DB2 RE calculates the amount of space that it needs. If the installation path is not big enough, it expand it automatically. If you do not want DB2 RE to expand the installation file system, you can exit the installation process here. You can expand the installation path automatically, or you can find another location for the installation that has enough space. Then, you need to restart the installation. In our example, the installer expands the /opt file system.

   Note: You might wonder about the size of the installation, shown in Figure 3-8 as 154 MB. Part of the DB2 RE installation includes an Eclipse-based help system. The help is installed into a path named Eclipse301, which is one level up from the path that you specify. For example, in our case, the DB2 RE software is in /opt/IBM/DB2TOOLS/re/ and the help is /opt/IBM/DB2TOOLS/Eclipse301. The help consumes more space, so be sure to allow for that in your planning.

   You cannot choose to skip installing the help files.

   Click **Next** to proceed with the installation.

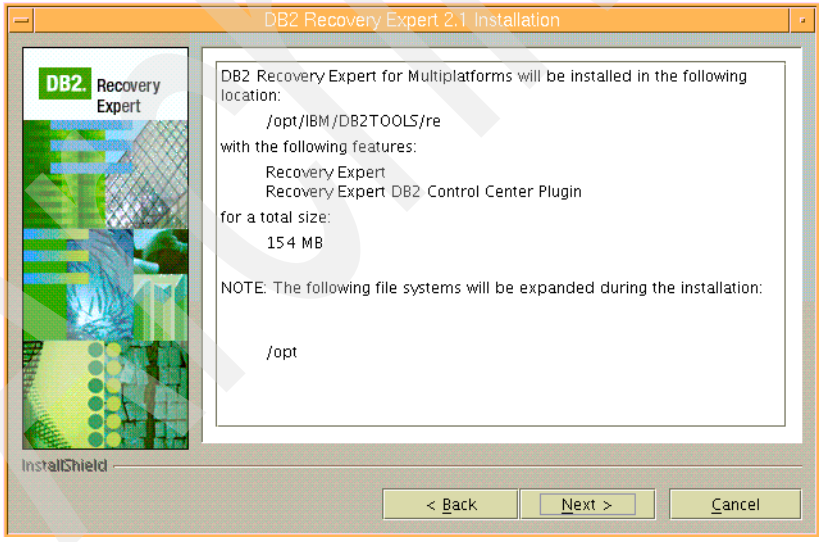


Figure 3-8 Confirming the installation

8. If the installation is successful, the window shown in Figure 3-9 opens. Click **Finish** to exit the installation program.

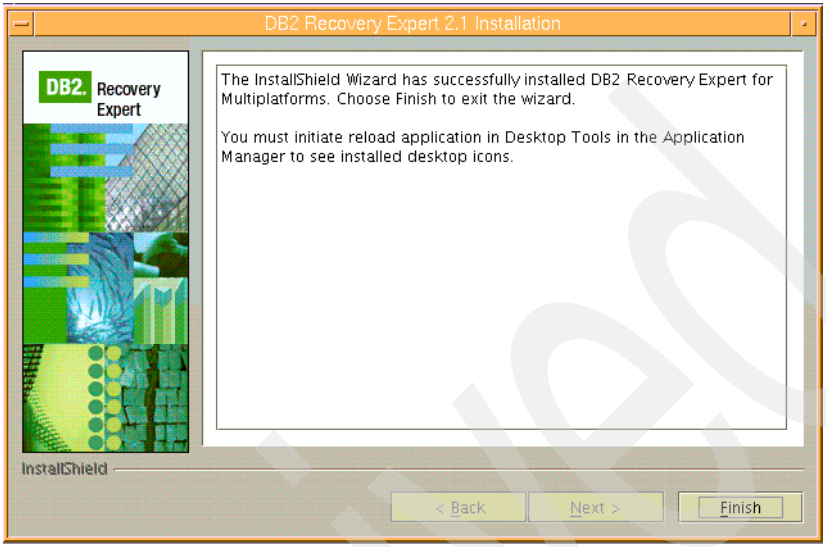


Figure 3-9 Installation successful

If you want to install the Fast Backup client, you should do this next, as described this in Chapter 9, “Fast Backup” on page 247.

If you want to install the DB2 Grouper component (both server and client), you should do this next. You can read about DB2 Grouper installation in Chapter 10, “Installing and using the DB2 Grouper component” on page 305.

Now that you have installed the DB2 RE files, you can proceed with configuration steps and first-time tasks as described in 3.4, “Configuration tasks” on page 33.

3.3.2 Installing DB2 Recovery Expert using the command line

If you do not want to use a graphical installation program, you can use a command line to install DB2 RE. The command line installation is supported on all platforms. As with the graphical installation, you must execute the command line installation program as root. When you invoke the installation program, you pass the `-console` option to the installer, as shown in the following AIX example:

```
./db2reaixsetup -console
```

The InstallShield line mode installer steps you through the installation process with the same sequence of questions and response options as with the GUI installation program.

Example 3-1 shows a clip from the beginning of a command line install session on Linux. The default response displays inside square brackets, for example [1]. You can press Enter to accept the default, or you can type a number. (This example does not include a full installation scenario.)

Example 3-1 Installing in console mode

```
[root@lead REV2]# ./db2rexlinuxsetup -console
InstallShield Wizard

Initializing InstallShield Wizard...

Searching for Java(tm) Virtual Machine...
.....................................
...................
-------------------------------------------------------------------------------
Welcome to the InstallShield Wizard for DB2 Recovery Expert for Multiplatforms

The InstallShield Wizard will install DB2 Recovery Expert for Multiplatforms on
your computer.
To continue, choose Next.

DB2 Recovery Expert for Multiplatforms
IBM Corp.
http://www.ibm.com/db2

Press 1 for Next, 3 to Cancel or 4 to Redisplay [1]
```

3.3.3 Reinstalling DB2 Recovery Expert

You cannot reinstall DB2 RE if it is already installed. If you do, the installer notifies you that you cannot reinstall the software, and it does not copy the installed files. If you accidentally delete or corrupt an important DB2 RE file, you need to uninstall DB2 RE completely and then install it again.

3.3.4 Uninstalling DB2 Recovery Expert

Note: The *IBM DB2 Recovery Expert for Multiplatforms: User's Guide*, SC18-9564, does not include the steps to uninstall DB2 RE.

This section describes briefly how to uninstall DB2 RE. The program than uninstalls DB2 RE does not remove any previously collected DB2 RE assets, such as log analysis metadata, minilogs, or Versioning Repository. If you uninstall only with the intent of reinstalling, you do not lose your assets. However, if you really want to remove all remnants of DB2 RE, you must delete the DB2

RE assets manually by deleting the entire DATA_DIR path and dropping the Versioning Repository partition group in each database. You might also need to delete the DB2 Grouper assets, if you are not using DB2 Grouper with another tool.

To uninstall DB2 RE, do the following according to your platform:

- Windows

  In a Windows environment, select **Start** → **Programs** → **DB2 Recovery Expert 2.1** → **Uninstall Recovery Expert** to launch the graphical uninstaller.

- UNIX

  In a non-Windows environment (such as AIX, Solaris, or Linux), login as root and launch the uninstall program using the following command:

  ```
  RE_install_dir/svruninst/uninstaller
  ```

  The program is an InstallShield graphical uninstaller.

  To uninstall in console mode, use this command:

  ```
  RE_install_dir/svruninst/uninstaller -console
  ```

3.4 Configuration tasks

The *IBM DB2 Recovery Expert for Multiplatforms: User's Guide*, SC18-9564, describes the configuration tasks and first steps for setting up the DB2 RE assets. We do not repeat all the information here. However, we attempt to clarify the process and show some examples. This discussion does not encompass every possible environment but should address the majority of the common questions about first steps with DB2 RE.

The three most important steps in configuration and first-time use are:

1. Customize the DB2 RE configuration file - recex.properties.
2. Create the Versioning Repository.
3. Build the Archive Log Inventory Data (ALID) metadata.

This section describes these tasks and provides more detail about space planning for DB2 RE assets. For information about other optional configuration steps, see *IBM DB2 Recovery Expert for Multiplatforms: User's Guide*, SC18-9564 at:

http://www.ibm.com/software/data/db2imstools/db2tools-library.html

3.4.1 Customizing the recex.properties file

DB2 RE has a configuration file where you can customize certain properties that control how DB2 RE behaves. The DB2 RE configuration file is called recex.properties and is laid out in a key=value format. This section describes the keys and value options, how to decide what values to use (if any), and the effect of making changes to this file.

We strongly recommend that you customize the recex.properties file before you start working with DB2 RE and that you re-examine your choices after you are more comfortable with how the file works and what kind of recovery activities you perform.

In a UNIX environment, the default permissions on the recex.properties file are set to be owned by root. The readme file suggests that you should not change the write permissions on this file. However, we found that development has lifted this restriction somewhat in DB2 Recovery Expert V2.1. Therefore, we recommend that after installation the root user change the ownership and permissions of the recex.properties file to allow write access by the group that contains the DB2 instance owners.

In our lab environment, all the instance owners were members of one group. So, we gave that group permissions to modify the recex.properties file. If your environment is different, you need to find suitable ownership and permissions that allow the appropriate personnel to modify the file. Alternatively, you can leave the file as owned by root and have the root administrator change the file as necessary. In some cases, DB2 RE continues to check the permissions on the file, which prevents you from starting DB2 RE. If that happens, DB2 RE displays an error message, and you need to modify the permissions appropriately at that point.

You can have one global properties file that affects all users and all instances, or you can create a separate properties file per instance. We describe both approaches here.

Global properties file

Example 3-2 shows the recex.properties file that we used on one of our ITSO lab systems, a Linux server named `LEAD`. We refer to this example in the following discussions of each of the properties.

Example 3-2 The recex.properties file from LEAD

```
# DB2 Recovery Expert configuration file
#

# define various product working directories in terms of
```

```
# the base DATA_DIR location:

DATA_DIR=/REassets
CONF_DIR={INSTALL_DIR}/conf

sysstorage.path={DATA_DIR}/sysstorage
minilogs.path={DATA_DIR}/minilogs
work.path={DATA_DIR}/work

# directives to configure the size and location of the
# logcache.path=<cache_directory_pathname>
# logcache.size=<size_in_megabytes>
logcache.path={DATA_DIR}/logcache
logcache.size=200

# directives to configure the starting log number for processing
# for each partition of each database:
# startlog.<database_name>.<partition_number>=<log_number>
#startlog.RECART.0=29
#startlog.TRADEDB.0=5
startlog.TRADSIMP.0=4

# enable this directive to prevent SQL files generated during the
# recovery process from being deleted:
recoverytask.keep_sql=true

# enable this directive to activate detailed event logging into
# the db2recex.log file (independently from DB2 DIAGLEVEL setting):
 log.level=4

# to define the product data directory and instance-specific
# configuration directory, use the DATA_DIR and CONF_DIR variables.
# Examples for Unix/Linux platforms are shown below:
# DATA_DIR={/}var{INSTALL_DIR}
# CONF_DIR={/}etc{INSTALL_DIR}
```

Configuration settings

The following describe the configuration settings in the `recex.properties` file and how to use these configuration settings:

- Comments

  Any line whose first character is a hash mark (#) is treated as a comment.

- Curly braces

  The default properties file shows all occurrences of the forward slash mark (/) to be enclosed in curly braces, for example {/}. This is not required for UNIX systems, but you can include them if you like. It is required for Windows

systems, because the property file parser cannot understand the Windows back slash character (\). For example these are all valid property settings for each environment:

- Windows: `sysstorage.path=C:{/}Mydir{/}sysstorage`
- UNIX: `sysstorage.path = {/}Mydir{/}sysstorage`
- UNIX: `sysstorage.path = /Mydir/sysstorage`

For ease of reading and typing in UNIX environment, we recommend that you not use curly braces. For Windows, of course, they are required.

Curly braces are also used within the definitions of the other properties as a way to indicate a variable. For example, you might want to set a work path to be underneath the DATA_DIR. You could hard code the full path name, or you could specify `{DATA_DIR}/workplace` in the property, which would cause the value for DATA_DIR to be substituted into the property. This is an important consideration when you have multiple instances.

- DATA_DIR

If the directory does not exist, DB2 RE creates it. DB2 RE also creates another subdirectory for each instance. This is where DB2 RE places its output files. For example, if the property is `DATA_DIR=/REassets` and you are working with an instance named `db2inst1`, the actual location of the files is `/REassets/db2inst1`. This is true on all platforms. The point to remember here is when using `{DATA_DIR}`, the instance name is added always to the path name.

The default location for DATA_DIR varies by platform:

- Windows: a path named DATA under the install path, for example: `C:\Program Files\IBM\DB2TOOLS\RecoveryExpert\data`
- UNIX: /var/*install_path*, or example: `/var/opt/IBM/DB2TOOLS/RecoveryExpert/`

We recommend that you not use the default in UNIX environments. You should create a separate file system or other isolated location for DB2 RE assets.

In our example, Example 3-2 on page 34, we created a separate directory named `/REassets` and placed all DB2 RE objects under that path.

Tip: If you have a DPF environment, the DATA_DIR must be on a shared file system, as described in the user's guide.

- CONF_DIR

  The configuration directory is where the recex.properties file resides. System administrators can edit this file to make system-wide changes to the configuration. If the directory does not exist, DB2 RE will *not* create it.

  The default location for CONF_DIR varies by platform:

  - Windows: a path named CONF under the install path, for example:
    `C:\Program Files\IBM\DB2TOOLS\RecoveryExpert\conf`
  - UNIX: /*install_path*/conf, for example:
    `/opt/IBM/DB2TOOLS/RecoveryExpert/conf`

  We recommend that you use the default location.

  In our example, Example 3-2 on page 34, we do specify the CONF_DIR property but do not override the default. Thus, the recex.properties file is located in /opt/IBM/DB2TOOLS/re/conf in this example.

Important: On UNIX platforms, the installation process adds two lines to the end of the properties file for DATA_DIR and CONF_DIR. By default, the CONF_DIR is added as /etc/install_path, which does not exist. Thus, it is not created, and the installed location (/*install_path*/conf) of the CONF_DIR is honored. We find this to be misleading and confusing. So, we specify the CONF_DIR as the location where it actually exists: /*install_dir*/conf.

We recommend that you make your own DB2 RE-specific DATA_DIR.

If a property is specified more than once, DB2 RE honors the last one. So, we recommend that you comment out or delete the two entries that are added at the end of the properties file.

- db2instance

  You might see references to this property in documentation, but do *not* try to set it. DB2 RE determines the instance internally. You should not set a property for it. In our example we do not specify this property.

- sysstorage.path

  The sysstorage location is where DB2 RE stores the assets that it creates and uses during recovery and log analysis. DB2 RE creates a subdirectory under the sysstorage path for each database and partition for the instance in which you are working. If you use the DATA_DIR name in the path, DB2 RE resolves that to include the instance name in the path.

  We recommend that you put this path under the DATA_DIR location. In our example, we set the `sysstorage.path={DATA_DIR}/sysstorage`.

Suppose we are working with single-partition instance `db2insl1`, and we have two databases, `RECART` and `TRADSIMP`. We also have another instance, `db2inst1`, with one database, `SAMPLE`. The actual directory tree looks like this:

```
/REassets/db2insl1/sysstorage/RECART/NODE0000/...
/REassets/db2insl1/sysstorage/TRADSIMP/NODE0000/...
/REassets/db2inst1/sysstorage/SAMPLE/NODE0000/...
```

Notice the instance name was inserted, and the database name and partition are added. This keeps all the DB2 RE assets separated by instance and by database.

If you have DPF instance, you would see the separate storage areas for each partition as follows:

```
/REassets/db2redpf/sysstorage/TPCH/NODE0000/....
/REassets/db2redpf/sysstorage/TPCH/NODE0001/....
/REassets/db2redpf/sysstorage/TPCH/NODE0002/....
```

- minilogs.path

  This property defines where the minilog output is stored. The default location is under the DATA_DIR. In our example, Example 3-2 on page 34, we choose to use the default.

- work.path

  The work.path property is where DB2 RE puts important interim files that it uses during recovery and log analysis. The default location is directly under the DATA_DIR path.

  Some of the files that go into work.path are the table data IXF files that are extracted during dropped table recovery and the generated Undo/Redo SQL that is created in log analysis or recovery.

  We recommend that you add a separate work.path subdirectory for clarity. In our example, Example 3-2 on page 34, we found the default location to be confusing and cluttered, so we specified another subdirectory named `work`.

- logcache.path

  When DB2 RE needs to retrieve a DB2 log file from offline storage such as Tivoli Storage Manager, it needs to have someplace to put the file. This property defines where to store the log files. The user's guide has a good explanation of this property. The default path is DATA_DIR/cache. If you specify a path that does not exist, DB2 RE creates it.

  We recommend setting this path to some specific path under the DATA_DIR so that you can keep track of all assets related to a particular instance and database easily. In our example, Example 3-2 on page 34, we set `logcache.path={DATA_DIR}/logcache`.

- logcache.size

  This property defines a ceiling for how much space is consumed by logs that are retrieved from offline storage. The user's guide has a good explanation of this property. The default size is 10 times the configured log size.

  If you have very large log files or are limited on space, you might want to or need to set this property. In our example, Example 3-2 on page 34, we set it just so we could see it work. However, we have no particular recommendation, because this parameter is dependent on your environment. The ceiling is about 2 GB. So, if you have that much space to spare, you can leave it as the default. Otherwise, you should set it smaller. Consider also the total size of the file system. If you have other DB2 RE files that might exist in the same file system, you can bump up against the file system size limit if you do not set this property.

  The size for this property is specified in megabytes. In our example, we set `logcache.size=200` which is interpreted by DB2 RE as 200 MB.

- startlog.dbname.partition

  The startlog property tells DB2 RE log analysis where to begin looking for transactions. By default, the starting point is log 0, which could be outdated in a real production database. The fewer logs DB2 RE has to scan, the faster the log analysis is. The user's guide has a good explanation of this property.

  We strongly recommend that you consider carefully what startlog value you want to set when getting started with DB2 RE. You should also consider how often you might want to update this value. For the first-time setup of the log analysis metadata, you probably want to choose a startlog value that requires DB2 RE to retrieve at least one log from offline storage (if you use offline storage). This setting proves that the interfaces are all working properly. The property is set per database and, in a DPF environment, per partition.

  In our example, we set the startlog value for one single-partition database (the others are commented out), `TRADSIMP` where we start with log 4 as follows:

  ```
  startlog.TRADSIMP.0=4
  ```

  Note: The log number is the numeric part of the log file name. For example, if you want to start with S0000294.LOG, do not specify the entire log file name `S0000294.LOG`. Just use the number `294`.

- recoverytask.keep.sql

  This property tells DB2 RE whether to overwrite the SQL file that is generated during a recovery. The default is `FALSE`, or unset, which means the file is not kept.

  We strongly recommend that you set this property to `TRUE`, which means the file is kept, because in most cases, you want to see what happened during a recovery. If part of the recovery involved undo or redo SQL, you should examine that SQL and keep a record of it in case other issues occur later. If you do not set this property, you have no record (other than in the DB2 logs) of what SQL was executed.

  The generated file is stored in the work.path location and is named ARY*xxxxx*.sql, where *xxxxx* is a numeric identifier. You can review the file after recovery and then rename it, delete it, or store it somewhere else for record keeping purposes.

  In our example, we set this property to `TRUE`.

- log.level

  This property influences how much detail is put into the DB2 RE log file. We describe this property in detail in “The DB2 Recovery Expert log file” on page 47.

  In our example we set `log.level=4`.

Instance-specific properties file

In the previous section, we described using one properties file and how DB2 RE creates separate work areas under the DATA_DIR for each instance. There might be cases, however, where you do not want to mix the instances together. You can also create separate properties files for each instance. The instance property file settings take precedence over the global property file. The syntax and layout of the properties themselves are identical to the global properties file. The differences are the location of the properties file, and the instance property file only needs the settings that are specific to that instance. It does not need to contain a full set of property settings.

To use an instance-specific properties file, you must place the file in a directory under the instance name. For example, suppose we have the same environment as described before, where we have one server with two instances called `db2insl1` and `db2inst1`. We want to use different DATA_DIRs for each instance.

So, we create a separate instance directory under the CONF_DIR location and put the customized recex.properties file in each path as shown here:

```
/opt/IBM/DB2TOOLS/re/conf/db2insl1/recex.properties
   DATA_DIR=/insl1assets
   CONF_DIR={INSTALL_DIR}/conf/db2insl1
   ... (the rest of the file is the same as before)

/opt/IBM/DB2TOOLS/re/conf/db2inst1/recex.properties
   DATA_DIR=/inst1assets
   CONF_DIR={INSTALL_DIR}/conf/db2inst1
   ... (the rest of the file is the same as before)
```

Note: We found that DB2 RE still wants to find a recex.properties file in the default location, *install_dir*/conf. So, you should keep one there, even though the *real* configuration file is in the instance directory.

Because the instance-specific properties file can be a bit tricky to understand, we recommend that you avoid using this approach if possible. There is one case where you *must* maintain separate properties file. If you have the same database name in more than one instance and if you want to set the startlog property, you must separate the properties file. The startlog property has no capability to distinguish between instances — only between partitions.

For example, suppose you have database `SAMPLE` in both `instA` and `instB`. The `instA` might need startlog value of 10, but `instB` might need 84. The startlog property is `startlog.`*`dbname.partition`*=n. There is no way to differentiate between `instA`'s need for startlog of 10 and `instB`'s need for startlog 84. You would have to create a separate properties file for each one. You do not have to change any other options in the properties file, they can still be the same otherwise.

3.4.2 Creating the Versioning Repository

Before starting to use DB2 RE, we strongly recommend that you create the Versioning Repository. You must do this for each database that you want to use with DB2 RE. (The steps to create the Versioning Repository are described in 4.2.2, “Creating the Versioning Repository” on page 56.)

You might be required to rebuild the Versioning Repository, especially after applying a DB2 RE fix pack. The fix pack readme file will say if this step is required. (We describe how to rebuild the Versioning Repository in 4.2.7, “Rebuilding the Versioning Repository” on page 68.)

3.4.3 Creating the Archive Log Inventory Data

Before starting to use DB2 RE, you must create the initial Archive Log Inventory Data (ALID). The first time you create the ALID, it can be time-consuming. Thus, we identify this task as part of the getting started tasks. You do not want to do a recovery only to find that the ALID does not exist. Then, you pay the price of creating it for the first time again. Instead, you can create it once (which takes some time) and update it often (which takes less time). You might also want to or need to rebuild the ALID from time to time, such as when you modify the startlog property.

We describe how to create and rebuild the ALID in 4.4, "Archive Log Inventory Data (DB2 RE metadata)" on page 77.

Important: Be sure to consider and to set the startlog property for each database. This is the single most important factor when you first create the ALID. See 4.4.2, "Creating and updating the ALID" on page 78 for details and Chapter 5, "Log analysis" on page 83 for usage examples.

3.4.4 Space planning for DB2 Recovery Expert and recovery assets

This section describes space planning in detail. DB2 RE does not use this space until you start using the product. The most important space decision is where DB2 RE puts its work files. (These work files are the DB2 RE assets that we refer to throughout this book.) If you do not customize the properties file, you should ensure that you have enough space in those paths to accommodate the DB2 RE assets. We describe how to set the properties in 3.4.1, "Customizing the recex.properties file" on page 34.

DB2 Recovery Expert assets

This section discusses the assets that DB2 RE uses and whether there is a configuration property that influences how the asset is used. (The list here maps to the list shown in "Recovery assets" on page 24.)

DB2 RE assets include the following:

- Metadata that is captured from log analysis

  This asset is the log information generated during log analysis. There is some static information (the ALID) and also more detailed information if you run the detailed reports. This data can be large, depending on the time span for which you ask.

  - Controlled from property in recex.properties?

    Yes, *sysstorage.path* defines the location of ALID and related files.

- Is there a ceiling on the amount of space consumed?

  No, the metadata uses what space it needs. However, it changes with each log analysis, so the space used fluctuates.

- Is any cleanup or maintenance required?

  No.

- Unloaded IXF files from recovery or object translation

  During certain types of recovery, DB2 RE might need to extract the data from a backup. This data is extracted in IXF format. If you have large tables to extract, or LOB type data, this information can consume a lot of space.

  - Controlled from property in recex.properties?

    Yes, indirectly. The unloaded IXF files are placed under the work.path location.

  - Is there a ceiling on the amount of space consumed?

    No, the IXF files use space as needed. If you work with large tables you should see how much space is available in the work.path or risk filling the file system.

  - Is any cleanup or maintenance required?

    Yes, volatile assets that are used during recovery or cloning but that are not needed afterward can be deleted when recovery is done to reclaim space.

- Minilogs

  Minilogs are special DB2 RE subsets of DB2 transaction logs and are generated by the **db2mm** command.

  - Controlled from property in recex.properties?

    Yes, using the minilogs.path property.

  - Is there a ceiling on the amount of space consumed?

    No, minilogs use as much space as needed.

  - Is any cleanup or maintenance required?

    No, minilogs can grow when you run the **db2mm** command. So, you might want to treat them more like DB2 logs for purposes of storage location and backup policies.

- Versioning Repository

  The Versioning Repository is a set of tables in a table space inside each database you want to use DB2 RE with. Versioning Repository is described in 4.2, “Versioning Repository” on page 56.

  - Controlled from property in recex.properties?

    No. The Versioning Repository is a DB2 object.

  - Is there a ceiling on the amount of space consumed?

    No. The Versioning Repository uses as much space as it needs but, in our experiments, the space it uses is small.

  - Is any cleanup or maintenance required?

    You can prune the Versioning Repository as described in 4.2.6, “Pruning the Versioning Repository” on page 66.

- Fast Backup

  To DB2 RE, the only assets that Fast Backup creates is the backup in the network storage volumes. This asset is available if a database or table space is located in the network storage volumes such as IBM ESS.

  - Controlled from property in recex.properties?

    No. Fast Backup are controlled from property located in the Fast Backup administration repository database.

  - Is there a ceiling on the amount of space consumed?

    Yes. This volume is limited by the network storage volume size.

  - Is any cleanup or maintenance required?

    No. You can manage the number of Fast Backup assets using Fast Backup client service. Normally, these assets are replaced with the new images whenever new Fast Backup are made.

- DB2 Grouper

  DB2 Grouper sets are stored as DB2 objects inside the DB2 Grouper table space. All DB2 Grouper functions are described in Chapter 10, “Installing and using the DB2 Grouper component” on page 305.

  - Controlled from property in recex.properties?

    No. DB2 Grouper sets are controlled completely from the DB2 Grouper client.

  - Is there a ceiling on the amount of space consumed?

    No.

  - Is any cleanup or maintenance required?

    You can remove old versions using the DB2 Grouper client.

DB2 recovery assets

DB2 owns and creates the following assets, but DB2 RE needs them also:

- Logs that are retrieved from DB2 offline storage

  These are the DB2 transaction logs used during Log Analysis or during a roll forward recovery.

  – Controlled from property in recex.properties?

    Yes, logcache.path defines the location for retrieved log files.
    Yes, logcache.size sets a ceiling on how much space can be consumed.

  – Is there a ceiling on the amount of space consumed?

    Yes, by honoring the logcache.size property. If no property is set, the maximum size used is 2047 MB.

  – Is any cleanup or maintenance required?

    No, but you can delete the logs if you no longer need them for DB2 RE. When DB2 RE retrieves the logs, they are not removed from the original location.

- Backups that might need to be retrieved from offline storage

  If you have a special recovery scenario in which the backup cannot be retrieved via normal DB2 APIs or if the backup is not available in its normal location, you might need to allow for space to put a disk copy of the backup.

  – Controlled from property in recex.properties?

    No.

  – Is any cleanup or maintenance required?

    To save space, you can remove the backup or return it to offline storage when it is no longer needed by DB2 RE.

- Additional logs for remote log analysis

  There can be recovery scenarios in which you have DB2 logs in a different location from where DB2 RE might look for them by default. You should allow enough space to hold those logs

  – Controlled from property in recex.properties?

    No.

  – Is any cleanup or maintenance required?

    To save space, you can remove the logs, when they are no longer needed by DB2 RE. Ensure they still exist someplace before you erase them, of course.

3.5 Troubleshooting

This section highlights some issues that you might experience when installing DB2 RE. We do not attempt to solve each problem but rather point out where you should look first if you encounter an issue.

3.5.1 Troubleshooting the installation

This section describes some of the typical or most common issues that you might encounter during the installation process. If you work through this section and continue to have issues with the installation, contact IBM Support.

- Permissions

  If you are installing on a UNIX system, check to be sure your install file has execute permissions.

- root and umask

  The install must be performed as root, and it must have a `umask` setting of 022. You can verify this with the **`umask`** command.

- Java issues

  Java issues include the following:

  - Ensure that your DB2 instance is enabled for Java. For more information, check the DB2 Information Center at:

    `http://publib.boulder.ibm.com/infocenter/db2help/topic/com.ibm.db2.udb.doc/ad/t0004675.htm`

  - DB2 RE needs a Java at 1.3.1 or higher. We used 1.4.1 and 1.4.2 in our tests.
  - Check that the JDK_PATH value in the DBM configuration is set.

- 32/64-bit issues

  DB2 RE needs to use a 32-bit Java. So, you might need to install one if you do not already have one. You should use an IBM Java that is certified for DB2.

- Space problems

  The DB2 RE attempts to expand the file system to be large enough to hold the installed files. It is possible that the expansion will exceed the capacity of the volume, which results in an error. You should ensure that you have enough space before you begin the installation process, rather than wait for an error to occur.

As of the writing of this book, and with the base DB2 Recovery Expert V2, the amount of space that you need for UNIX systems is as follows:

– installed path: 50 MB
– /var: 100 MB
– /tmp: 150 MB

These sizes can vary. In AIX, the DB2 RE installation might also install Java 1.4.2 in the product installation directory. Thus, that installation will consume more space.

3.5.2 Troubleshooting DB2 Recovery Expert

This section describes the DB2 RE log file, and then provides information about some common issues with DB2 RE and how to detect, prevent, or recover from these issues.

The DB2 Recovery Expert log file

If you experience problems when running DB2 RE, it is useful to see if any errors or warnings are being logged. DB2 RE does have the capability to log diagnostic information about actions it takes in the GUI. The first and best problem determination aid for DB2 RE is the DB2 RE log file.

Restriction: DB2 RE does not currently log information to a file when you issue commands from the command line. It only logs GUI tasks.

There are two things to consider when you want to look at the DB2 RE diagnostic log information:

- Location of the DB2 RE log file
- Amount of information in the log file

Location of the log file

The default location of the log file varies depending on the platform:

- Windows platforms

  On Windows platforms, DB2 RE logs warning and error-level messages inside the db2diag.log file for the DB2 instance for which it is running. The location of this file is specified in the `DIAGPATH` parameter in the database manager configuration. To see the database manager configuration, use the command:

  ```
  db2 get dbm cfg
  ```

  For the default DB2 instance (DB2), the location of this file is usually:

  C:\Program Files\SQLLIB\DB2\db2diag.log

- Non-Windows platforms

  On other platforms, DB2 RE does not use the db2diag.log file, instead it creates its own db2recex.log file, which is located in the path specified by the DIAGPATH database manager configuration parameter for the DB2 instance being used. To see the DIAGPATH, use the command:

  ```
  db2 get dbm cfg |grep DIAGPATH
  ```

  The default path is ~/sqllib/db2dump (where ~ is the instance home directory.)

  You can, of course also use the DB2 Control Center to retrieve this information, as shown in Figure 3-10, which shows that the DIAGPATH is blank, meaning it uses the default.

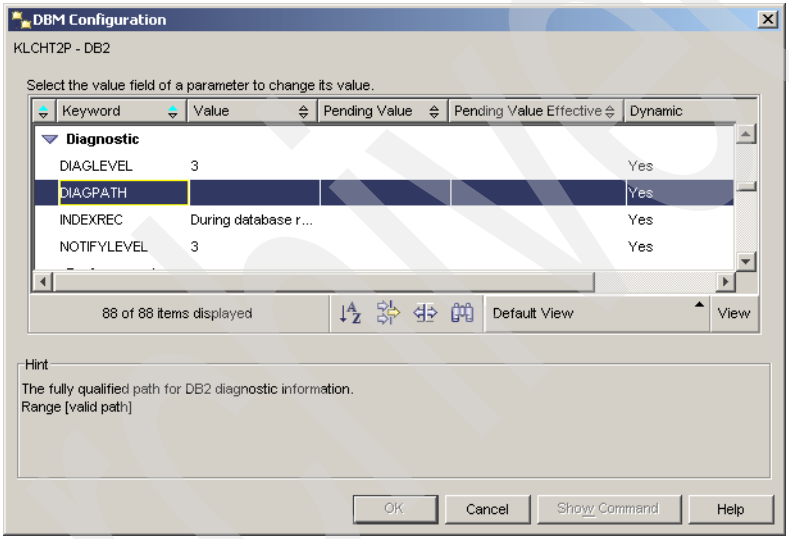


Figure 3-10 DIAGPATH value in the DB2 Control Center

Changing the amount of information logged

DB2 RE honors the DB2 DIAGLEVEL database manager configuration parameter. The default value is 3, which logs errors and warnings. If you change DIAGLEVEL to 4 to include informational messages, DB2 RE also writes more detail to the log file. However, if you are only interested in DB2 RE messages, you do not set the DB2 DIAGLEVEL to 4 because you get a large amount of extra information about DB2 as well.

With DB2 Recovery Expert V2, you can now specify an additional logging directive for DB2 RE only in the recex.properties file. This property is called log.level, and it has the same usage as the DIAGLEVEL. The default value is 3

(warnings and errors). To get more information, you can set it to `4` (warnings, errors, and informational messages.)

We recommend setting `log.level=4`, because the information that DB2 RE provides is necessary, especially when you are just getting started with DB2 RE. There are a few differences between the Windows and non-Windows platforms with respect to the log.level setting:

- Windows platforms

  When you set `log.level`, DB2 RE uses the db2recex.log file. If you do not enable the `log.level` property, DB2 RE writes to the db2diag.log file. If you always want DB2 RE messages in the db2diag.log file, the only option you have is to use the `DIAGLEVEL` DBM configuration parameter and pay the price of getting the DB2 informational messages as well as the DB2 RE messages. We do not recommend this approach.

- Non-windows platforms

  DB2 RE always writes to the db2recex.log file for UNIX and Linux platforms, so enabling the `log.level` property only changes the amount of detail in the log file. The amount if detail is not excessive and is of such good value that we recommend always using `log.level=4`.

Table 3-2 describes the results of different settings and our recommendations.

Table 3-2 Location and contents of DB2 RE application log file

	DB2 RE log data location and detail level	
	Windows	**non-Windows**
default	► db2diag.log ► level 3 - warning/error	► db2recex.log ► level 3 - warning/error
DB2 DIAGLEVEL 4	► db2diag.log ► level 4 - warning/error/info for DB2 RE and DB2	► db2recex.log ► level 4 - warning/error/info for DB2 RE only
log.level=4 *	► db2recex.log ► level 4 - warning/error/info for DB2 RE only	► db2recex.log ► level 4 - warning/error/info for DB2 RE only
* recommended setting		

3.5.3 Using online help and finding error messages

DB2 RE uses an Eclipse-based help system, which displays the help in your default browser. The common browsers (Mozilla, Firefox, Internet Explorer) are supported on the appropriate platforms. For more information, see *IBM DB2 Recovery Expert for Multiplatforms: User's Guide*, SC18-9564 at:

http://www.ibm.com/software/data/db2imstools/db2tools-library.html

The help is installed automatically when you install DB2 RE.

To launch the online help for DB2 RE, select **Help** → **Recovery Expert Help** from the DB2 RE menu bar, as shown in Figure 3-11. You can also launch online help directly from the Log Analysis window, by clicking **Help**.

Figure 3-11 Launch online help from DB2 Recovery Expert menu

The Eclipse-based help for DB2 RE has the same look and feel of the standard DB2 Information Center online help.

If you install the DB2 Grouper client, the DB2 Grouper help is available also in the same browser session. In Figure 3-12 on page 51, we show a sample of online help for a Windows machine that has DB2 RE and DB2 Grouper installed. Note that the Fast Backup help is included in the DB2 RE help, even if you do not install Fast Backup.

The DB2 Grouper help opens only if you install the DB2 Grouper client. Thus, it only opens in Windows, because the client is only available for Windows.

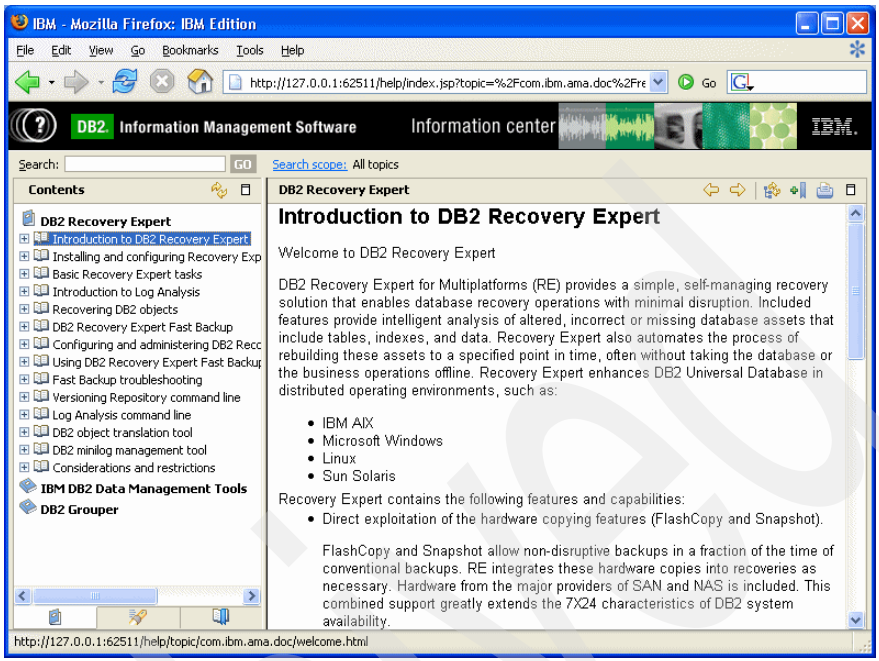


Figure 3-12 Online Help

The information in the online help parallels the information in *IBM DB2 Recovery Expert for Multiplatforms: User's Guide*, SC18-9564, except the online help does not contain the DB2 RE and Fast Backup error message descriptions from the appendix. If you encounter a DB2 RE error (ARY*xxxxx*) or Fast Backup error (GDB*xxxxx*), you must look it up in the user's guide.

The online help is served via an internal (to DB2 RE) application server on a port that is opened for the duration of the DB2 RE session. Notice that in the URL, the host name is the loopback address. Because the help browser session is tied to the DB2 RE session, if you exit DB2 RE, the help is no longer operational.

4

Creating and managing recovery assets

DB2 Recovery Expert can become an integral part of your recovery strategy. However, you must be sure that everything is in place to facilitate the various functions that are available to DB2 Recovery Expert. This chapter provides details about which assets are required for a successful recovery strategy and how to maintain these assets. It contains the following sections:

4.1 Backups

This section discusses the importance of backups, the types of backups that are available, and how often you should schedule backups.

4.1.1 The importance of backups

It is very important to take regular backups to protect your data. Regular backups help to reduce the time it might take to recover the database or table space if recovery is required. The more frequently the backup, the fewer logs that DB2 UDB has to roll forward through (on average) to recover a database or table space. This strategy is especially important when using DB2 Recovery Expert (DB2 RE). DB2 RE uses backups for the following:

- Recovering a dropped table

  The Versioning Repository only stores the DDL for table spaces, tables, and all related objects. A full offline backup is required from which DB2 RE extracts an image of the data in the table.

- Recovering a table to a point-in-time

  DB2 RE can extract an image of the table from a full offline backup as a starting point for recovering a table to the point-in-time using Redo SQL.

- Cloning a database

  DB2 RE can use a full offline backup image as a source for cloning.

- Log analysis

  DB2 RE can use a full online or offline backup as a starting point when reconstructing UPDATEs.

Having a more recent backup generally reduces the time that is required for any of these tasks.

4.1.2 Backup types

DB2 RE supports both full online and offline backups when using log analysis. It supports full offline backups when using object translation. It supports online, incremental, or delta backups for recovery operations where DB2 Restore is called to perform the recovery.

Fast Backup

DB2 Recovery Expert V2 introduces Fast Backup. This is a hardware-level backup, not a DB2 backup. Fast Backup functionality is integrated with DB2 RE. Therefore, DB2 RE can also use Fast Backup image copies as a backup asset to

perform recovery operations. See Chapter 9, “Fast Backup” on page 247 for further details about Fast Backup.

4.1.3 Tivoli Storage Manager and other vendor libraries

DB2 Recovery Expert V2 works directly with Tivoli Storage Manager and other vendor libraries. Thus, you do not need to extract the images to disk. During the recovery process, DB2 RE determines the list of backups and on which type of media the backups are stored. When a backup is selected, DB2 RE works directly with the storage media server.

4.1.4 Scheduling backups

DB2 RE does not schedule backups. It is your responsibility to schedule backups. You can schedule backups using any scheduling tool, such as DB2 Script Center, **cron** (on AIX), or a Windows task scheduler (such as `at`). The database administrator decides how often to take backups, taking into consideration the following factors:

- Importance of reducing recovery time

  If it is essential that recovery is as quick as possible, then you should take backups as often as possible. If recovery time is less important, then you can reduce the frequency of backups.

- Frequency of DDL run

  With DB2 RE, a dropped object can be recovered only to immediately before the drop, if both of the following are true:

  – A backup has been taken since the time the object was created.
  – The versioning repository has been updated since the last DDL was run that affected the object.

  Ideally, you update the versioning repository and take a backup of the relevant table space immediately after running any DDL. That way, you are always able to recover dropped objects to immediately before the drop. This is fairly impractical in a real environment, but it is still important to consider how often DDL is run when deciding how often to schedule backups and Versioning Repository updates.

- Batch windows

  Backups require resources. So, it is best to take backups when there is as little activity as possible in the database (ideally, when there is no activity at all). The usage patterns of the database and the existence or length of batch windows are, therefore, important when deciding how often to schedule backups.

4.2 Versioning Repository

This section discusses the importance of the Versioning Repository, how to create and update it, how often to update it, how to prune it, and how to rebuild it.

Note: DB2 RE windows, documentation, and help might also refer to the Versioning Repository as the *Repository*. For consistency in this book, we refer to it as the *Versioning Repository*.

4.2.1 The importance of the Versioning Repository

The Versioning Repository stores definition information about database objects so that you can recover these object if they are dropped accidentally. The Versioning Repository does not store any data for the tables, only the table structure DDL type of information.

DB2 Recovery Expert V2 can now recover objects other than tables, table spaces, and databases. These objects are also defined in the Versioning Repository. The new objects are:

- Buffer pool
- Event monitor
- Partition group
- Schema
- Stored procedure
- User-defined type

4.2.2 Creating the Versioning Repository

You should create the Versioning Repository immediately after you install DB2 RE. You must create and maintain a Versioning Repository for each database for which you will use DB2 RE. You might also need to rebuild or to recreate the Versioning Repository. For example, when you install DB2 Recovery Expert V2 over an existing DB2 Recovery Expert V1, you must rebuild the Versioning Repository. You can create the Versioning Repository using either the GUI or command line, as described in the following sections.

Creating the Versioning Repository using the GUI

The GUI method for creating the Versioning Repository is the same for all platforms. When you use the GUI, the Versioning Repository is created and is populated with object information from the DB2 system catalog.

To create the Versioning Repository using the GUI, follow these steps:

1. Start the DB2 RE GUI.
2. Connect to your database.
3. From the menu, select **Tools** → **Versioning Repository** → **Create** as shown in Figure 4-1.

Note: The Versioning Repository menu offers three choices: create, update, and prune. If the Versioning Repository already exists, the create option is disabled. The create option is available only when the Versioning Repository does not yet exist. We discuss how to drop or re-create the Versioning Repository in 4.2.7, “Rebuilding the Versioning Repository” on page 68.

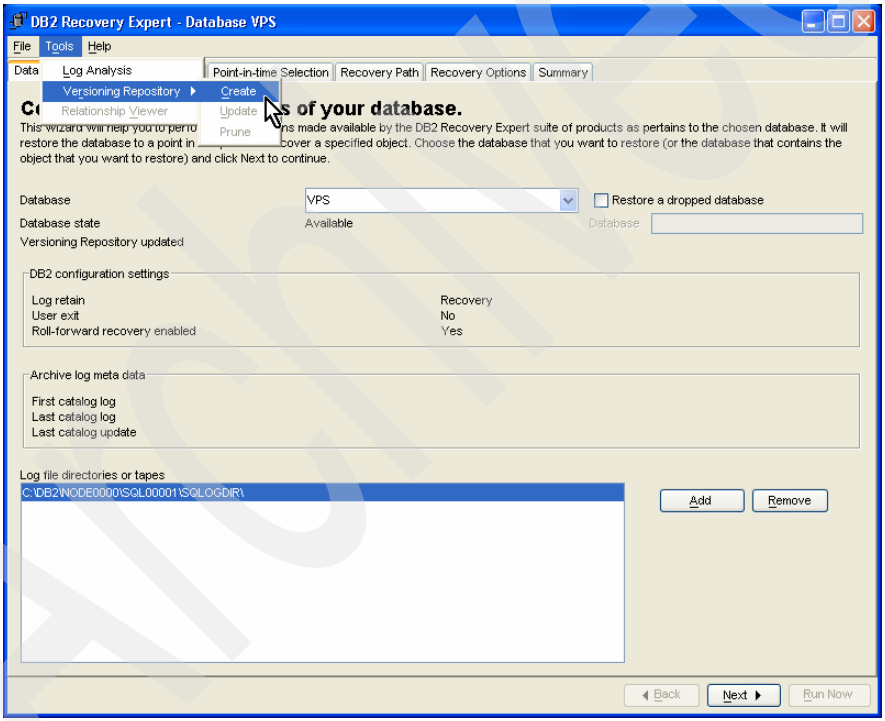


Figure 4-1 Creating the Versioning Repository using the GUI

4. Click **Run** to execute the DDL, as shown in Figure 4-2.

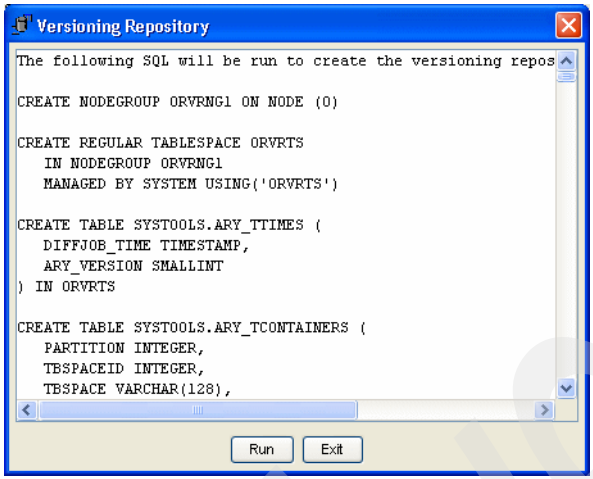


Figure 4-2 Running the Versioning Repository DDL

Watch the execution status, and wait for the completion message. The tables are created first, and then they are populated. Depending on how complex your current database structure is, it might take a while to complete. In our use, it usually ran in about one minute. The status window shows the start time, so you can watch the progression, as shown in Figure 4-3.

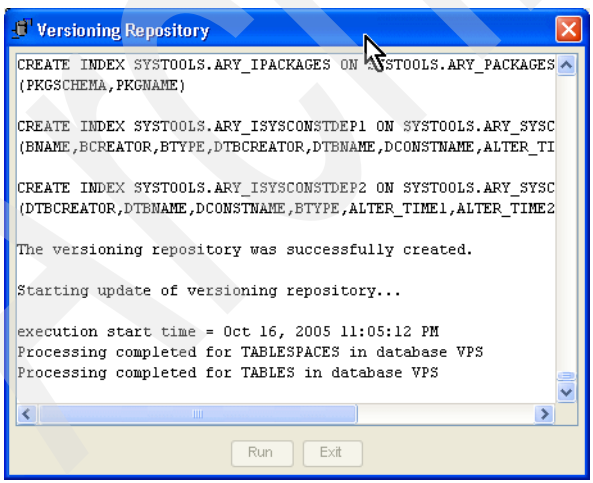


Figure 4-3 Watching the status for create Versioning Repository

When the creation completes, a message is displayed, as shown in Figure 4-4.

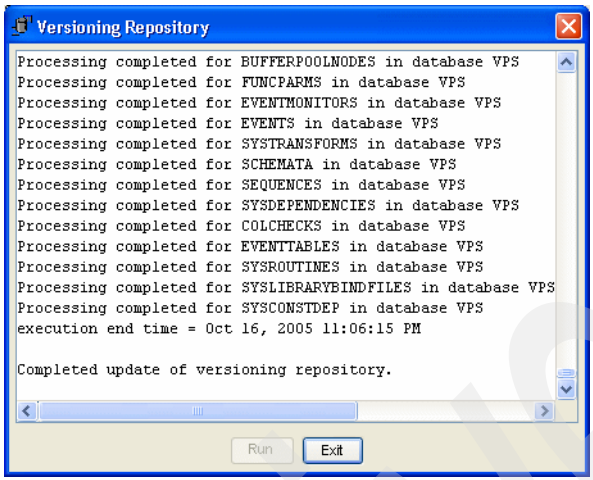


Figure 4-4 Completed Versioning Repository

5. Click **Exit** to close the window.
6. Now the DB2 RE status window shows the most recent date and time when the Versioning Repository was updated, as shown in Figure 4-5. Knowing this information is useful if you need to do object recovery later. If you schedule the Versioning Repository to run in a **cron**, for example, you can check this status to be sure the Versioning Repository is current.

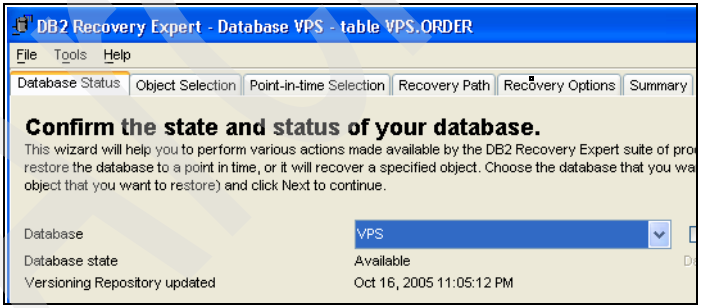


Figure 4-5 DB2 RE status window showing Versioning Repository status

Creating the Versioning Repository using the command line

There are two ways to create the Versioning Repository using the command line, as described in the following sections.

Running the Java utility

DB2 RE provides a Java interface to the Versioning Repository which allows you to create, update, and prune the Versioning Repository by running a command line program. There are slight differences in how to invoke the Versioning Repository Java utility on Windows and non-Windows. When you run the **CreateVersionRepository** Java utility from the command line, it creates and updates the Versioning Repository, just as in the GUI. The command line that you use depends on whether you are running on a Windows or UNIX platform, as follows:

- Windows

  Log in to Windows as a user who has create privileges in the database that you want to affect. Enter the following command on one line:

  ```
  db2rewin32 -is:javaconsole CreateVersionRepository -d database -u userid
  -p password
  ```

  If you omit the user ID or the password, you are prompted for the values.

  In Windows, the status messages open in a new window that is launched when you run the command. The window closes when the command finishes. The messages are also logged in the db2recex.log file. We describe how to configure DB2 RE to use this log file in “The DB2 Recovery Expert log file” on page 47.

- UNIX platforms

  In AIX, Solaris, and Linux, log in as the instance owner to run this command. Enter the following command on one line:

  ```
  db2recex CreateVersionRepository -d database -u username -p password
  ```

  If the DB2 RE installation path is not in your $PATH, you must prefix the command with the full path to the db2recex program. For example:

  ```
  /opt/IBM/DB2TOOLS/re/bin/db2recex CreateVersionRepository -d TRADSIMP
  -u db2insl1 -p xxxx
  ```

  Of course, you could also change to the install path (**cd**) before executing the program.

  The output messages open in the window and are also logged to the db2recex.log file. We describe how to configure DB2 RE to use this log file in “The DB2 Recovery Expert log file” on page 47.

Running DDL

You can run the same DDL that the GUI runs. The DDL is in a file named objres.ddl that is located in the DB2 RE installation path. You should connect to each database for which you want to use DB2 RE, and then run the DDL using your favorite method. Example 4-1 shows a portion of Windows example.

Unlike the GUI and the Java program, which run the create and the update together, if you choose to create the Versioning Repository manually using the command line, you must run the create and then run the update. The steps for running the update are described in 4.2.3, “Updating the Versioning Repository” on page 62.

You should not modify the DDL in any way.

Example 4-1 Running objres.ddl from command line

```
C:\DB2Tools\DB2RE>db2 connect to vps

   Database Connection Information

 Database server        = DB2/NT 8.2.1
 SQL authorization ID   = MARCIA
 Local database alias   = VPS

C:\DB2Tools\DB2RE>db2 -tvf objres.ddl
CREATE NODEGROUP ORVRNG1 ON NODE (0)
DB20000I  The SQL command completed successfully.

CREATE REGULAR TABLESPACE ORVRTS IN NODEGROUP ORVRNG1 MANAGED BY SYSTEM
USING('ORVRTS')
DB20000I  The SQL command completed successfully.

CREATE TABLE SYSTOOLS.ARY_TTIMES ( DIFFJOB_TIME TIMESTAMP, ARY_VERSION SMALLINT
) IN ORVRTS
DB20000I  The SQL command completed successfully.

CREATE TABLE SYSTOOLS.ARY_TCONTAINERS ( PARTITION INTEGER, TBSPACEID INTEGER,
TBSPACE VARCHAR(128), CONT_NAME VARCHAR(256), CONT_TYPE VARCHAR(9), NUM_PAGES
VARCHAR(100), ALTER_TIME1 TIMESTAMP, ALTER_TIME2 TIMESTAMP, VERSION_END_TIME
TIMESTAMP ) IN ORVRTS

DB20000I  The SQL command completed successfully.
...
```

4.2.3 Updating the Versioning Repository

The Versioning Repository must be kept up-to-date, because it is the sole reference for recovering dropped objects. You should update the Versioning Repository any time you make DDL changes.

Updating the Versioning Repository using the GUI

The GUI update method is the same for all platforms. When you use the GUI to update the Versioning Repository, DB2 RE examines the current system catalog object definitions and saves the changes to the Versioning Repository tables with additional information that is specific to DB2 RE. To update the Versioning Repository using the GUI, follow these steps:

1. Start the DB2 RE GUI.
2. Connect to your database for which you want to update the Versioning Repository.
3. From the menu, select **Tools** → **Versioning Repository** → **Update** as shown in Figure 4-6.

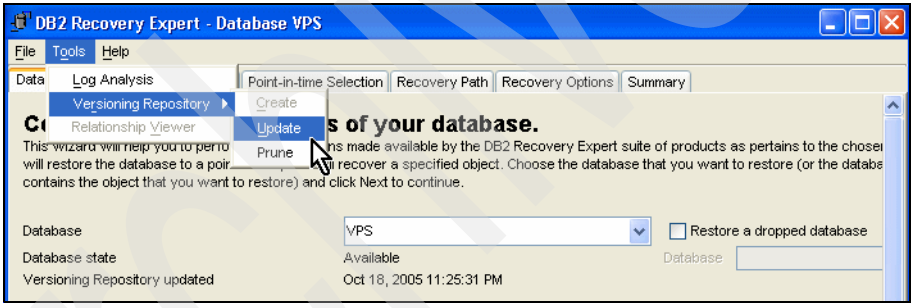


Figure 4-6 Updating Versioning Repository using the GUI

4. Click **Run** to initiate the update, as shown in Figure 4-7.

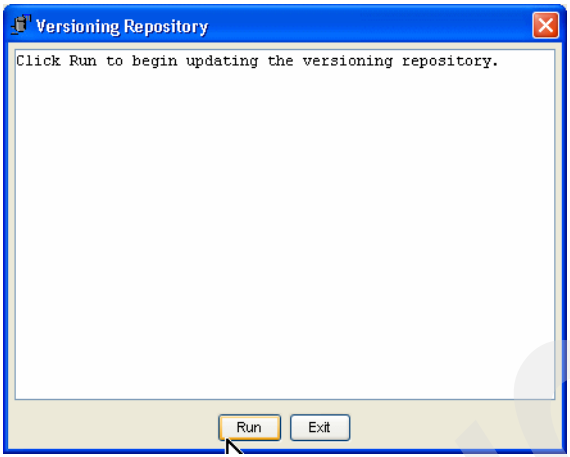


Figure 4-7 Running the Versioning Repository update task

You can watch the update status and wait for the completion message. The duration of the update is not long but depends on how much DDL activity has occurred since the last Versioning Repository update.

5. When the update is complete, click **Exit** to close the window.

   The DB2 RE status window always shows the most recent Versioning Repository update date and time.

Updating the Versioning Repository using the command line

It is very useful to update the Versioning Repository from the command line, because you can include the update in a batch script to be run on a schedule. DB2 RE provides a Java interface to the Versioning Repository which allows you to create, update, and prune the Versioning Repository.

Tip: Take care to notice the difference in the command names between the create, update, and prune commands. To create, you use the **`CreateVersionRepository`** command. However, to update or to prune, the commands do not include the `version` string (for example: **`UpdateRepository`**).

There are slight differences in how to invoke the Versioning Repository Java utility on Windows and non-Windows, as follows:

- Windows

  Log in to Windows as a user who has create privileges in the database you want to update. Enter the following update command on one line:

  ```
  db2rewin32 -is:javaconsole UpdateRepository -d database -u userid
  -p password
  ```

  If you omit the user ID or the password, you are prompted for the values.

  In Windows, the status messages open in a new window that is launched when you run the command. The window closes when the command finishes. The messages are also logged in the db2recex.log file. We describe how to configure DB2 RE to use this log file in “The DB2 Recovery Expert log file” on page 47.

- UNIX platforms

  In AIX, Solaris, and Linux, log in as the instance owner to run this command. Enter the following command on one line:

  ```
  db2recex UpdateRepository -d database -u username -p password
  ```

  If you omit the user ID or password, you are prompted for the values.

  If the DB2 RE installation path is not in your $PATH, you must prefix the command with the full path to the db2recex program. For example:

  ```
  /opt/IBM/DB2TOOLS/re/bin/db2recex UpdateRepository -d TRADSIMP -u db2insl1
  -p xxxx
  ```

  Of course, you could also change to the installation path (**cd**) before executing the program.

  The output messages open in the window.

4.2.4 When to update the Versioning Repository

DB2 RE does not provide any facility for scheduling Versioning Repository updates. It is your responsibility to schedule these. You can do this using any scheduling tool (for example, the DB2 Script Center, **cron** on UNIX or Linux, or the Windows Task Scheduler). You use the Versioning Repository to restore dropped objects. So, you should keep the Versioning Repository up-to-date with the current structure of all objects. Ideally, you should run a Versioning Repository update every time you alter an object with DDL.

Restriction: You can use a version of an object in the Versioning Repository to recover a dropped (data) object only if a full offline backup image exists that was taken *after* that version was captured to the Versioning Repository. Thus, whenever you take a backup, always update the Versioning Repository just beforehand. Then, there is a Versioning Repository update that corresponds to the state of the backup.

In order to be useful, you must combine Versioning Repository updates with backups. Because updating the Versioning Repository consumes very little time or resources (especially in stable production environments where DDL is run infrequently), we recommend that you schedule a Versioning Repository update immediately before every backup. If you already have a backup job scheduled, it is trivial to add the Versioning Repository update command just prior to the backup

Note: Because some DB2 RE operations (for example, recovering a dropped table) actually run DDL, you should run a Versioning Repository update (and backup) after executing these DB2 RE operations.

4.2.5 Backing up the Versioning Repository

The Versioning Repository stores its information in a set of tables inside the DB2 database. Currently the Versioning Repository tables are all inside one table space named `ORVRTS`. This table space is created automatically when you create the Versioning Repository. If you take backups at the table space level, make sure to include the Versioning Repository table space in your backup schedule.

For example, if you have two database from a Linux system, `TRADEDB` and `RECART`, you can back them up together in one script, as shown in Example 4-2. For one database, take a full online database backup, and for the other, take offline table space backups and send them to Tivoli Storage Manager.

Example 4-2 Simple backup script before DB2 RE

```
# online backups first
db2 "backup db TRADEDB online to /db2backups"
# do offline tablespace backups for RECART spaces
db2 "backup db RECART tablespace (SYSCATSPACE, USERSPACE1, RECARTTS,
USERPRODUCTTS) use tsm"
```

Now, assume that you install DB2 RE and want to make sure that the Versioning Repository updates are captured in the backups. The new backup script might look similar to that shown in Example 4-3 on page 66. It add calls to the

Versioning Repository update program and adds the Versioning Repository table space, ORVRTS, to the table space backup command. There are many ways you can accomplish this task, these scripts are just simple examples.

Example 4-3 Simple backup script after DB2 RE

```
# Update the RE Versioning Repository for TRADEDB
/opt/IBM/DB2TOOLS/re/bin/db2recex UpdateRepository -d TRADEDB -u db2insl1 -p xx
# online backups first
db2 "backup db TRADEDB online to /db2backups"
# Update the RE Versioning Repository for RECART
/opt/IBM/DB2TOOLS/re/bin/db2recex UpdateRepository -d RECART  -u db2insl1 -p xx
# do offline tablespace backups for RECART spaces
db2 "backup db RECART tablespace (SYSCATSPACE, USERSPACE1, RECARTTS,
USERPRODUCTTS, ORVRTS) use tsm"
```

4.2.6 Pruning the Versioning Repository

Over time your Versioning Repository can grow large with many artifacts from DB2 objects. You might not require or want these artifacts and duplicate versions of DB2 objects. DB2 RE provides a utility to prune the old objects. You can use this utility from the GUI or from the command line.

Pruning the Versioning Repository using the GUI

There are no platform differences in using the GUI to prune the Versioning Repository. (The examples in this section are from a Windows system.) To prune the Version Repository using the GUI, follow these steps:

1. Start the DB2 RE GUI.
2. Connect to your database.
3. From the menu, select **Tools** → **Versioning Repository** → **Prune**, as shown in Figure 4-8.

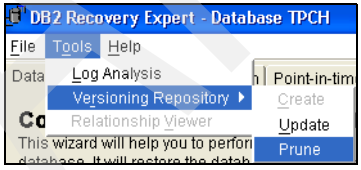


Figure 4-8 Pruning the Versioning Repository using the GUI

A window opens, as shown in Figure 4-9 on page 67, in which you can select the point in the Versioning Repository to which you want to prune.

4. You can either choose a specific date and time or choose to remove all versions prior to the first full database backup. Make your selection and click **Run**.

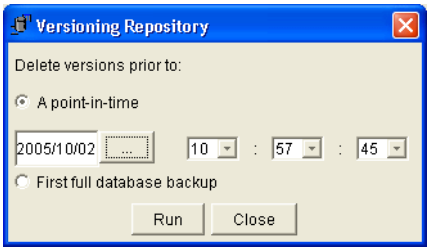


Figure 4-9 Choose Versioning Repository prune point-in-time

DB2 RE prunes the Versioning Repository and displays a message when it completes.

5. Click **OK**.

Note: There is no information logged in the DB2 RE log file about pruning the Versioning Repository.

Pruning the Versioning Repository using the command line

You can also prune the Versioning Repository using the command line. The invocation is the same as for the other Versioning Repository command line options. For pruning, you can also provide the point-in-time value.

Note: There is no information logged in the DB2 RE log file about pruning the Versioning Repository.

The command that you use to prune the Versioning Repository depends upon your operating system, as follows:

- Windows

  Enter the following command on one line:

  ```
  db2rewin32 -is:javaconsole PruneRepository -d dbname -u user -p xxx
  -t timestamp
  ```

  The *timestamp* parameter is optional. If you omit it, the prune removes anything prior to the first full database backup. If you include the *timestamp*, it must be in one of the following formats:

  - `-t YYYY-MM-DD HH.MM.SS for example, -t 2002-12-06 12.04.00`
  - `-t YYYYMMDDHMMSS for example, -t 20021206120400`

- UNIX

  In AIX, Solaris and Linux, log in as the instance owner. Enter the following command on one line:

  ```
  db2recex PruneRepository -d dbname -u user -p xxx -t timestamp
  ```

  The *timestamp* parameter is optional. If you omit it, the prune removes anything prior to the first full database backup. If you include the *timestamp*, it must be in one of the following formats:

  - -t YYYY-MM-DD HH.MM.SS for example, -t 2002-12-06 12.04.00
  - -t YYYYMMDDHMMSS for example, -t 20021206120400

4.2.7 Rebuilding the Versioning Repository

Under most conditions you do not need to rebuild the Versioning Repository. However, you might under the following conditions:

- You applied maintenance to DB2 RE and the readme says to rebuild.
- You are directed to rebuild by IBM support.
- You restored an old database into a new database.
- You had a DB2 RE error situation and cannot access the Versioning Repository any more.

The steps for rebuilding are to first remove the Versioning Repository entirely. Then, just perform the same steps to create it as described in 4.2.2, “Creating the Versioning Repository” on page 56.

Removing the Versioning Repository

The Versioning Repository exists within its own database partition group and table space. To remove the Versioning Repository, just drop the partition group using these steps:

1. Connect to the database
2. Drop database partition group `ORVRNG1`.

CAUTION: When you drop the partition group, all objects under that partition group are also dropped. You should check first to see if any other non-DB2 RE tables were created inadvertently in the DB2 RE table space that are named `ORVRTS`. You should never create user tables in that table space. However, if there are tables in there, you must move them to another location prior to rebuilding the Versioning Repository.

If you are in a situation, such as after applying maintenance, you must perform the rebuild on every database where you use DB2 RE. When the partition group is dropped, you can create the Versioning Repository.

What if I do not rebuild the Versioning Repository?

If you do not rebuild the Versioning Repository in a scenario where you should do it, you might see strange results from DB2 RE. For example, in our lab environment, we applied the DB2 RE fix pack to the server. Later, we restored an old backup of a database that was taken before the fix pack was applied. The Versioning Repository still existed in the backup but was not the correct form that the new code was expecting. When we tried to update the Versioning Repository, we got the following error:

```
SQL0206N "B.COMPRESS" is not valid in the context where it is used.
```

We did not realize right away what was wrong. However, when we did figure it out, it was simple to rebuild the Versioning Repository on the restored database. You should always rebuild the Versioning Repository when you restore a database, just to be safe.

4.2.8 Extracting DDL from the Versioning Repository

DB2 RE provides a way to extract the object DDL from the Versioning Repository, totally outside the scope of a recovery task. This ability might be useful if you only want to see what an object looked like in the past, but you do not need to restore the object.

There is no GUI facility for extracting DDL. To extract DDL from the Versioning Repository, you use the command line. The command is **GenerateDDL**, and there are several parameters, depending on the type of object that you want to extract. We show two examples here. For more details about how to use this feature, see *IBM DB2 Recovery Expert for Multiplatforms: User's Guide*, SC18-9564 at:

http://www.ibm.com/software/data/db2imstools/db2tools-library.html

The syntax for the command varies depending on the platform, as follows:

- Windows

  In this example we extract the DDL for a table space. The command must be on one line, or you can use a script.

  ```
  db2rewin32 GenerateDDL -is:javaconsole -d TPCH -o tablespace -n singlets
  -u db2admin -p xxx -f singlets.ddl
  ```

  In Windows environment, the output file is placed in the directory from which you run the command. The DDL output contains all the DDL for the table space as well as any objects that exist in that table space, such as tables and indexes.

- UNIX

  In AIX, Solaris and Linux platforms, log in as the instance owner to run this command.

  In this example we extract the DDL for one table. The command must be on one line or in a script.

```
db2recex GenerateDDL -d tradsimp -u db2insl1 -p xx -o table
-n MARCIA.KEYGENEJB -t "2005-10-22 15.00.00" -f VR_keygenejbV2.ddl
```

  The output DDL file is placed in the DATA_DIR location. The output DDL file also includes the GRANTs and index create statements for the table.

4.2.9 Versioning Repository usage checklist

When you implement the Versioning Repository in your environment, you can use the following checklist to ensure that you have done everything for each database:

- Verify DB2 configuration settings
- Verify any modifications to the recex.properties file
- Create the initial Versioning Repository in the database
- Back up the database after initial creating the Versioning Repository
- Schedule regular updates of the Versioning Repository just prior to regular backups
- Include the Versioning Repository table space, `ORVRTS`, in backup jobs

4.3 Minilogs

This section describes when and how to use minilogs and whether they should be part of your recovery strategy.

4.3.1 The importance of minilogs

Minilogs are an extra feature of DB2 RE that you can use if you need to speed up the forward recovery of your *critical* tables. Minilogs extracts from the DB2 logs and are created in DB2 RE. The format and processing is the same as when processing DB2 logs. The extracted minilog files are just data files outside of the control of DB2 RE. So, they should be backed up like normal data. If minilogs are lost, you can regenerate them from the DB2 logs if necessary.

The time savings comes from DB2 not having to search through all of the DB2 logs. Instead, DB2 can get the required information from the minilogs. The

benefit is derived from running I/O intensive log reads from critical times to off-peak times. Therefore, the minilogs must be up-to-date for DB2 RE to be able to use them instead of the DB2 logs.

Note: Minilog files should be updated regularly. The frequency, as with any backup strategy, is a trade-off between the cost of running the regular update job and the time that is required to recover those tables in a possible recovery scenario.

4.3.2 Creating and updating minilogs

There is no GUI for the minilog management tool. You invoke the Minilog Management tool by invoking the **db2mm** command. You update and create minilogs using this tool.

The **db2mm** program is installed into the DB2 RE bin library and in Windows. It is set in the system path variable at installation time. Therefore, it can be called from the Windows command prompt.

To run the **db2mm** command on UNIX or Linux, you must use **db2recex** to run the Minilog Management tool by typing the following path before the **db2mm** command:

```
/opt/IBM/DB2Tools/RecoveryExpert/bin/db2recex
```

For example:

```
/opt/IBM/DB2Tools/RecoveryExpert/bin/db2recex db2mm -d sample -o db2inst1
-n department
```

You can use 34 different parameters with the **db2mm** command. You must use some of these in combination and others are mutually exclusive. The simplest use of the minilog command is as follows, which creates the minilog file pair for one table:

```
db2mm -d database_name -n table_name
```

Let us look at the online help text in detail, as displayed by entering **db2mm -h** on the command line, as shown in Example 4-4.

Example 4-4 First part of the online help for minilog management

```
C:\Documents and Settings\Administrator>db2mm -h

DB2 Minilog Management Tool.  Version 2.

 Usage: db2mm -d database [-O nodenum]
 [-b "YYYY/MM/DD HH:MM:SS"] [-e "YYYY/MM/DD HH:MM:SS"]
 [-u username] [-p [password]]
 [-N nodegroup] [-t tablespace] [-o tableowner] [-n tablename]
 [-l tablelist] [-T TID] [-F FID] [-S] [-G {b | f}] [-I] [-P]
 [-m outputpath] [-s syspath] [-L logpaths] [-w workpath]
 [-B backup_base_path] [-W codepage] [-M]
 [-q {r | u}] [-f filename] [-c scope] [-r] [-k] [-v] [-h]
 [-k] [-K] [-Y] [-Z]
```

Table 4-1 shows the descriptions for each of the 34 parameters for the **db2mm** command.

Table 4-1 Online minilog help listing of db2mm parameters

Minilog Parameter	Detailed Description
-b	Specifies the start date and time for minilog generation in the format "YYYY/MM/DD HH:MM:SS". If specified, the existing minilogs will be overwritten.
-B	Specifies base path of backup images for analyzed database. If omitted, the product will try to locate backup images automatically.
-c	Specifies commit scope for generated redo/undo SQL code. If omitted, the commit scope is 1.
-e	Specifies the end date and time for minilog generation in the format "YYYY/MM/DD HH:MM:SS". Must be greater or equal to the value of -b.
-d	Specifies the database name, tables of which would be used for minilog generation. Option can only be omitted if option -I is used.
-f	Specifies file name for generated redo/undo SQL code. If omitted, the SQL code is generated for each minilog separately, placed in the same directory as corresponding minilog and has the same file name as minilog with addition of ".REDO.sql" or ".UNDO.sql".
-F	Specifies the numeric identifier of table, which will be used for minilog generation.

Minilog Parameter	Detailed Description
-G	Specifies direction for masked update reconstruction for SQL code generation. The values are: b - always use rollback f - always use rollforward. Must be used together with option -q. If not specified, the product will choose the best available direction.
-h	Displays help on the product.
-I	Specifies that the product should display information about minilogs existing in the catalog.
-k	Specifies that no new minilogs will be created, only existing minilogs will be updated/rewritten.
-K	Specifies that the product should never read data from backup images. This restriction is related to building SQL code and building system storages. Using this option with option -Y automatically turns off option -P.
-l	Specifies that the minilogs must be built for a list of tables (full table names separated with spaces).
-L	Specifies list of additional paths to log files for database. If omitted, the product will only use path to log files from DB configuration.
-M	Specifies that product collects only minimum of meta information to system storages in order to save disk space and enhance performance.
-m	Specifies path of generated minilog(s). Default path is ./sysstorage/ - See note below.
-n	Specifies the name of the table(s), which will be used for minilog(s) generation.
-N	Specifies the node group, table(s) from which will be used for minilog(s) generation.
-o	Specifies the owner ID for the table(s), which will be used for minilog(s) generation.
-O	Specifies the number of node, operation in which will be used for minilog generation. If omitted, default node number is used.
-p	If contains value, this value is used as a password to connect to DB2. If contains no value, the user is prompted for password to connect to DB2. If omitted, option -u uses empty password to connect to DB2.

Minilog Parameter	Detailed Description
-P	Specifies that log records should be prepared before they get written into minilog. Preparation means reconstruction of data details for masked update records.
-q	Specifies that the SQL generation should be performed for minilogs satisfying the set of filters. During SQL generation, minilog set is not being updated in any way. The values are r - Generate redo SQL code. u - Generate undo SQL code. Values cannot be combined.
-r	Specifies that minilog(s) must be re-generated, not updated.
-s	Specifies base path of system storages created and used by product. Default path is ./sysstorage/
-S	Specifies that the minilogs for system tables are not built.
-t	Specifies the table space, table(s) from which will be used for minilog(s) generation.
-T	Specifies the numeric identifier of table space, tables from which will be used for minilog generation.
-u	Specifies the user ID to establish connection to database. If omitted, the product uses the ID of currently logged on user.
-v	Displays version of the product.
-w	Specifies paths to intermediate work files for a session. Default work path is current directory.
-W	Specifies the code page for text output in ICU format. Default code page is console codepage of the operating system.
-Y	Specifies that the product should never read data from database table page files. This restriction is related to SQL code and building system storages. Using this option with option -K automatically turns off option -P.
-Z	Specifies that product should erase and rebuild system storages.

4.3.3 For which objects should I create minilogs

The whole point of creating minilogs is so that DB2 RE will choose them in the recovery path as the alternative to reading some of the much larger DB2 logs. This would show in the DB2 RE GUI as a cheaper recovery path option. The best way of ensuring that a table's minilogs become candidates for DB2 RE is to choose a table which has a small number of infrequent updates. A small number of updates would mean the resulting minilog file will be much smaller than the source log file. The candidacy of using minilogs in the recovery path is also improved by keeping them up-to-date so that there is less chance of DB2 RE having to also use real logs in that table's recovery.

4.3.4 When to update minilogs

If you have a regular but infrequent job which updates your minilog candidate tables, then you should schedule the minilog update after that job. Another ideal time would be just after a weekly maintenance update of the dimension table values. If you attempt to create minilogs for a table which has not had any update activity logged, no minilogs are created, and you see the following message:

```
C:\RE\Minilogs>db2mm -d sample -n org
There is no DB2 log activity to add to minilog(s) for specified options.
```

4.3.5 When to use minilogs

You use minilog to speed up the recovery process. You use the minilog management tool, **db2mm**, to specify which tables are the most important. Then, the log records for those chosen tables are extracted. So, you should use minilogs for tables that are critical in so far as losing these tables would cause disruption to your business if any one of the tables was individually unavailable.

A common question is "How should we handle minilogs in our backup strategy?" The minilogs are just data files. So, you can back up minilogs as is usual for any other data. However, you can always regenerate the minilogs from the real DB2 logs if necessary.

4.3.6 Creating minilogs for a single table

Example 4-5 is a simple example on Windows that specifies which database and table to create the minilogs. The defaults are taken for the rest of the parameters.

Example 4-5 Create minilog for table DEPARTMENT

```
C:\RE\Minilogs>db2mm -d sample -o db2admin -n department
New table DB2ADMIN.DEPARTMENT (16.2) found. Minilog was created.
```

To run this command on UNIX or Linux, you need to prefix the command with **db2recex** as follows:

```
/opt/IBM/DB2Tools/RecoveryExpert/bin/db2recex db2mm -d sample -o db2inst1 -n
department
```

When you run the **db2mm** command for a table, two files are created: an mlg file and an mlx file. In this example the files are called:

- DB2ADMIN.DEPARTMENT(16.2).mlg
- DB2ADMIN.DEPARTMENT(16.2).mlx

The pair of numbers in the file names refer to the table space identifier and the table file identifier.

When no path is specified for the minilogs, that the -m parameter is omitted, on Windows the files are created relative to the current directory, for example:

```
C:\RE\Minilogs\sysstorage\SAMPLE\NODE0000\IBMDEFAULTGROUP\MINIDEM0
```

where MINIDEM0 is the table space name.

The -m parameter specifies where the minilogs are written. However, when recovering using the GUI, you must ensure that this path is known to DB2 RE.

If there are no updates that minilogs can make, either because there have been no log updates or because you provided an incorrect table or owner name, then you see the following message:

```
There is no DB2 log activity to add to minilog(s) for specified options.
```

4.3.7 Creating minilogs for all the tables in a table space

If you have system or dimension type tables that are critical to your business, that are updated infrequently, and that all reside in a single table space, you have a good candidate for creating minilogs. You can run the minilog command against a table space to extract all the log data for each table, as shown in Example 4-6.

Example 4-6 Run minilog command against a table space

```
C:\RE\Minilogs>db2mm -d SAMPRE -t RELAY
Minilog for table DB2ADMIN.RACE (3.3) is updated.

Minilog for table DB2ADMIN.SHOP (3.4) is updated.

New table DB2ADMIN.VAN1 (3.2) found. Minilog was created.
```

In Example 4-6 on page 76, the minilogs for tables RACE and SHOP already exist but are updated with the latest log records. New minilogs are created for table VAN1, either because **db2mm** was not run against this table before or because there was no update activity. If there is a fourth table in the RELAY table space without any update activity, no message or minilogs are produced.

4.3.8 Directory path for minilogs

If you do not specify the -m path file parameter on the **db2mm** command, the minilogs are written to following path:

{RE_Install_Path}{/}data{/}DB2_Instance_Name{/}minilogs{/}database_name \NODEnnnn\nodegroup\tablespace

Check that this is the path coded in the recex.properties file in the DB2 RE conf directory, as follows:

minilogs.path={DATA_DIR}{/}minilogs

The minilogs are written to a directory which is determined when the **db2mm** command is run. If you do not code the -m path parameter, the path in the properties file is used. Subsequently, when the DB2 RE is determining the potential recover paths driven from the GUI, the same path should be used to read the minilogs, as is coded in the properties file.

Note: The -m parameter directs where the minilogs are created. We advise that you do *not* use this parameter. Instead, accept the default path. Otherwise, the DB2 RE GUI will not be able to find the minilogs unless the path minilog path is correctly set in the recex.properties file.

4.4 Archive Log Inventory Data (DB2 RE metadata)

This section discusses some key concepts of the Archive Log Inventory Data (ALID) and how to manage it.

4.4.1 Overview of ALID or metadata

The ALID contains information about transactions in log files, object name to DB2 internal identification mapping information, and details about operations such as dropped objects. Usually, the information is required when running log analysis, and it is always required when performing a recovery.

The ALID is created from information found in the logs. It is used by subsequent commands to report information and simplify access to the logs for retrieving

additional details in the future. An updated ALID is good for log analysis performance because it has key information that is readily available. DB2 RE reads only a limited number of log files to satisfy the log analysis request for the specified time period.

4.4.2 Creating and updating the ALID

You can create or update the ALID using either the DB2 RE GUI or the **db21a** command line tool. For **db21a**, the ALID is updated or created whenever it is run without the **-A** or **-P** options. For the DB2 RE GUI, selecting Update Archive Log Inventory Data is the equivalent of not specifying the **-A** option, as shown in Figure 4-10.

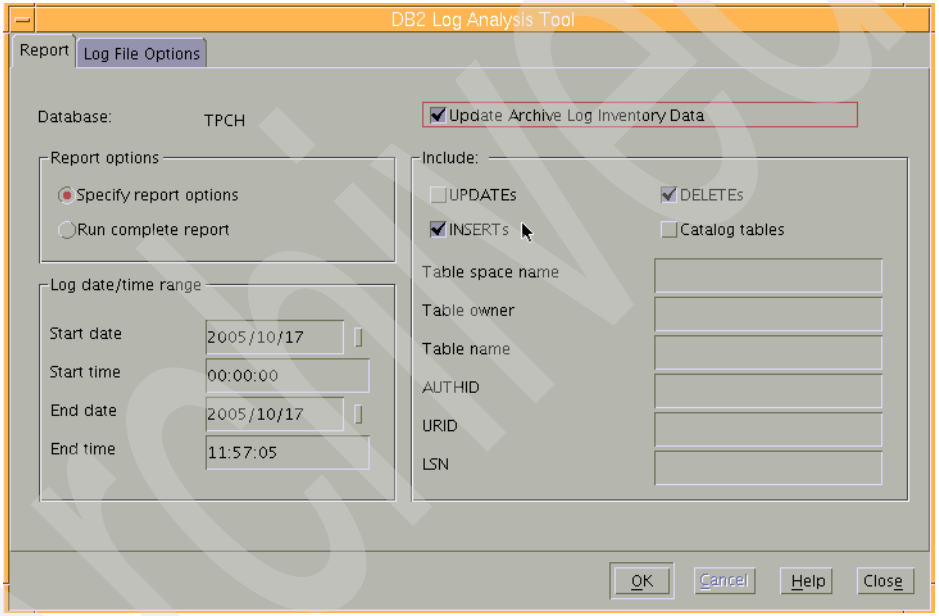


Figure 4-10 Log Analysis Tool

Under the Log File Options tab, selecting the All logs options is the equivalent of not specifying the `-P` option with the **db21a** command, as shown in Figure 4-11. If you choose any other option here, you are in effect using the `-P` option because all logs are not analyzed.

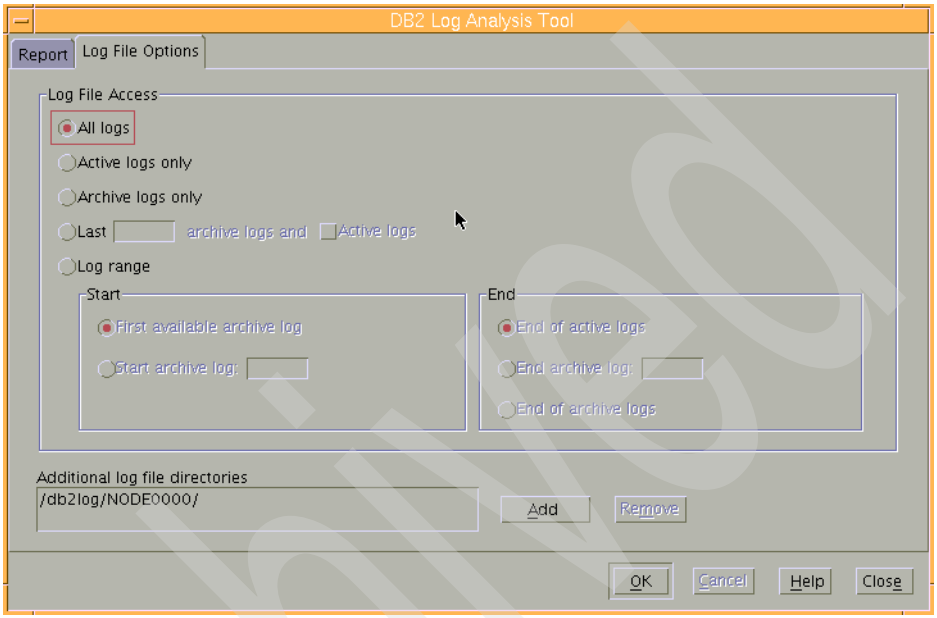


Figure 4-11 Log File Options

In summary, to update the ALID from DB2 RE GUI, the following two conditions should be met:

- Update Archive Log Inventory Data is selected as in shown Figure 4-10 on page 78.
- All logs is selected as shown in Figure 4-11.

If either of these conditions is not met then the ALID is not updated. If you meet the first requirement but not the second one, notice that the Update Archive Log Inventory Data option is not available, as shown in Figure 4-12 on page 80.

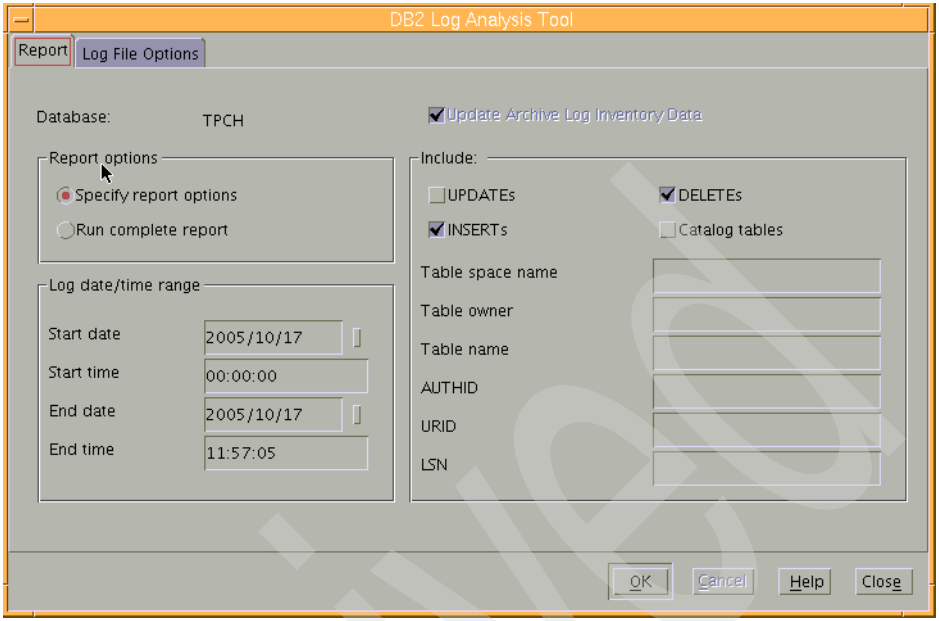


Figure 4-12 Update ALID option greyed out

Using the startlog parameter

The ALID update starts from the log number that is defined with the startlog parameter in the recex.properties file, as follows:

```
startlog.<dbname>.<db2 node number>=<log number>
```

(For information about how to configure the recex.properties file, see 3.4.1, “Customizing the recex.properties file” on page 34.) If the startlog parameter is not specified, then the ALID update starts from first available log file. When you update the ALID, it starts from where it left off. Thus, the ALID updates are incremental. The time that is required to complete an ALID update is proportional to the number of logs that need to be processed. If you do not update the ALID for a very long time and if a lot of logs have accumulated, then the ALID update might take some time.

Choosing a startlog value

You need to ensure that the chosen log number for the database is suitably far enough back in time to include the desired transactions as well as additional historical information. When using the startlog parameter, the idea is to avoid having DB2 RE start with `log 0` or the `first available archive log`, because that might be very far in the past and can result in unnecessary processing. However, you need to pick a value that goes back far enough. Generally, the

starting log should at least include the time when the most recent full backup was performed so it will be available to the log analysis command when it needs to perform reconstruction for UPDATEs.

4.4.3 Rebuilding the ALID

You can rebuild the ALID from scratch only using the **db2la** command line tool as follows:

```
db2recex db2la -d vps -b <begin timestamp> -e <end timestamp> -Z
```

For Windows system, **db2recex** is not required.

We do not recommend rebuilding the ALID very frequently. Typically, you only want to rebuild the ALID if it becomes inconsistent for some reason.

If you run **db2la -Z** from the command line, always specify a small, recent begin or end time range (the -b and -e parameters). If your goal is just to create the ALID initially, then pick a small begin or end window of only a few minutes, when you know there was not much activity. You do not care about the report output, but you want to limit the report phase processing as much as possible.

Important: Begin and end time stamps with log analysis only play a role with SQL report generation. They do not relate to the ALID creation and update.

Currently, you cannot re-create the ALID without running log analysis. Executing log analysis leads to creating a general report. If you leave off the -b and -e options, then log analysis tries to generate a report for the entire life of the available logs or starting from the startlog parameter, if specified, which is probably not what you want. This type of log analysis will take a long time. After the ALID is created and you want to run a real log analysis report, you can specify a begin and end time as needed.

Note: If you change the startlog value, then you need to re-create the ALID from scratch for affected database.

5

Log analysis

In DB2 UDB, all transactions (inserts, deletes, and updates) against a database are logged in transaction log files. These log files are needed for crash recovery, roll forward, and replication. Log analysis extracts transaction information in log files and then formats that information in a readable format. You can also decide to undo or redo these changes. Log analysis works with both active and archived logs.

Note: In order to use log analysis, the `LOGRETAIN` database configuration parameter must be set to RECOVERY. For information about how to check or change database configuration parameters, see 3.2.1, “DB2 configuration requirements” on page 21.

This chapter explains the concepts of log analysis and how to include log analysis in your recovery strategy. It includes the following sections:

- 5.1, “Log analysis for auditing” on page 84
- 5.2, “Log analysis for undoing or backing out table changes” on page 92
- 5.3, “Log analysis using the command line” on page 99
- 5.4, “Performing remote log analysis” on page 101
- 5.5, “Report types and interaction of time and log options” on page 109
- 5.6, “Log analysis in a DPF environment” on page 110
- 5.7, “Considerations and recommendations” on page 113

5.1 Log analysis for auditing

Log analysis is useful when you need to see what type of transactional activity is taking place in your database. In native DB2, there is no functionality available to format the logs and to generate SQL from them. Here are two typical situations where log file auditing can be useful:

- You noticed that there was a spike in log space usage on a certain day and you need to determine what transactions are consuming the extra log space.
- You find some missing, extra, or modified rows in a table and you need to investigate when the changes were made and by which AUTHID.

5.1.1 Prerequisites for log analysis for auditing

By default, the DB2 logger does not capture the AUTHID in transaction logs. There are two ways to enable capturing of AUTHID in log files:

- You can set the following registry variable:

  ```
  db2set DB2_LOGGING_DETAIL=AUTHID
  ```

  Then, you can recycle the DB2 instance (**db2stop/db2start**). Any transactions that follow this include AUTHID information as well. This method is the preferred method of capturing AUTHID because it covers all tables in the database and there is minimal logging overhead.

  Note: AUTHID must be specified. It should be all capital letters in the **db2set** command.

- You can enable Data Capture Changes for the tables of interest. This is one of the options in the **Alter table** command that is used primarily for replication. For a detailed discussion about this topic, see 5.7.2, “Data capture changes” on page 114.

To use log analysis for auditing, you should enable AUTHID capturing as early as possible.

Application ID versus AUTHID

Note that Application ID is *not* the same as AUTHID. Example 5-1 illustrates this point.

Example 5-1 The DB2 list applications

```
Auth Id  Application  Appl.    Application Id                 DB      # of
         Name         Handle                                 Name    Agents
-------- ------------ -------- ------------------------------ ------- ------
DB2INSK1 db2bp        10       *LOCAL.db2insk1.050926220029   VPS     1
DB2INSK1 java         9        *LOCAL.db2insk1.050926215700   VPS     1
DB2INSK1 java         8        *LOCAL.db2insk1.050926215659   VPS     1
```

You can have multiple applications that connect to the database with same AUTHID. For example, this connection is typical when you use connection pooling from a WebSphere® application server, where all the connections are made using the user ID that is specified in the data source.

5.1.2 Analyzing changes to a table, a sample scenario

In this scenario, we notice that we cannot find any blue 1973 Boran model cars in the CAR table since 27 September. (The DDL for all the tables that we use in this scenario is available in B.4, “DDL of VPS.CAR table” on page 373.) In this case, we can use DB2 Recovery Expert (DB2 RE) to analyze changes to CAR table.

To run a log analysis, follow these steps:

1. Launch DB2 RE and connect to the database. The DB2 RE main window opens, as shown in Figure 5-1 on page 86.

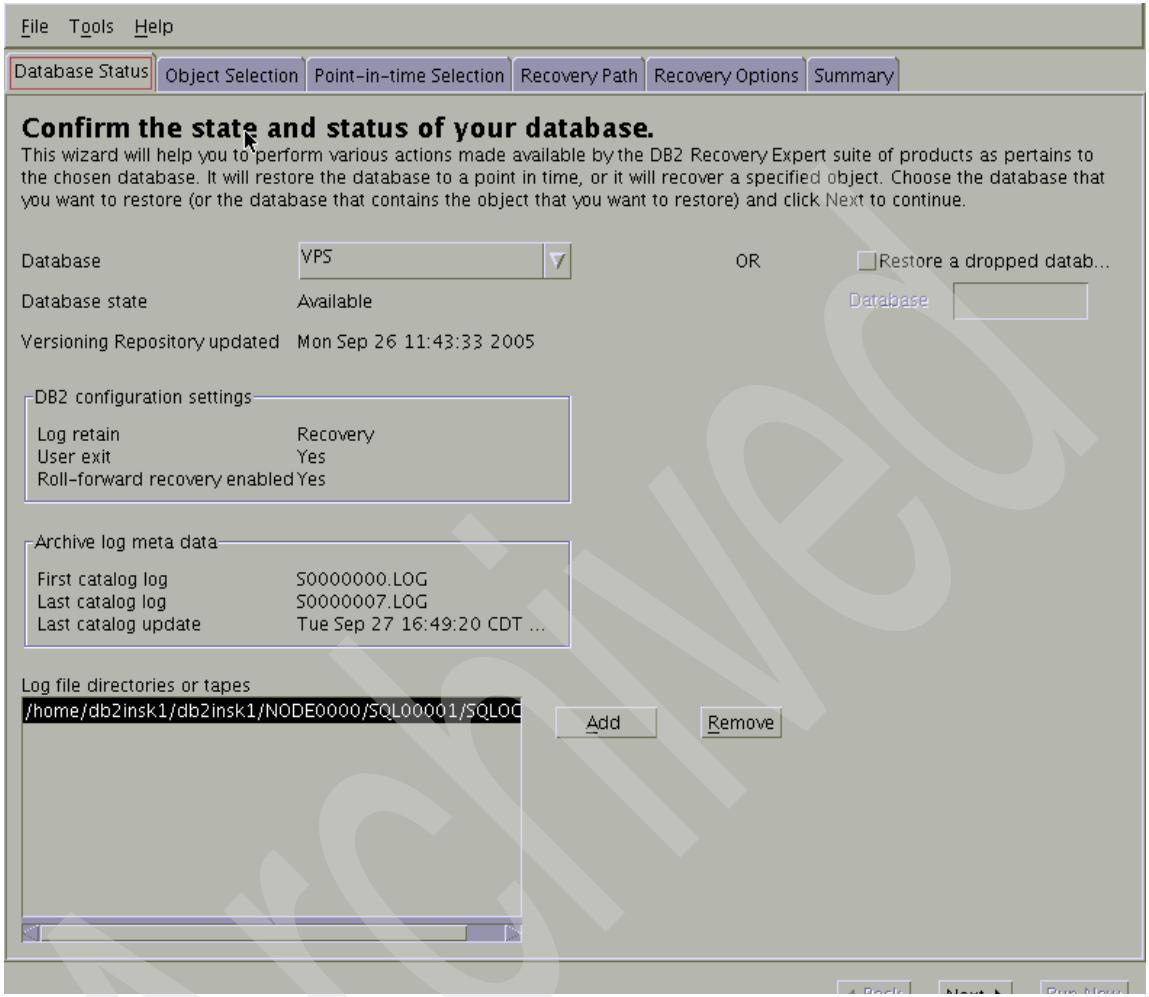


Figure 5-1 DB2 RE main window

2. Select **Tools** → **Log Analysis**, as shown in Figure 5-2.

Figure 5-2 Selecting Log Analysis

3. Go to the Report tab and make the selections as shown in Figure 5-3.

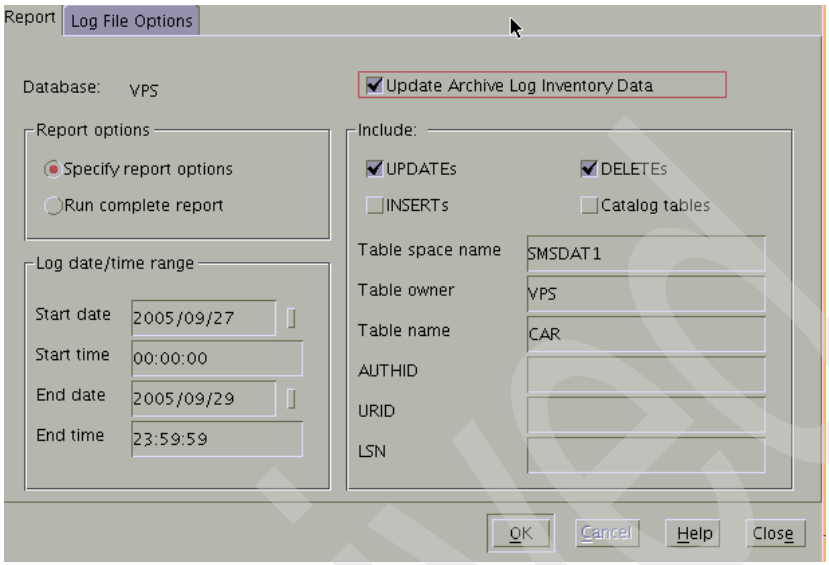


Figure 5-3 Entering information in the Report tab

In the Report tab, you have the following options:

– Specify report options

  Selecting this option runs the log analysis report based on certain filters such as table name and table space name that are specified on rest of the window.

– Run complete report

  Selecting this option runs a complete report without any filters for all objects starting from transaction log number 0. You can control the starting log number by configuring the startlog parameter (see “Using the startlog parameter” on page 80). With this parameter, you can tell log analysis to start from a recently archived log file.

– URID and LSN

  URID is another name for *Unit of work ID*. LSN is the *log sequence number*. A given transaction ID can have multiple LSNs in it. For example, the following would delete all the rows in the CAR table:

```
db2 delete from vps.car
```

  This delete operation would have one URID and each deleted row would have a corresponding LSN. DB2 uses these numbers to manage transactional activity.

– Update Archive Log Inventory Data

You get two different types of reports depending on whether or not you choose this option. If this option is not selected, then a quick report is run to give you a draft version of the changes. This option mostly works fine but can give you inaccurate results. For example, if a bunch of tables are dropped and you run log analysis without updating the Archive Log Inventory Data (ALID), then the log analysis only looks at the DB2 catalog information where it will not find these tables because they have been dropped. Therefore, log analysis reports all transactions against these tables lumped together under `UNKOWN` table name. When you choose to update the ALID, log analysis determines the actual table names and lists transactions for each table separately.

Tip: We recommended that you update the ALID every time you run log analysis. You can do this by selecting Update Log Inventory Data on the Report tab. You must choose this option if you plan on undoing or redoing transactions.

4. Click **OK**.
5. Go to the Log File Options tab, as shown in Figure 5-4.

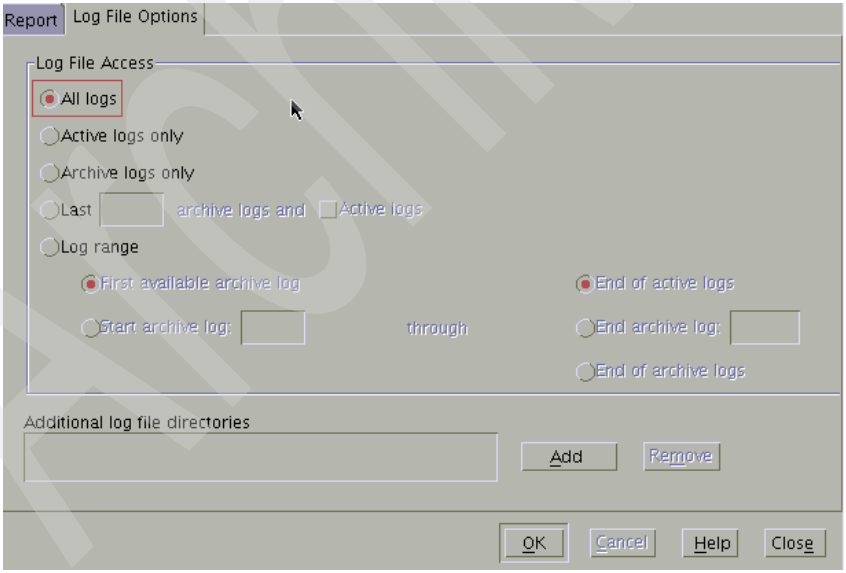


Figure 5-4 The Log Analysis Log File Options tab

Tip: We recommend that you always choose the All Logs option unless you have a good reason for specifying only a subset of logs.

There are a lot of options that are provided regarding where you want to go to look for log files. In this scenario, we are not sure how many log files might span for the delete or update activity, so we select **All Logs**.

Attention: If you do not select the All Logs option , then the ALID is not updated, regardless of the selection you made on previous window.

If you make a selection other than the All Logs option, then you are telling it to work with a specific range of logs. For the implications of making a selection other than the All Logs option, see 5.5.2, “Time specifications and Log file options and precedence” on page 109.

6. Click **OK**.

   A window opens showing that 43 deletes have occurred against the VPS.CAR table (Figure 5-5). This is the general report that has been produced by Log Analysis.

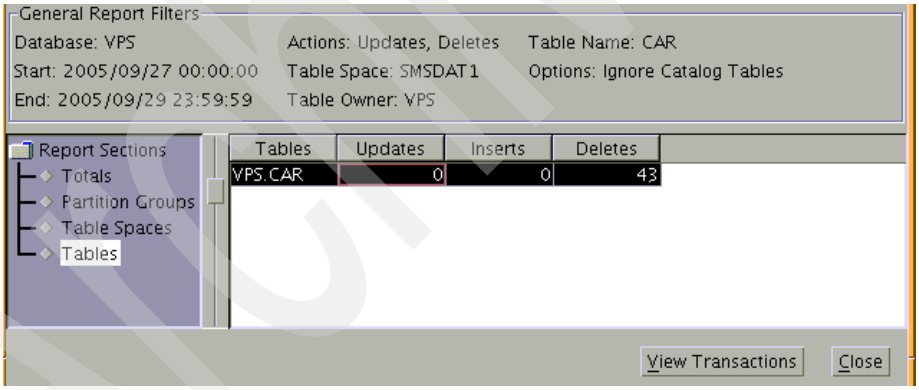


Figure 5-5 Results window showing deletes in table VPS.CAR

7. Double-click the table entry to see more details (Figure 5-6). We can see that deletes were issued by AUTHID DB2INSK1. So, now we know that the DB2INSK1 AUTHID deleted the missing 43 rows from the VPS.CAR table.

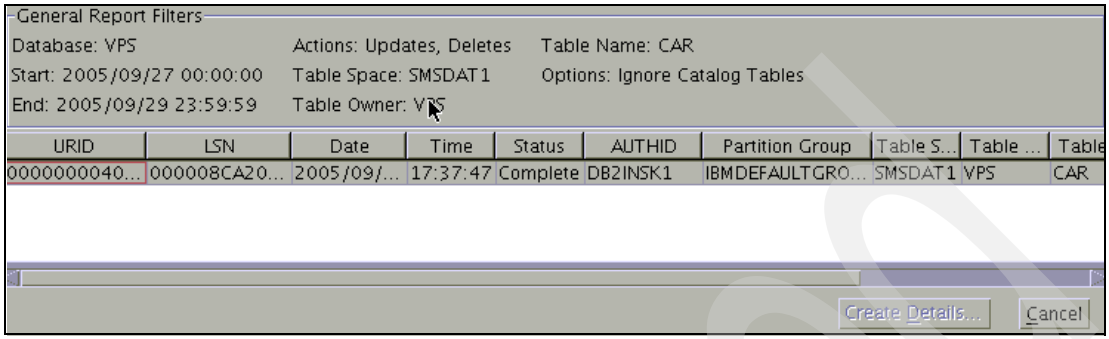


Figure 5-6 Delete details

8. We want to see exactly what was deleted. So, double-click the entry to get to the next window (Figure 5-7).

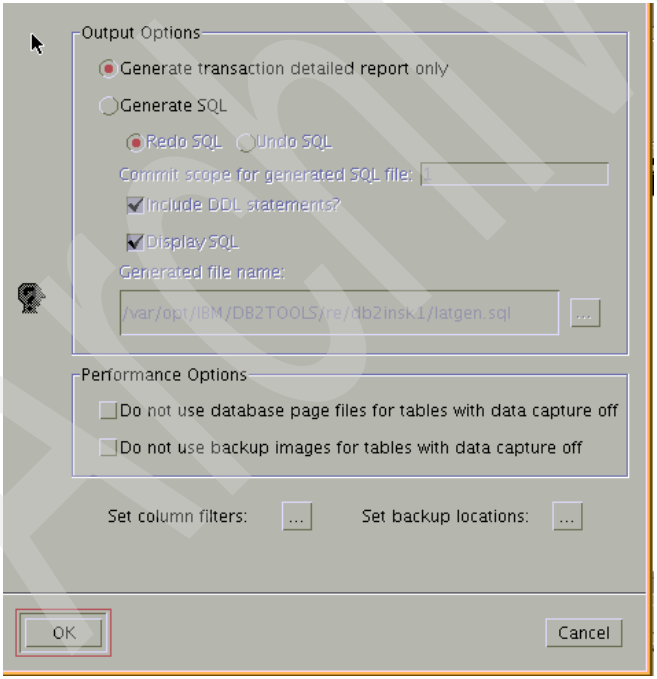


Figure 5-7 Output Options: Generate transaction detailed report only

9. Make the selection for generating the report (Figure 5-8).

   You can see the deleted rows show that the Blue 1973 model Borans were deleted. You can also notice that the URID or UOW ID is the same for all deletes, meaning that this delete was done within one Unit of Work. There is one LSN per deleted row.

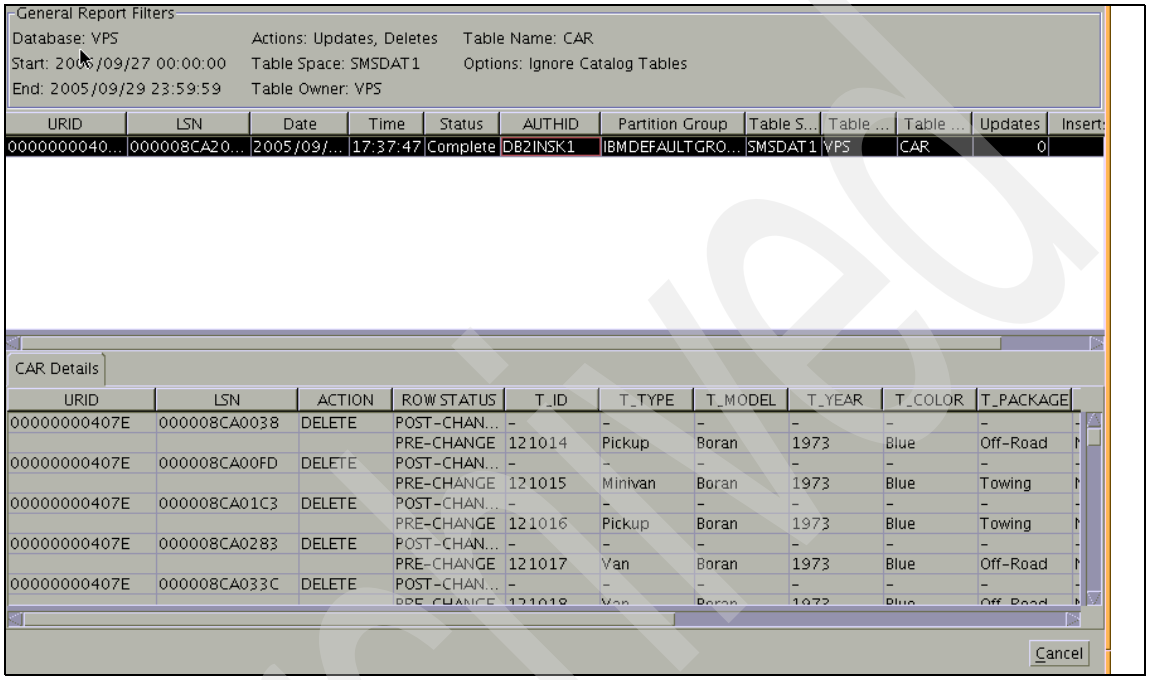


Figure 5-8 Detailed report for delete activity

5.2 Log analysis for undoing or backing out table changes

You can also use log analysis to back out or undo changes to a table. This ability is very helpful in a large environment because you do not have to restore and roll forward an entire table space or database to undo a transaction (INSERTs, UPDATEs, and DELETEs).

5.2.1 Scenario description

In this scenario, we build on our example in 5.1, “Log analysis for auditing” on page 84. In that section, we stopped at the point of just finding out the deleted transactions and analyzing them. In this section, we back out those transactions.

5.2.2 Prerequisite for undoing or backing out SQL

We want to make sure that while we are undoing transactions, no other user is updating the same rows. In our example, it is possible that some user inserts the rows back into the table between the time we run the log analysis to generate undo SQL and the time we actually undo the deletes. This action could either lead to duplicate rows or to undo from DB2 RE failing if there is an unique index violation.

We can use **REVOKE** command to revoke privileges to certain AUTHIDs such that users cannot update the table. (For information about the **REVOKE** command, see *SQL Reference, Volume 1, V8*, SC09-4844.)

We recommend that to be absolutely sure that no one makes changes to the table, you quiesce the table space using the following command:

```
DB2 QUIESCE TABLESPACES FOR TABLE <table> INTENT TO UPDATE
```

This command prevents access to the table space across connections by any user ID other than the one that did the quiesce. You should issue this command *before* you generate SQL.

5.2.3 Undoing or backing out INSERTs or DELETEs

The process for undoing INSERTs or DELETEs is similar because an entire deleted or inserted row is logged in the transaction log files. Undoing or redoing updates involves reconstructing the update statement because the entire updated row is not logged. For more information, see 5.7.4, “Masked update reconstruction” on page 115.

So, for this scenario, follow these steps, picking up from Figure 5-7 on page 90:

1. Instead of choosing the Generate transaction detailed report only option, choose **Generate SQL**, as shown in Figure 5-9, to create a detailed report that is based on the general report that was produced previously.

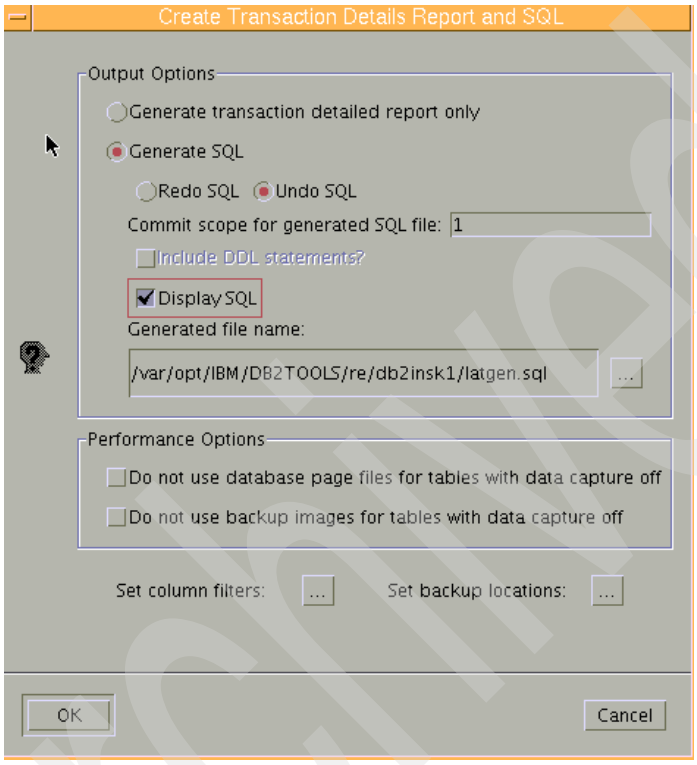


Figure 5-9 Undo Selection window

2. Select **Undo SQL**. Note that the commit scope for generated SQL file option is set to 1, which means to place a `commit` statement after every transaction in the generated SQL file. You can specify a larger number to commit transactions in batches.

> **Tip:** If you are undoing a large number of transactions, we recommended that you increase the commit scope from the default of 1. Committing a batch of transactions gives better logging performance compared to committing after every transaction.

In this scenario, we also select **Display SQL** because we want to see how the deletes are undone.

3. Click **OK**. The Detailed Report window opens, as shown in Figure 5-10.

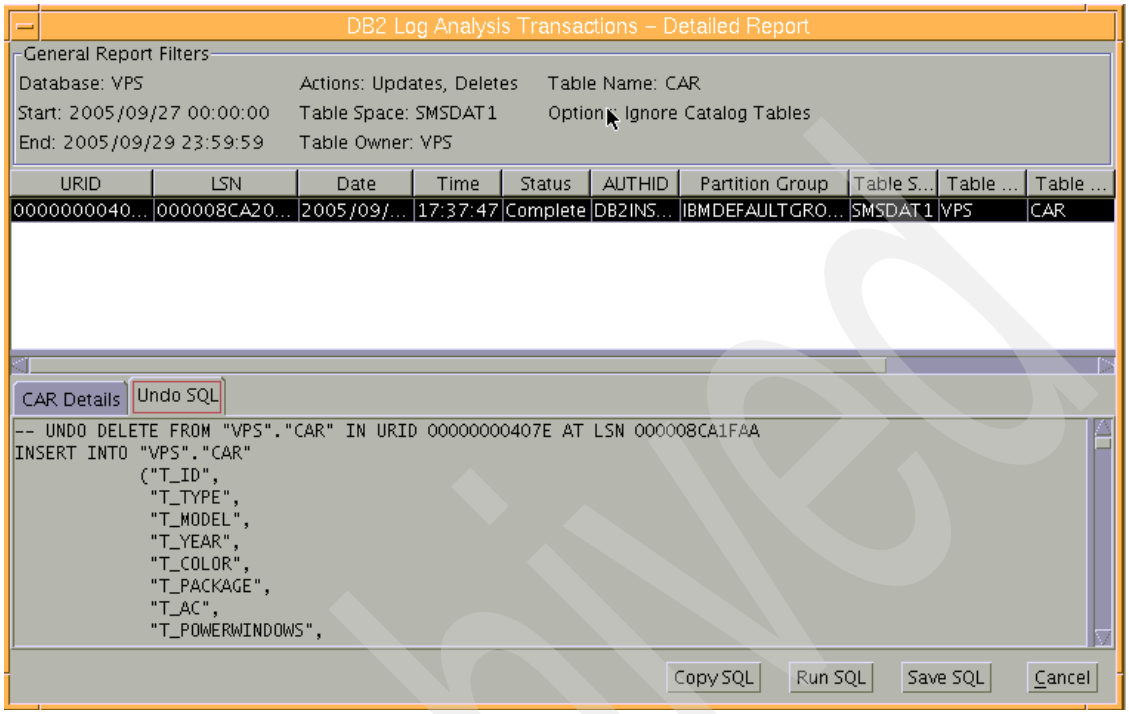


Figure 5-10 Undo report detail

4. This window shows the undo SQL that was generated in addition to details of the actual delete operation under CAR Details. You can click **Save SQL** to save the SQL statements to a file for your records. For this scenario, we click **Run SQL** to execute the INSERT or undo SQL. After the undo statements ran, the window shown in Figure 5-11 opens, indicating a successful operation.

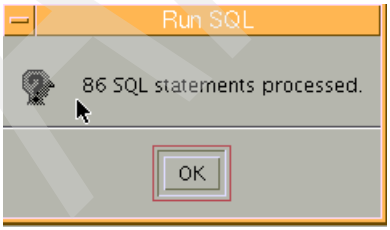


Figure 5-11 Undo result

Note that it executed 86 statements when we had 43 inserts. There is a `commit` statement for every insert, so this is why we have a total of 86 statements. Now, we have back our missing rows.

Note: The only place where the commit scope that is specified in the commit scope for generated SQL file option takes effect is on the original output file that contains all of the generated SQL. If you run SQL from the DB2 RE GUI, the commit scope is always 1, regardless of what is specified in this option. If you choose to Save SQL and execute it later, you can edit this option as desired.

5.2.4 Undoing or backing out updates

This section discusses how to back out updates. For this scenario, a user has updated all 1973 Boran *blue* cars to *red* by mistake, and now we need to back out these changes.

Using DB2 Recovery Expert to back out updates

The initial steps to back out update are the same as those that are described in 5.1.2, “Analyzing changes to a table, a sample scenario” on page 85 until we get to the window shown in Figure 5-5 on page 89. In the update case, we see the window that is shown in Figure 5-12.

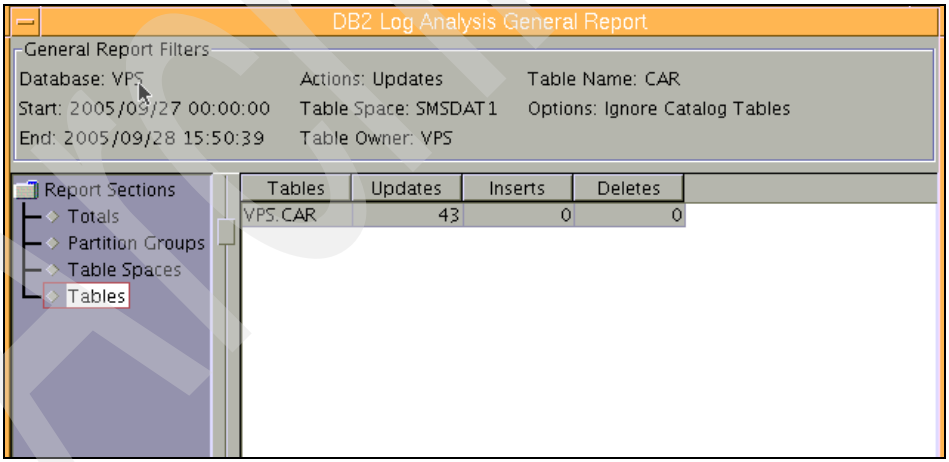


Figure 5-12 Window showing update activity

Then, follow these steps:

1. Drill down to see details of this update, as shown in Figure 5-13.

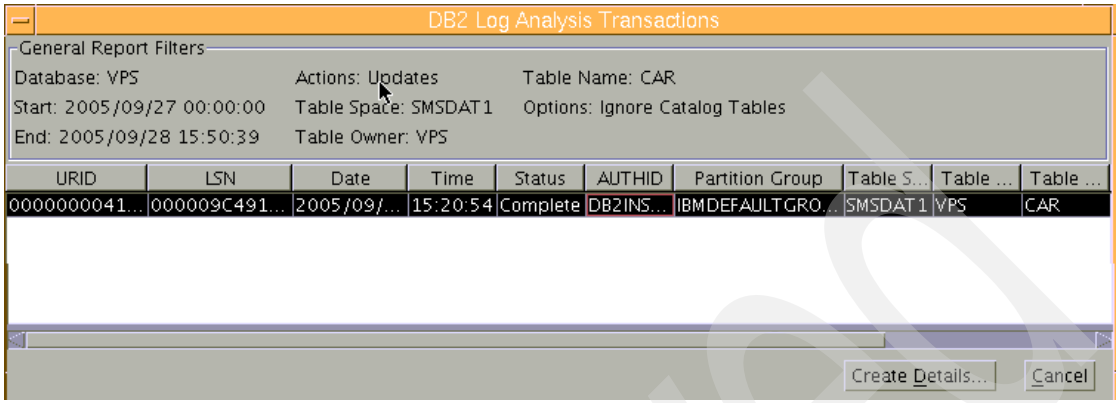


Figure 5-13 Update undo details

2. Double-click this entry to move to the next step, as shown in Figure 5-14.

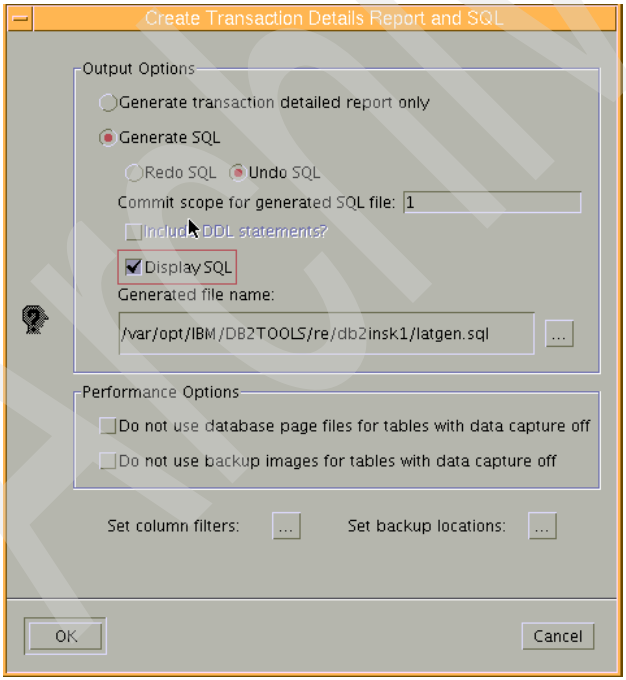


Figure 5-14 Update undo options

In this scenario, Performance Options come into play. As described in 5.7.2, “Data capture changes” on page 114, if the DATA CAPTURE CHANGES option is enabled for a table, then we have the pre-update and post-update image of a row available in the log files. Otherwise, DB2 RE needs to construct the update statement along with corresponding undo update statement. This reconstruction is called *Masked Update Reconstruction*. There are two types of update reconstruction:

- Backward update reconstruction
- Forward update reconstruction

For more information, see 5.7.4, “Masked update reconstruction” on page 115.

3. Click **OK**. The Detailed Report window opens, as shown in Figure 5-15.

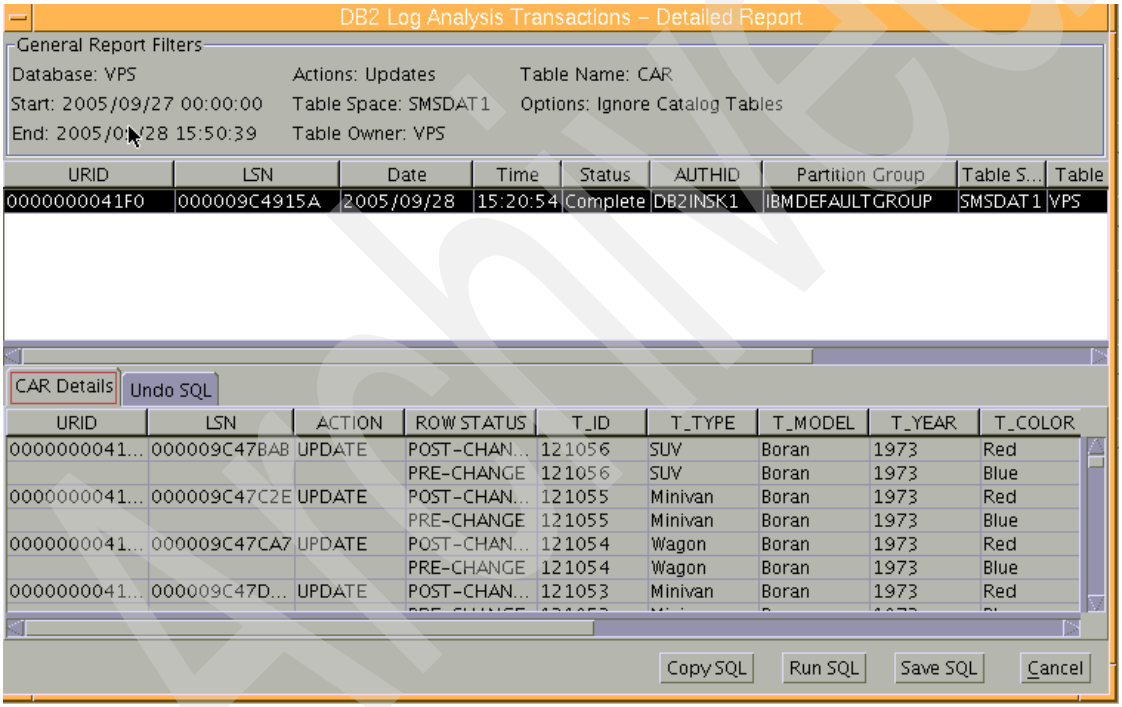


Figure 5-15 Pre update and post update details

Under the CAR Details tab, we can see the pre-update and post-update images. We can see the that *T_COLOR* was changed to *red* from *blue*. Under the Undo SQL tab, we can see the undo statements created by DB2 RE (as shown in Figure 5-16).

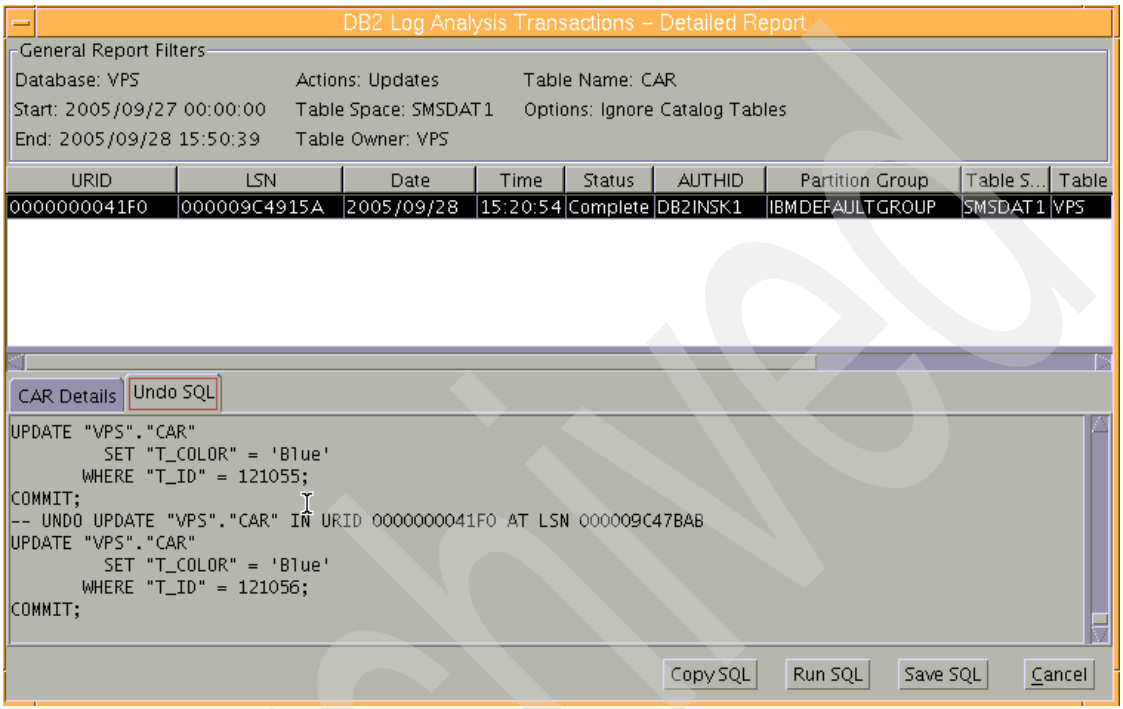


Figure 5-16 Undo statements Created by DB2 RE

Normally, DB2 RE would create update statements such that they include all column values in the predicate to fully qualify each row that it is going to update. However, notice that in our case, there is only one predicate with the clause `T_ID`. DB2 RE detected that there is a primary key on the `T_ID` column, so it needs to create a predicate on the `T_ID` column to make sure it is updating the correct row.

Tip: We recommend to create a primary key on a table whenever possible. DB2 RE can generate redo or undo SQL statements faster for tables with primary keys.

4. Click **Run SQL**. It shows that 86 statements were processed, as shown in Figure 5-17.

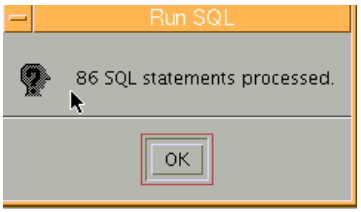


Figure 5-17 Run Undo SQL results

Again, we have 43 statements for undo and 43 for commits. Now, we have back our blue 1973 Borans.

5.3 Log analysis using the command line

The command for doing log analysis is **db2la**. In this scenario, we delete all 1973 Boran model cars. Here is the delete statement:

```
delete from vps.car where t_year=1973 and t_model='Boran'
```

Now we run a general report to see all changes to VPS database on 29 September using the following command:

```
db2recex db2la -d vps -b "2005/09/29" -e "2005/09/30"
```

Note: On UNIX platforms, you should add **db2recex** in front of **db2la**. On Windows, you can run **db2la** by itself.

This command shows that 300 deletes have been done against table VPS.CAR. (For a sample of this report, see B.1, “Excerpt from general report created by db2la” on page 366.) The options used in the examples are:

-b Start date and time for log analysis
-e Stop date and time for log analysis

In 5.2.2, “Prerequisite for undoing or backing out SQL” on page 92, we discussed that you need to make sure that no one is updating the table between the time you generate report and the time when you execute the undo SQL. With **db2la**, we have one more method of doing this. From our session we issue the following command:

```
export DB2OPTIONS=+c
```

This command turns off auto commit for our session. We can verify that auto commit is off with the following:

```
db2 list command options
```

We can lock the table in exclusive mode, and it stays locked until a commit is issued:

```
db2 lock table vps.car in exclusive mode
```

Now, we want to determine more detail about the these deletes. So, we run the following command:

```
db2recex db2la -d vps -b "2005/09/29" -e "2005/09/30"  -D -n CAR -c 300 -q u
-f undo.sql
```

This command produces a detailed report to the standard output (window) and saves the undo SQL in the undo.sql file. (An excerpt from the detailed report is available for your reference in B.2, “Excerpt of detailed report created by db2la” on page 368. A sample of undo SQL that is created is available in B.3, “Excerpt of the undo SQL” on page 370.)

The options that we use in this example are:

-c specifies the commit scope.

-D Generates detailed report.

-q Specifies SQL generation should be performed while detailed report is being generated. The following are option of the **-q** option:

- **-r** Generate redo SQL
- **-u** Generate undo SQL

-f Output file for generated undo SQL.

We specified commit scope of 300, which means that it will only place a commit after 300 statements. This is the number of deletes that we had. We do not want to issue a commit earlier, because we would lose our exclusive lock on the table. We can now review and execute **undo.sql** to get back our deleted rows, as follows:

```
db2 -tvf undo.sql
```

To turn auto commit back on, issue the following command:

```
export DB2OPTIONS=-c
```

Note: The **db2la** command updates the ALID every time it runs. You can specify the **-A** or **-P** options if you do not want to update the ALID.

5.3.1 Which interface is better

You can make the decision of whether to use the graphical log analysis tool or the command line **db21a** tool based on your personal preference. However, each interface has its advantages and disadvantages.

The advantage of the graphical interface is that it is easy to use. However, its disadvantage is that you can only generate undo or redo SQL per transaction or unit of work and per table.

The advantage of the command line interface is that if you create a general report based on a time period you can then produce undo or redo SQL for all transactions on one table, a list of tables, or one table space in that time period all in one file. With **db21a**, there is also the option of using LOCK TABLE when undoing SQL.

5.4 Performing remote log analysis

With DB2 Recovery Expert V2, there is new functionality that allows you to perform log analysis remotely. Remote log analysis is very useful when your production database server is used heavily and when you do not wish to put more load on it by installing new tools and applications. You can run reports remotely and create undo and redo SQL from a remote system. You can then run this SQL on your production server, thereby minimizing the amount of processing that has to be done on production server.

5.4.1 Prerequisites for remote log analysis

To do remote log analysis, the following prerequisites must be met:

- The computer where log analysis will be performed and DB2 database server computer must have the same operating system and the same DB2 fix pack level.
- The DB2 client must be installed on the same computer as DB2 RE.
- Required log files and backups must be available locally on the server where you perform log analysis.
- The database and node must be cataloged from the remote computer. See 5.4.2, “Cataloging a remote node and database” on page 102 for details.

Time zone consideration

If the client machine has a different time zone than the server machine, you need to make adjustments to time specifications. For example, if the server is in eastern time zone and the client is in pacific time zone, the server time is ahead by three hours. If you are interested in a time period from T1 a.m. to T2 noon eastern, you should specify T1-3, T2-3 in order to compensate for the time zone difference.

Tip: You can make the client machine time zone the same as the server's time zone in order to eliminate the time compensation requirement. The client machine does not have to be physically in the server's time zone.

5.4.2 Cataloging a remote node and database

To catalog the remote database server's node entry, use the following command:

```
db2 catalog tcpip node KODRMT remote server kodiak server db2c_db2insk1
```

In this command:

- *KODRMT* is the name of TCP/IP node that is cataloged.
- *kodiak* is the remote host name of database server
- *db2c_db2insk1* is the name of TCPI/P listener service for the DB2 instance as defined on the production database server.

You can determine the service name from database server instance with the following command:

```
db2 get dbm cfg
```

Look for the variable SVCENAME.

To catalog a remote database at a newly created TCP/IP node, use the following command:

```
db2 catalog database VPS at node KODRMT
```

In this command:

- *VPS* is the remote database that you want to catalog
- *KODRMT* is the newly defined TCP/IP node.

Now you should be able to access the database remotely.

5.4.3 Remote Analysis using DB2 Recovery Expert GUI

In this scenario, we find that all 1973 year cars in the VPS.CAR table have disappeared on 12 October. After you have cataloged the remote database, you perform log analysis similar to doing log analysis on a local database (see Figure 5-1 on page 86 and Figure 5-2 on page 86 for how to access log analysis). Then, follow these steps:

1. Specify parameters under Report tab as shown in Figure 5-18.

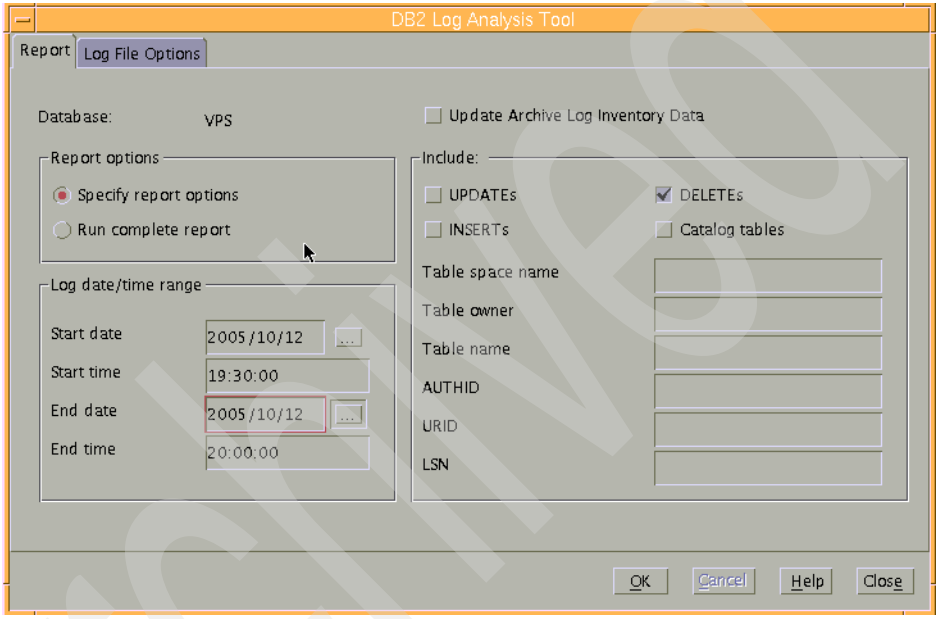


Figure 5-18 Remote log analysis main window

2. Go to the Log File Options tab. With remote log analysis, we need to have log files available locally. Therefore, copy logs from the remote server to the local server and specify the path in the Additional log file directories, as shown in Figure 5-19.

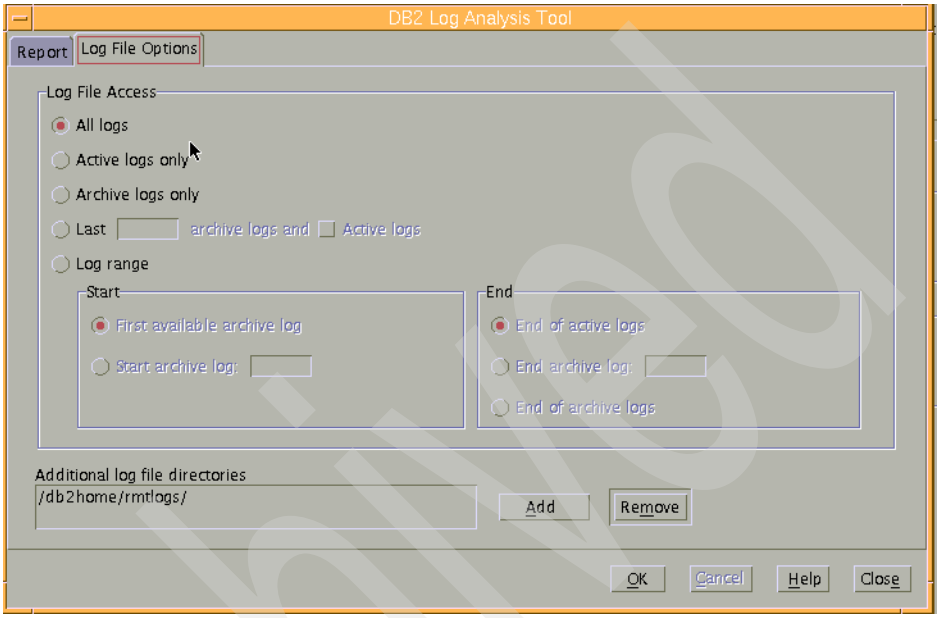


Figure 5-19 Remote Log File Options

3. Click **OK** to start the log analysis. You get a general report that shows 2230 deletes against the VPS.CAR table, as shown in Figure 5-20.

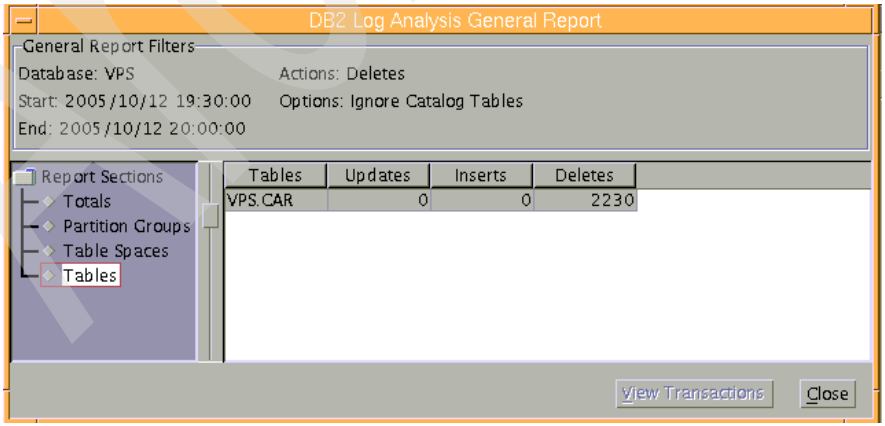


Figure 5-20 Remote log analysis general report

4. Double-click this entry to see further details, as shown in Figure 5-21.

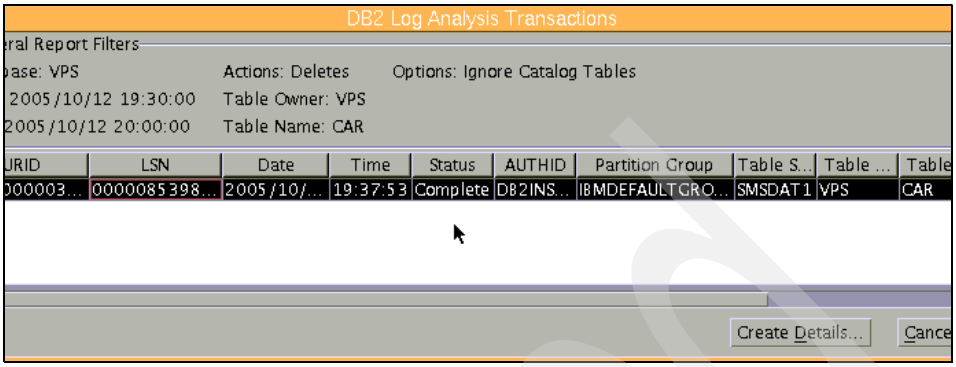


Figure 5-21 Remote log analysis general report details

5. Click **Create Details** to create a detailed report, as shown in Figure 5-22.

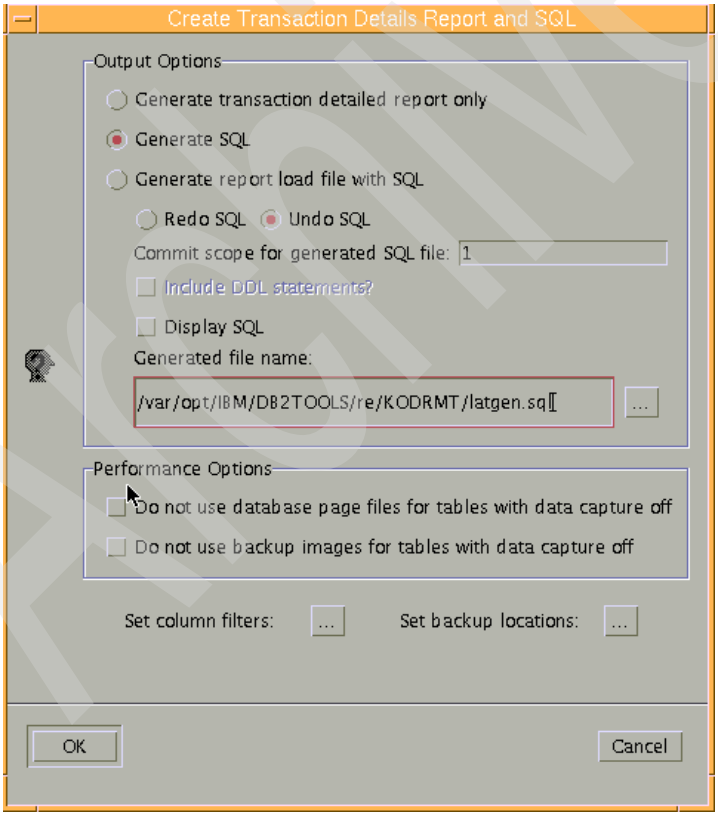


Figure 5-22 Remote Detailed report options

6. Specify to generate a detailed report and also generate undo SQL, which is saved under the generated file name.
7. Click **OK**, and it brings us to Figure 5-23.

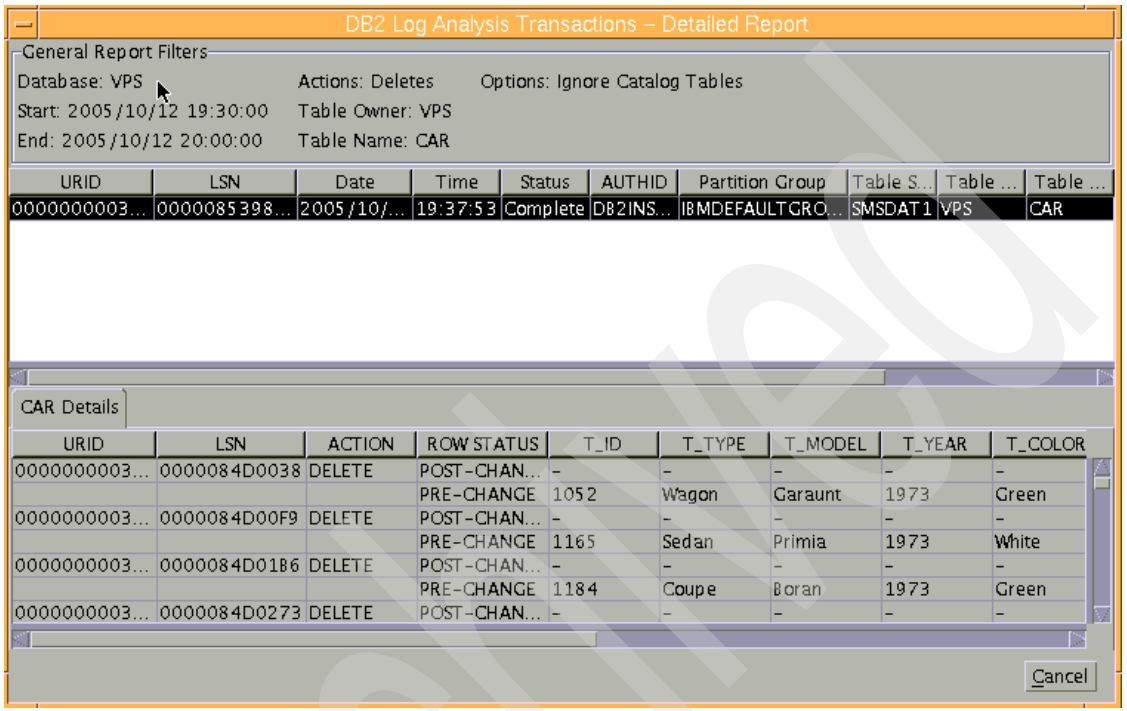


Figure 5-23 Remote log analysis detailed report

This window give us the detailed report. However, there is no tab for undo SQL and no option to run SQL or save SQL because we are doing remote log analysis. You cannot execute generated SQL remotely, so it does not give you any options on the GUI. However, generated undo SQL was saved to the location that was specified under the generated file name. You can take this file to the production server and execute it to undo the deletes.

5.4.4 Remote log analysis using db2la

To perform remote log analysis using the command line tool, you connect to the cataloged remote database from the client machine. When you are connected, the procedure for log analysis is quite similar to doing log analysis locally on the server. You run a general remote log analysis command as follows:

```
db2recex db2la -d vps -a d -b "2005/10/12 19:30:00" -e "2005/10/12 20:00:00"
-u db2insk1 -p db2insk1 -L /db2home/rmtlogs
```

For remote log analysis, you need to provide a extra parameters, as follows:

-u user ID for connect to database.
-p password for user ID.
-L location of the log files copied over from database server.

After the log files are copied over from the database server, you should check or change the log file permissions such that they are readable by the user ID that is executing the DB2 RE commands on client. This command creates a general report and dumps that report to the window. Example 5-2 shows an excerpt from the report.

Example 5-2 Sample of general report from remote log analysis

```
TABLE                       DELETES
--------------------------- -------
SYSIBM.SYSCOLUMNS                12
SYSIBM.SYSCONSTDEP                1
SYSIBM.SYSINDEXAUTH               1
SYSIBM.SYSINDEXCOLUSE             1
SYSIBM.SYSINDEXES                 1
SYSIBM.SYSKEYCOLUSE               1
SYSIBM.SYSTABAUTH                 1
SYSIBM.SYSTABCONST                1
SYSIBM.SYSTABLES                  1
SYSIBM.SYSTABLESPACES             1
SYSIBM.SYSTBSPACEAUTH             1
SYSIBM.SYSUSERAUTH                1
VPS.CAR                        2230
```

In this scenario, we see 2230 deleted rows cars in the VPS.CAR table. Now, we run the following command with additional options to run a detailed report and to create undo SQL.

```
db2recex db2la -d vps -a d -b "2005/10/12 19:30:00" -e "2005/10/12 20:00:00"
-u db2insk1 -p db2insk1 -L /db2home/rmtlogs -n car -D -q u
-f /home/db2insk1/work/undo.sql
```

This command runs the detailed report and dumps it to the window. The undo SQL for the delete statements is saved in the undo.sql file. In the detailed report, all deleted rows are from the year 1973, confirming that these are the deletes that we want to undo. Example 5-3 on page 108 shows a sample of the detailed report that is dumped to the window.

Example 5-3 Detailed report remote log analysis

```
ROW STATUS  T_ID  T_TYPE T_MODEL T_YEAR T_COLOR T_PACKAGE   T_AC T_POWERWINDOWS T_AUTOMATIC
T_DOORS T_PRICE T_DATA
----------- ----- ------ ------- ------ ------- ----------- ---- -------------- -----------
------- ------- -----------
POST-CHANGE -     -      -       -      -       -           -    -              -           -
-       -
PRE-CHANGE  37922 Pickup Boran   1973   Black   Performance N    Y              N           5
5769.00 Sold as-is.

------------------------------------------------------------------------------

ACTION DATE       TIME     URID         LSN          STATUS   AUTHID
------ ---------- -------- ------------ ------------ -------- ----------
DELETE 10/12/05   7:37 PM  0000000003CA 0000084EEA3B COMPLETE DB2INSK1

NGNAME              TSNAME              TABLE OWNER  TABLE NAME
------------------- ------------------- ------------ -------------------
IBMDEFAULTGROUP     SMSDAT1             VPS          CAR

ROW STATUS  T_ID  T_TYPE  T_MODEL T_YEAR T_COLOR T_PACKAGE T_AC T_POWERWINDOWS T_AUTOMATIC
T_DOORS T_PRICE  T_DATA
----------- ----- ------- ------- ------ ------- --------- ---- -------------- -----------
------- -------- ----------------
POST-CHANGE -     -       -       -      -       -         -    -              -           -
-        -
PRE-CHANGE  37935 Minivan Garaunt 1973   Red     Off-Road  N    Y              Y           4
87565.00 Needs a tune up.

------------------------------------------------------------------------------

ACTION DATE       TIME     URID         LSN          STATUS   AUTHID
------ ---------- -------- ------------ ------------ -------- ----------
DELETE 10/12/05   7:37 PM  0000000003CA 0000084EEAFE COMPLETE DB2INSK1

NGNAME              TSNAME              TABLE OWNER  TABLE NAME
------------------- ------------------- ------------ -------------------
IBMDEFAULTGROUP     SMSDAT1             VPS          CAR
```

You can execute **undo.sql** remotely from the client or locally on the production server to recover the deleted rows.

5.5 Report types and interaction of time and log options

This section discusses the two modes of report generation and how time and log file options interact.

5.5.1 General report and detailed report

When you execute the **db21a** command without the **-D** option, it creates the general report. For the DB2 RE GUI, this would correspond to the reports that are shown in Figure 5-5 on page 89 and Figure 5-6 on page 90. When you execute the **db21a** command with **-D** option it creates a detailed report. This corresponds to the running the report as shown in Figure 5-7 on page 90. The detailed report is based on the last general report that was run. Therefore, it is very important to run the general report with proper options first. DB2 RE GUI enforces the proper sequence. However, with **db21a**, you have to make sure that the general report is first run with the proper parameters.

The detailed report is always based on the last general report. So, if you run a detailed report for parameters that are not covered by the general report, you will see unexpected results. For example, if you run a general report for time period T1 and then you run a detailed report for a different time period, T2, you will not see any transactions because the general report does not have transactions from the T2 time period and it ignores transactions from T1 time period.

5.5.2 Time specifications and Log file options and precedence

Log analysis generates reports based on the time period provided, as shown in Figure 5-3 on page 87 for DB2 RE GUI and 5.3, “Log analysis using the command line” on page 99 for db2la. There are a couple of situations where you might not see some or all transactions for the specified time frame:

- You specify a time period for which some or all the log files needed are earlier than the log specified by startlog parameter. In this case, any transactions that are in log files prior to *startlog* are not reported. See “Using the startlog parameter” on page 80.
- You specify the **-P** option with the **db21a** command or an option other than All Logs in the DB2 RE log analysis GUI under the Log file options tab. When you do this, DB2 RE looks at a specific range of logs, and it does not consider any logs outside of that range. If some or all of the time range that is provided falls outside the log range, then you might not see some or all of those transactions.

5.6 Log analysis in a DPF environment

Log analysis in DPF environment is very similar to log analysis in a single partition environment. DB2 UDB uses a *shared nothing* architecture, which means that every DB2 partition acts like a stand-alone database with its own log files, also known as *per node logging*. With per node logging, you can repeat the same log analysis process that we described earlier in this chapter.

This section presents a scenario that highlights some differences. (For details about using log analysis, see 5.1, “Log analysis for auditing” on page 84 and 5.2, “Log analysis for undoing or backing out table changes” on page 92.)

The scenario in this section uses a TPCH database. We work on the VPS.CAR table, which was created in the TPCH_PART table space. We have three database partitions on the server. Database partition 0 is the catalog partition. A user has inserted some rows into the VPS.CAR table by mistake on 5 October. We need to find these rows and undo these INSERTs.

Prerequisites for DPF environment are the same as those described in 5.1.1, “Prerequisites for log analysis for auditing” on page 84 and 5.2.2, “Prerequisite for undoing or backing out SQL” on page 92.

5.6.1 Analyzing changes to the table

In a DPF environment, when you choose a database from main screen, the default partition to which DB2 RE connects is the first logical partition on the local machine. In a multiple logical node configuration on a machine, this partition is the logical node with the lowest number. If no logical nodes exist, DB2 RE connects to the local partition. Figure 5-24 on page 111 shows that we are on partition 0, which is the lowest partition that is available in our environment. Partition 0 is also the catalog node. We recommend that you always start from catalog node.

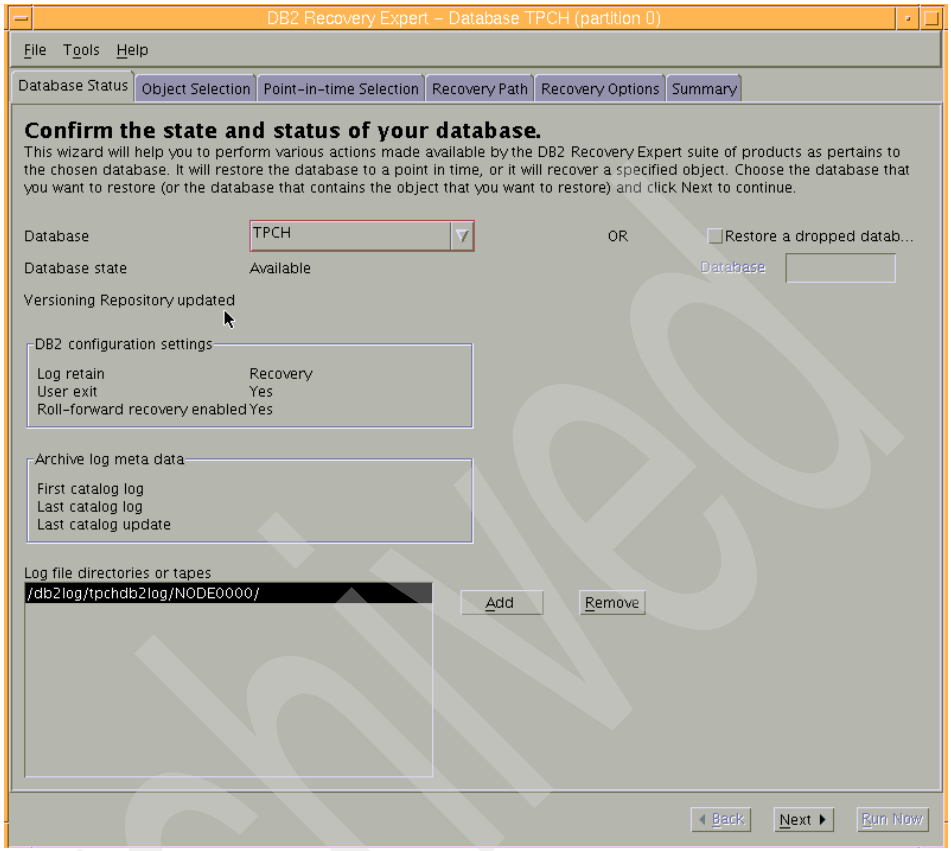


Figure 5-24 DPF Log Analysis main screen

We run log analysis to create a general report for 5 October. We then click **Partition Groups** to determine that there were 558 inserts in `IBMDEFAULTGROUP` on database partition 0, as shown in Figure 5-25.

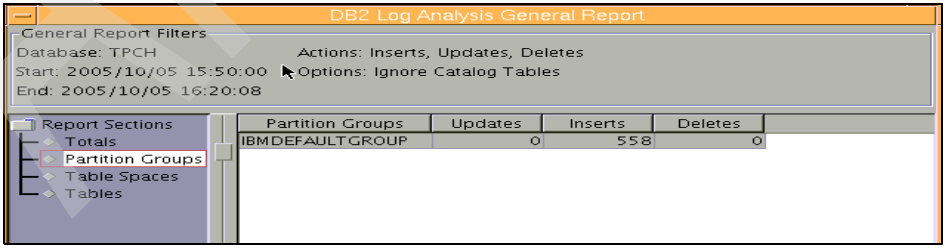


Figure 5-25 Log analysis General Report on inserts

By clicking **Table Spaces** and **Table**, we determine that inserts were done against the VPS.CAR table, which is in TPCH_PART table space. We click this entry to get more details and to generate undo SQL, as shown in Figure 5-26. Notice that DB2 RE produced `DELETE` statements because we are undoing `INSERT` statements. We can click **Run SQL** to execute these `DELETE` statements.

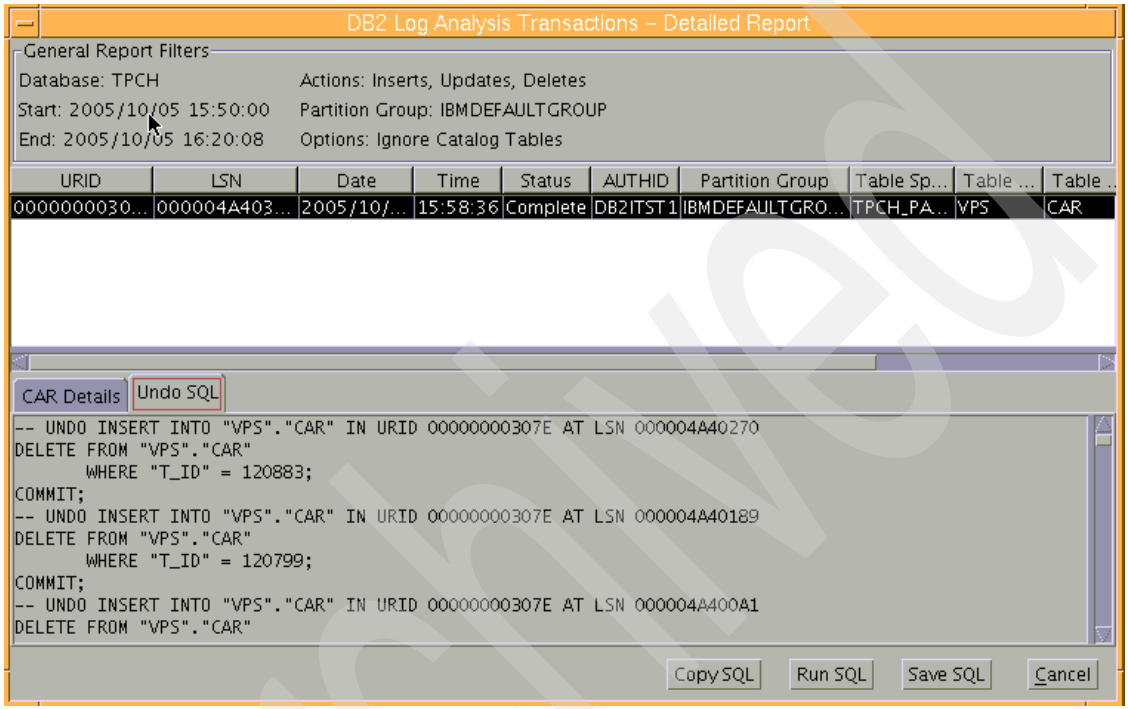


Figure 5-26 Detailed Report showing undo SQL

At this point, we have undone the changes on database partition 0. This table has data on a catalog partition as well, which is why we found transactions here.

> **Note:** If a table does not have data on a catalog node, then log analysis on the catalog node does not return entries for such a table.

Now, we follow this exact procedure on every database partition where this table exists to undo all inserts against the VPS.CAR table.

5.6.2 Switching between nodes

You can switch between nodes by selecting **File** → **Connect** from the main DB2 RE window to open the connection window where you can specify to which partition you want to connect, as shown in Figure 5-27.

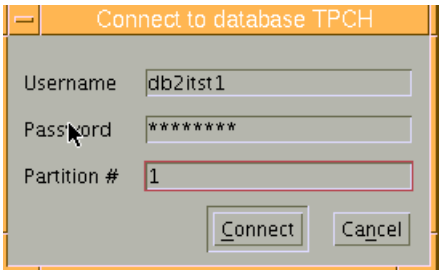


Figure 5-27 Switching db2 nodes from DB2 RE

We connect to database TPCH on partition 1 when we click **Connect**. If DB2 RE is installed on a shared file system, then you can switch to any database partition from the same DB2 RE GUI. Otherwise, you can only switch to logical database partitions on the same server. In order to switch to a database partition on a different server, you would need to log on to that server and start DB2 RE from there.

5.7 Considerations and recommendations

This section discusses considerations and recommendations when using log analysis.

5.7.1 Recovery scenarios without DB2 Recovery Expert

Recovery scenarios without DB2 RE can be very time consuming and disruptive. In a recovery scenario with native DB2 UDB, you have to restore a database or table space by rolling forward through the logs and stopping before the point-in-time when the transactions were done. In large environments, this method of recovery could translate into several hours of down time. Moreover, it is not always possible to restore a table space and roll forward to a point-in-time. Thus, you must restore the database to be able to stop at the desired point-in-time. For example, take the following scenario:

- You run some SQL against table TBL1 in table space TSP1 at time T1 by mistake and you need to undo it.
- Another Table TBL2 in same table space TSP1 has since been dropped at time T2.

When you restore TSP1, it forces you to roll forward to at least time T2. When TBL2 was dropped, catalogs were also modified to reflect this change. If DB2 allows you to stop roll forward before T2, the database becomes inconsistent, because TSP1 is out of sync with the catalog table space. Therefore, you have to roll forward to time T2, meaning you cannot undo the transactions in table TBL1 using a table space restore. You would have to restore entire database to be able to stop before time T1.

5.7.2 Data capture changes

Data capture changes is an option that you can enable for a table when you create the table or when you use the **ALTER** command as follows:

```
db2 alter table tablename data capture changes
```

In this command, *tablename* is the name of the table.

This option indicates that any extra information regarding the SQL changes to this table are written to the log. This option was created for tables that needed to be replicate using the capture program. Extra information written to logs can include the following:

- The AUTHID.
- DB2 logs full-row images for each UPDATE statement. Without this option, DB2 only logs information about the changes that are caused by an UPDATE statement.

In most circumstances, we do not recommend enabling data capture changes, because the extra logging overhead can be substantial. However, you might want to enable it in the following circumstances:

- If UPDATE activity is low.

  If you have an environment where the main SQL activity is INSERTs and DELETEs compared to very few UPDATEs, then data capture changes will not cause much extra logging, so the benefits could outweigh the small overhead.

- If the performance of log analysis is critical.

  Having data capture changes enabled for a table can benefit the performance of log analysis, because it removes the need to use masked update reconstruction which can be time consuming, see 5.7.4, “Masked update reconstruction” on page 115 for details. If data capture changes is enabled, then the log record for an UPDATE contains all the data required to reconstruct that UPDATE.

- If you plan to replicate the table.

  If you plan to replicate the table using DB2 Replication, you must enable data capture changes for that table.

5.7.3 Only shows committed transactions

In DB2 log files, time stamps are recorded only for `commit` statements. Therefore, time given for a change to a row in a log analysis report is the time at which the change was committed. All changes made in one transaction (URID) will show as having occurred at the same time. An effect of this is that transactions are only included in a log analysis report if they were committed between the begin and end times that you specify, regardless of when the changes to the rows actually occur.

5.7.4 Masked update reconstruction

The masked updated reconstruction method is the method that the Log Analysis tool uses to reconstruct update statements. By default, when a row is updated, DB2 UDB does not write a full before or after image of the row to the log. It only logs information about the columns that were changed. Thus, the Log Analysis tool cannot construct full before and after images of the row from only the log record for the update. To construct before and after images of the row, the Log Analysis tool must do one of two things:

- Forward update reconstruction

  Start with the image of the row from a backup image and roll forward through all the updates to the row in the logs to get to the before image for the update in question. Then it can reconstruct the after image by applying the change given in the log record.

- Backward update reconstruction

  Start with the image of the row that currently exists in the table and roll back through all of the updates to the row in the logs to get to the after image for the update in question. Then, DB2 RE can reconstruct the before image by reversing the change given in the log record.

If you use the **db2la** command without the **-G** option or use the Log Analysis tool GUI, DB2 RE selects the most efficient update reconstruction method automatically.

5.7.5 Generated always columns

If a column is defined as GENERATED ALWAYS, this means that DB2 generates values for the column. The user is not allowed to specify a value for

the column. This causes a problem when log analysis generates INSERTs in undo SQL for a table with a GENERATED ALWAYS column. The undo SQL cannot use specific values in the GENERATED ALWAYS column, so it has to use DEFAULT as the value. This means that, when you recover a table with a GENERATED ALWAYS column using redo or undo SQL that contains INSERTs, the values inserted into the GENERATED ALWAYS column might not be the same as the original values. This is unavoidable due to the design of GENERATED ALWAYS columns.

5.7.6 DB2 Recovery Expert can quiesce the table space

When you run the Log Analysis tool using backward update reconstruction (see 5.7.4, “Masked update reconstruction” on page 115), DB2 RE quiesces the table space at the point when you generate a detailed report or redo or undo SQL. For the DB2 RE GUI, this is the point at which you double-click a transaction in the Log Analysis Transactions window (see Figure 5-13 on page 96) or select whether to produce undo or redo SQL (see Figure 5-14 on page 96).

For the **db2la** command, this is the second iteration of the command, where you use the **-D** option. The quiesce is required because DB2 RE needs to access the current table data to reconstruct updates, and it is important that data is not changed by users during this process.

If you use forward update reconstruction, however, it does not quiesce the table space. You can force **db2la** to use forward update reconstruction using the **-G f** flag. From DB2 RE GUI, this could be done by selecting **Do not use database page files for tables with Data Capture off** , as shown in Figure 5-14 on page 96.

5.7.7 DB2 Recovery Expert works one row at a time

In this example, the original update statement was as following:

```
update vps.car set t_color='Red' where t_year=1973 and t_model='Boran' and
t_color='Blue'
```

Now, because there might be 1973 Boran model Red cars in the table even before you did the update, we cannot do an update statement such as:

```
update vps.car set t_color='Blue' where t_year=1973 and t_model='Boran' and
t_color='Red'
```

Doing this update statement sets all of the red 1973 Borans to blue, not just the ones that were updated. For this reason, we need to undo updates one row at a time. We had updated 43 rows, and we need to make sure we only change these 43 rows to their original state.

Moreover, DB2 logs one entry or LSN for every row which is either deleted, inserted, or updated. There is no information about the original statement that actually caused the change to the row. Therefore, DB2 RE generates undo/redo SQL per row for DELETE and INSERT activity as well.

Redo SQL might not be the same

For the same reasons as above, the redo SQL might not be the same as original SQL issued by user. Consider this example:

```
db2 delete from VPS.CAR
```

This command deletes all rows in the table, but redo SQL comprises of one delete statement per deleted row. It would produce the same results as long as the state of the table is same as when the original statement was run.

5.7.8 Log analysis for tables with LOBs

The process for analyzing tables with LOB data is the same as for tables without LOB data. But remember that DB2 RE can only recover LOB data if it is logged in the first place. With LOB data columns, there is a Not Logged option, which basically means do not log the LOB data. If you are not logging the LOB data, then DB2 RE clearly cannot create report or generate undo or redo SQL for it. There is also a one gigabyte (GB) hard limit per LOB object logged to DB2 log files. Thus, you cannot log LOB objects larger than 1 GB as per DB2 architecture.

When undo SQL is created for a LOB table, you would see something similar to Example 5-4 in the undo or redo SQL file.

Example 5-4 Undo SQL created for a table with LOB data

```
-- UNDO DELETE FROM "DB2INSK1"."EMP_PHOTO" IN URID 000000000359 AT LSN
000001388782
-- BEGIN PARAMETERS
-- BFILE /var/opt/IBM/DB2TOOLS/re/db2insk1/000001388782.0
-- END PARAMETERS
INSERT INTO "DB2INSK1"."EMP_PHOTO"
           ("EMPNO",
            "PHOTO_FORMAT",
            "PICTURE")
    VALUES ('000150',
            'gif',
            ?);
```

Notice the `BFILE` in the comment line. When DB2 RE recovers a table with LOB data, undo or redo SQL file has pointers to where the LOB data is actually

stored. Each LOB is saved in an individual file in your DATA_DIR. As you may already know that you cannot insert LOBS from DB2 command line. Therefore, for the LOB column, DB2 RE has specified host variable question mark (?). When you run undo or redo SQL, DB2 RE GUI performs extra processing to insert the LOB data. However, if you generate the SQL using the **db2la** command outside the GUI, the SQL is not directly runnable by DB2. You either need to write your own program to process the LOBs, or you can use the ARYSQL program that comes with DB2 RE.

Tip: The file name of each LOB corresponds to the LSN number of that row modification

To use ARYSQL in Windows:

```
db2rewin32 -is:javaconsole arysql -d dbname -f fileName [ -u username [ -p
password ]]
```

To use ARYSQL in UNIX and Linux:

```
db2recex arysql -d dbname -f filename [ -u username [ -p password ]]
```

Log analysis against table with LOB data in different table space

Currently, DB2 RE does not support running detailed reports or generating undo or redo SQL for tables that have LOB/Long data in a separate table space than the rest of table data. Therefore, you cannot analyze or back out changes to such tables using log analysis.

5.7.9 Triggers and referential integrity

Tables with triggers and referential integrity are special cases, because SQL in one table can cause SQL statements against other tables. In DB2 logs, we have just the SQL statements but no information if it was done due to referential integrity constraints or a trigger. Therefore, DB2 Grouper tool has been introduced to manage such situations. See Chapter 10, “Installing and using the DB2 Grouper component” on page 305 for more details.

The DB2 RE log analysis function displays activity from the DB2 recovery log. Referential integrity changes are seen on the recovery log as SQL activity, inserts, updates, and deletes. It is not explicitly visible on the recovery log that an referential integrity relationship exists between two objects. Therefore, when filtering the log to produce undo or redo SQL, ensure that any changed objects in an referential integrity set are not excluded via filters. Filtering by a time interval

will display all log activity over a time period, and therefore will display all referential integrity activity and affected objects.

Special handling may be required for cascade DELETEs when doing log analysis to generate undo or redo SQL. Below are two examples where the log analysis tool will create SQL for tables that have an referential integrity relationship. These examples show only one parent or dependent relationship. Multiple relationships need to be considered by the user as well.

Consider the following cascade delete scenario where referential integrity has been defined between the parent and dependent tables with the delete rule of cascade (see Figure 5-28). Looking at the log, a delete of the parent row, D(p), causes the cascade delete of four dependent tables rows D(d1), D(d2), D(d3), and D(d4).

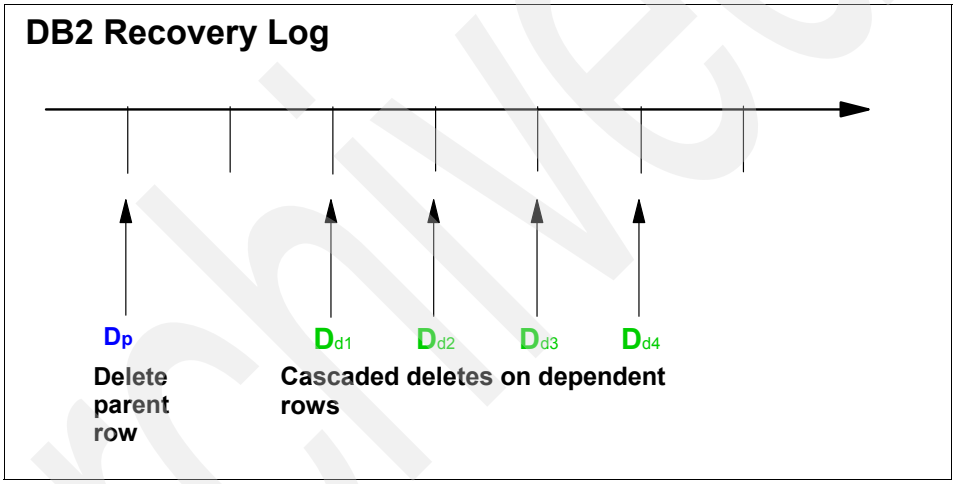


Figure 5-28 Referential integrity delete cascade operation

Example 1 - Log analysis to generate redo SQL

In the first example, we show how DB2 RE generates the redo SQL for a cascaded delete operation. In this case, the redo SQL should be analyzed to determine whether or not the cascaded delete operations will cause errors that can be ignored or errors that require special handling. If log analysis is invoked to generate redo SQL for the cascade delete scenario in Figure 5-28, the following SQL statements would be generated (shown in Example 5-5).

Example 5-5 Redo SQL DELETE sequence

```
DELETE <parent>
DELETE <dependent1>
DELETE <dependent2>
DELETE <dependent3>
```

```
DELETE <dependent4>
```

If the user executed the above generated SQL without editing, the DELETE from the parent will complete successfully, and the DELETE from the dependent tables would fail. (The specific dependent DELETE error would be `SQL0100W No row was found for FETCH, UPDATE or DELETE; or the result of a query is an empty table. SQLSTATE=02000`.). In this example, the error conditions are acceptable because the first parent DELETE statement would cause subsequent cascade DELETEs against the dependent tables, therefore making the explicit dependent DELETE statements unnecessary.

Example 2 - Log analysis to generate undo SQL

In the second example, we show how DB2 RE generates the undo SQL for the same cascaded delete operation. In this case, the undo SQL is created in reverse order. Therefore, for these a cascade INSERT statements, special handling is required as illustrated in this example. If log analysis is invoked to generate undo SQL for the cascade delete scenario in Figure 5-28 on page 119, the following SQL statements, shown in Example 5-6, would be created.

Example 5-6 Undo SQL INSERT sequence

```
INSERT <dependent4>
INSERT <dependent3>
INSERT <dependent2>
INSERT <dependent1>
INSERT <parent>
```

If the user executed the above generated SQL without editing, the `INSERT` for each dependent table would fail, and the `INSERT` to the parent table would complete successfully. (The specific dependent row `INSERT` errors would be `SQL0530N The insert or update value of the FOREIGN KEY "<dependent table foreign key constraint>" is not equal to any value of the parent key of the parent table. SQLSTATE=23503`.) In this example, the error conditions are not acceptable and can be remedied in one of the following ways:

- Edit the SQL and place the dependent table(s) `INSERT`s after the parent table `INSERT`s.
- Generate the SQL for the parent and dependent table(s) separately and then run the undo SQL for the parent table first.

- Turn off integrity checking during this operation.

  Note: Take great care with this option because other updates that are outside of the DB2 RE undo operations against these tables will not have any integrity checking.

  a. db2 set integrity for table off.
  b. Run the undo SQL for the objects that are related to referential integrity.
  c. db2 set integrity for table immediate checked.

Referential integrity and object recovery

When using DB2 RE to recover objects, the user must take referential integrity constraints into consideration.

- Objects that are updated and participate in an referential integrity relationship should be recovered together to the same point-in-time. The user is responsible to ensure that referential integrity constraints are honored. For example, if you recover to a point-in-time where DB2 RE will generate undo or redo SQL against a parent table, the user must understand that a referential integrity relationship exists, and ensure that the dependent table are subsequently restored to the same point-in-time.
- If running undo SQL, either:
  - Run the undo SQL separately for each table. In this case the parent table should always be recovered before the child table.
  - Run undo SQL on tables simultaneously, but turn off integrity checking first.
- If running redo SQL, either:
  - Edit the redo SQL, removing any deletes that occurred as a result of referential integrity, before run it.
  - Run the redo SQL as is, but make sure you commit every row and ignore error messages that are related to delete cascaded row.
  - Turn off integrity checking, then run the full redo SQL.

5.7.10 What if data has changed

If the table has changed, since the period of time for which you generate undo SQL then DB2 RE may not be able to generate undo SQL. Consider the example of a user who runs this update statement between time T1 and T2:

```
update vps.car set t_color='Red' where t_year=1973 and t_model='Boran' and
t_color='Blue'
```

thereby updating all 1973 blue Borans to red 1973 blue Borans. Sometime later, between T3 and T4, another user updates like this:

```
update vps.car set t_color='Silver' where t_year=1973 and t_model='Boran' and
t_color='Red'
```

If you try to generate undo SQL for the first update statement, it would report:

```
NOT ENOUGH DATA TO GENERATE UPDATE STATEMENT
```

This happens because the updated rows have been changed again. In order to undo the first statement, you need to undo the second update statement first. Once it has been undone, you can undo the first one. For this reason, we recommend that you should always run log analysis from the time of interest to current time. This would show you if other changes have occurred to the table and you can take appropriate actions. If too many changes have taken place in the table for which you are trying to create undo SQL, it may be easier to recover table to a point-in-time when the contents were known to be consistent.

6

Backing out insert, update, and delete operations

This chapter describes how to use DB2 Recovery Expert to back out the insert, update, and delete operations on table, table space, and database to a point-in-time. This chapter also shows the details of using DB2 Recovery Expert to perform redirected restore and recovering related objects using DB2 Recovery Expert and DB2 Grouper.

This chapter includes the following sections:

6.1 Point-in-time recovery of a table after running DDL

This section explains how to undo delete operations against the VPS.CAR table when DDL changes have occurred and possibly other SQL have run since the delete activity.

6.1.1 Prerequisites for this scenario

This scenario uses the VPS database. The VPS.CAR table exists in the SMSDAT1 table space. This scenario requires the following:

- LOGRETAIN must be set to RECOVERY.
- Versioning Repository *must* have been updated or created since the last DDL was run.
- A full, offline backup with data for the object that you are recovering must exist.
- The ALID should be updated for DB2 Recovery Expert (DB2 RE) to determine the best option for recovery. If the ALID is not updated, then DB2 RE might not determine the optimal recovery path.

Tip: We recommend that you update Versioning Repository every time you do a recovery that involves DDL changes.

6.1.2 The time line for this scenario

Table 6-1 on page 125 shows the time line for this scenario.

Table 6-1 Scenario time line

Time	Action
T1	full offline backup taken
T2	Delete job run against table VPS.CAR
T3	DDL job run against table VPS.CAR to add one column
T4	some unknown SQL run against VPS.CAR

The delete job deleted all 1950 model cars from the table.

6.1.3 Details of activity against table

Example 6-1 shows the number of rows in the table before delete jobs ran.

Example 6-1 Row counts before delete job

```
/home/db2insk1/yasir->db2 "select count(*) from vps.car"
1
-----------
     119986

  1 record(s) selected.
```

Example 6-2 shows the number of rows after the delete jobs ran. A total of 2061 rows were deleted.

Example 6-2 Row counts after delete job

```
/home/db2insk1/yasir->db2 "select count(*) from vps.car"
1
-----------
     117925

  1 record(s) selected.
```

Some time after the delete activity, at T3, the following DDL changes were made to the table:

```
db2 alter table vps.car add t_condition varchar(30)
```

Now, we need to do undo the delete operations.

6.1.4 Recovering the table to a point-in-time

To recover the table to a point-in-time, follow these steps:

1. Start DB2 RE and connect to the database. Figure 6-1 shows the status after the connection has been made.

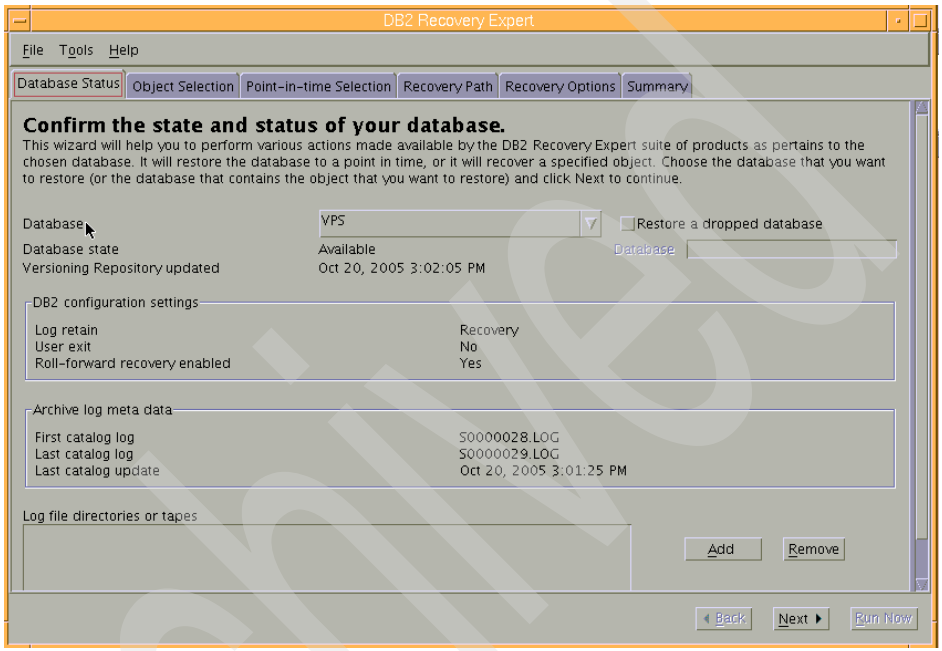


Figure 6-1 DB2 RE main screen

2. Move to the Object Selection tab (Figure 6-2 on page 127) to select the table that you want to recover.

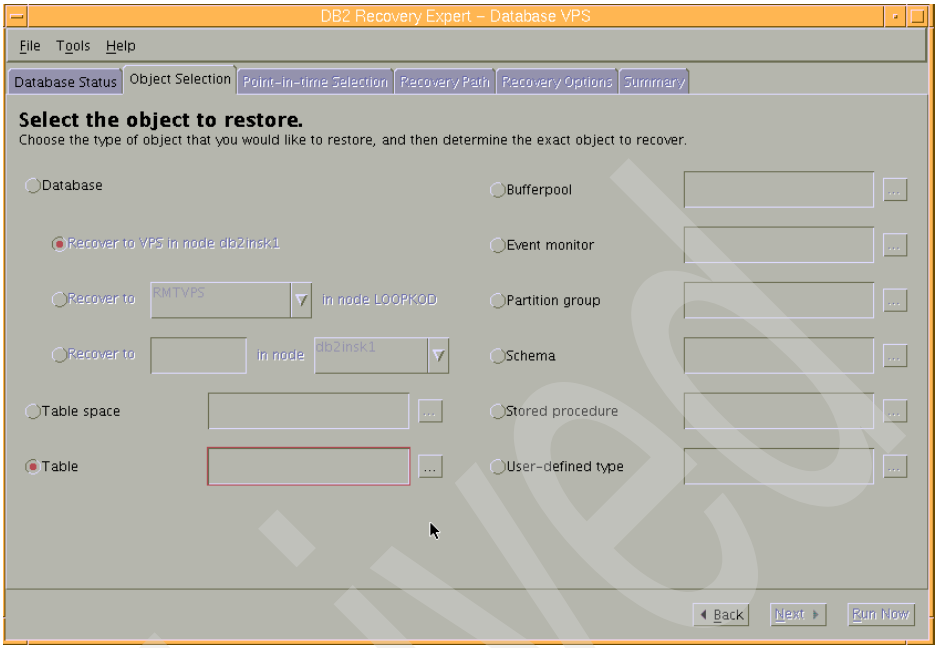


Figure 6-2 DB2 RE Object Selection

3. You can select the browse icon (the "**...**") to browse a list of tables (Figure 6-3).

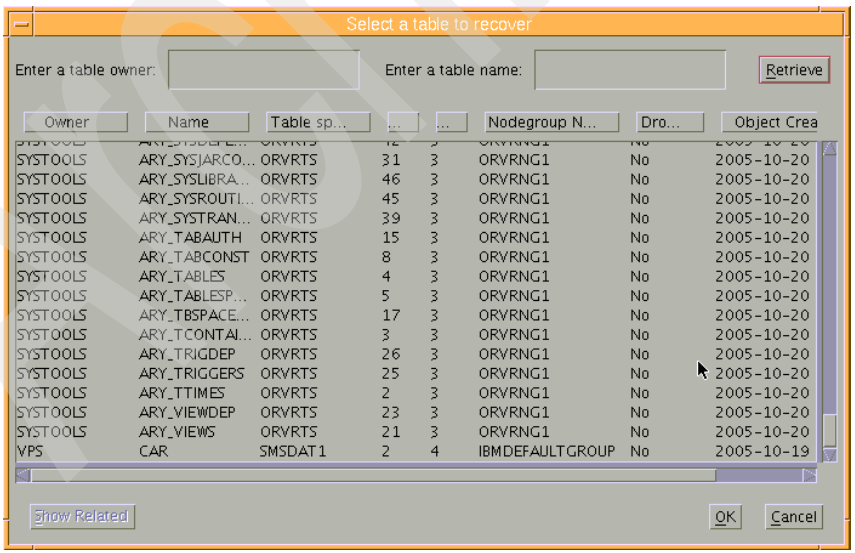


Figure 6-3 DB2 RE table list window

4. Select the VPS.CAR table and click **OK**.
5. Select a point-in-time to which you want to recover, as shown in Figure 6-4.

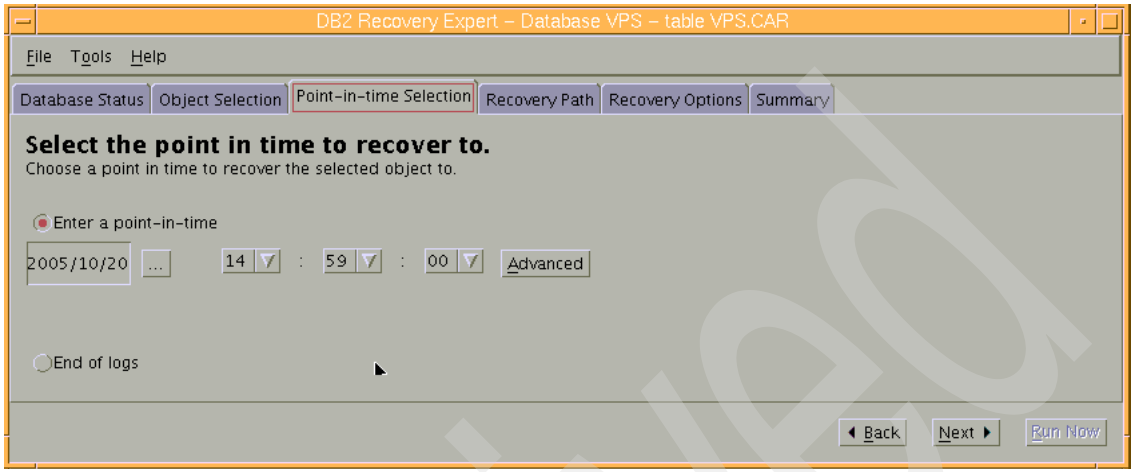


Figure 6-4 DB2 RE point-in -time selection window

6. Specify a time that corresponds to T1 in the scenario time line. Then, click **Next** to see the Recovery Path, as shown in Figure 6-5.

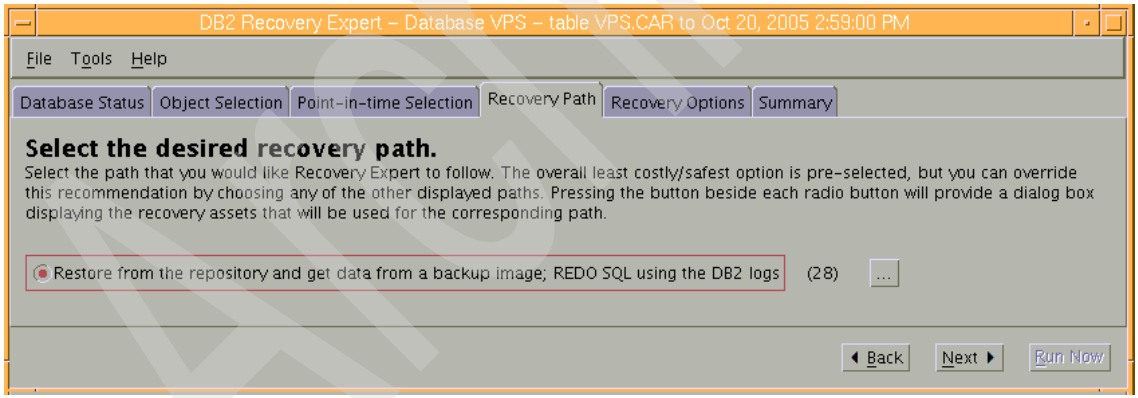


Figure 6-5 DB2 RE Recovery Path window

DB2 RE recovers the data from a backup image and redoes the SQL. However, in this scenario, the DDL has been run. Thus, it is not possible to simply run undo SQL to get to the desired point-in-time.

7. Click **Next** to choose the recovery options, as shown in Figure 6-6.

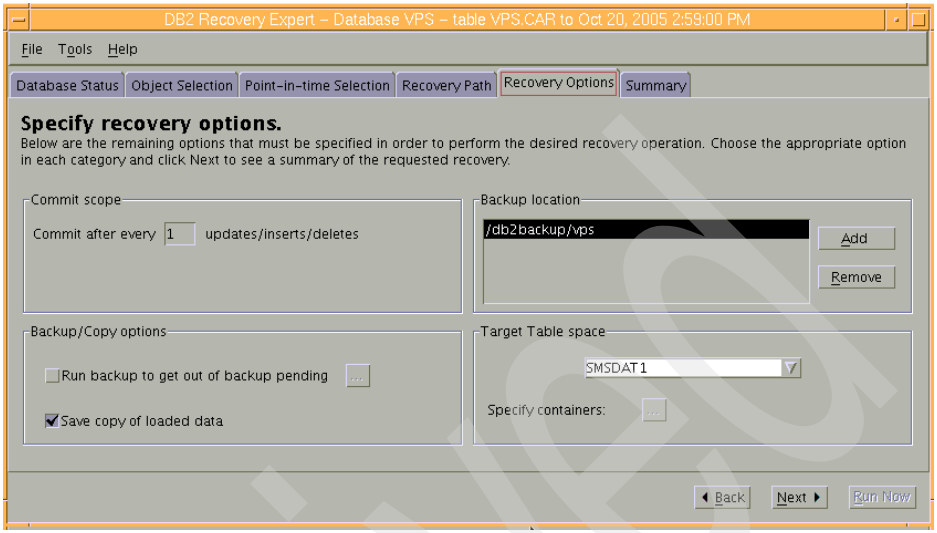


Figure 6-6 Recovery Options window

8. Under Backup Copy Locations, select **Save copy of loaded data**.

   DB2 RE runs a DB2 **LOAD** command under the covers. This option indicates that a `COPY YES` clause is added to the **LOAD** command.

   Alternatively, you could select **Run backup to get out of backup pending**, which is the equivalent of adding a `COPY NO` clause to the **LOAD** command. With this selection, DB2 RE issues a backup of the table space to bring it out of the backup pending state. With the `COPY NO` option, the loaded data is not logged to DB2 logs. Therefore, it is not possible to recover data that is added through this load operation. Thus, DB2 leaves the table space in backup pending state to force the user to take a backup of the table space. With `COPY YES`, the table data is backed up, and there is no need to take a backup.

9. Click **Next** to go to the Summary window. Execute the recommended actions, as shown in Figure 6-7 on page 130.

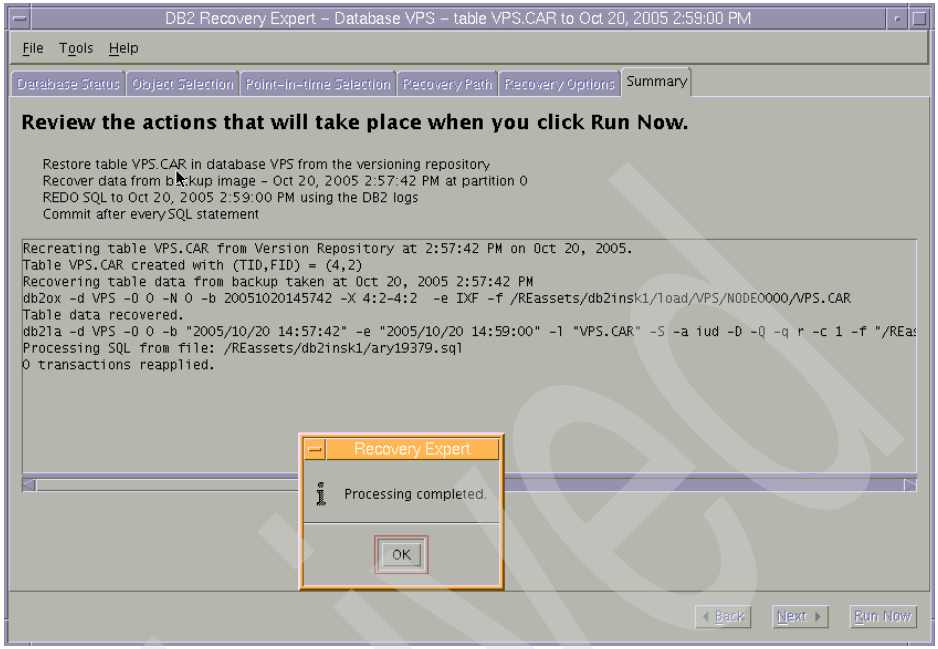


Figure 6-7 DB2 RE Summary and result window: VPS.CAR table recovered

We see that zero (0) transactions were applied after loading data from the backup image. This information is correct because we wanted to recover to a point-in-time before T2 and after T1. We check the row counts, and we have our original number of rows back, as shown in Example 6-3.

Example 6-3 Row count after the recovery to point-in-time

```
/home/db2insk1/yasir->db2 "select count(*) from vps.car"
1
-----------
     119986

  1 record(s) selected.
```

We also notice that DDL against the table has been undone, as shown in Example 6-4.

Example 6-4 CAR table structure: added column has been dropped.

```
/home/db2insk1->db2 describe table vps.car

Column                          Type      Type
name                            schema    name                Length   Scale
Nulls
------------------------------- --------- ------------------ -------- -----
------
T_ID                            SYSIBM    INTEGER                4     0 No
T_TYPE                          SYSIBM    VARCHAR               50     0 No
T_MODEL                         SYSIBM    VARCHAR               50     0 No
T_YEAR                          SYSIBM    INTEGER                4     0 Yes
T_COLOR                         SYSIBM    VARCHAR               50     0 Yes
T_PACKAGE                       SYSIBM    VARCHAR               50     0 Yes
T_AC                            SYSIBM    CHARACTER              1     0 Yes
T_POWERWINDOWS                  SYSIBM    CHARACTER              1     0 Yes
T_AUTOMATIC                     SYSIBM    CHARACTER              1     0 Yes
T_DOORS                         SYSIBM    SMALLINT               2     0 Yes
T_PRICE                         SYSIBM    DECIMAL               10     2 No
T_DATA                          SYSIBM    VARCHAR              500     0 Yes
```

6.1.5 Recovery options without DB2 Recovery Expert

There is no table level restore functionality in DB2 UDB. Table space level restore cannot be done because DDL has been run. You cannot restore a table space and stop at a point-in-time before the time DDL was run, because DB2 catalogs would get out of sync with objects definition in the table space. For this reason, DB2 would not allow to stop roll forward before the time DDL was run. Therefore, you cannot restore at table space level either. The only option is to restore the entire database using a roll forward to a point-in-time before DDL was run. This method of recovery can cause long outages in a large production environment.

6.2 Recovering table space to point-in-time when DDL has been run

This section describes recovering a table space to a point-in-time before the time when DDL is run against one of the tables in the table space. In this example, multiple tables have been changed in this table space, and one of the tables has been altered with DDL.

6.2.1 Prerequisites for Scenario

Prerequisites for this scenario are the same as described in 6.1.1, “Prerequisites for this scenario” on page 124.

6.2.2 Scenario time line

Table 6-2 lists the time line for this scenario.

Table 6-2 Time line for operations against table space SMSDAT1

Time	Action
T1	Full offline backup taken
T2	Rows deleted from VPS.CAR table
T3	DDL is run against VPS.CAR table
T4	Rows deleted from VPS.CUSTOMER table
T5	Present time

We want to do recovery such that SQL for VPS.CUSTOMER and DDL against VPS.CAR is undone but the delete job for VPS.CAR at T2 is preserved. In this case, the best option is to recover the table space SMSDAT1 to a point-in-time between T2 and T3.

6.2.3 Details of activity against table space

Example 6-5 shows row count in the VPS.CAR table before at time T1.

Example 6-5 Row count prior to delete job against vps.car

```
/home/db2insk1/yasir->db2 "select count(*) from vps.car"

1
-----------
     119986

  1 record(s) selected.
```

Example 6-6 shows the row counts in the VPS.CAR table at time T2.

Example 6-6 Row count after delete job against vps.car

```
/home/db2insk1->db2 "select count(*) from vps.car"

1
-----------
     117925

  1 record(s) selected.
```

At T3, the following DDL is run against the VPS.CAR table:

```
db2 alter table vps.car add t_condition varchar(30)
```

Example 6-7 shows the number of rows in the VPS.CUSTOMER table before time T4.

Example 6-7 Row count before delete job against vps.customer

```
/home/db2insk1->db2 "select count(*) from vps.customer"
1
-----------
     253456

  1 record(s) selected.
```

Example 6-8 shows the number of rows in the VPS.CUSTOMER table after time T4.

Example 6-8 Row count after delete job against vps.customer

```
/home/db2insk1->db2 "select count(*) from vps.customer"
1
-----------
     250313

  1 record(s) selected.
```

6.2.4 Recovering table space to a point-in-time

To recover table space to a point-in-time, follow these steps:

1. Connecting to the database, go to the Object Selection window, and select the SMSDAT1 table space, as shown in Figure 6-8.

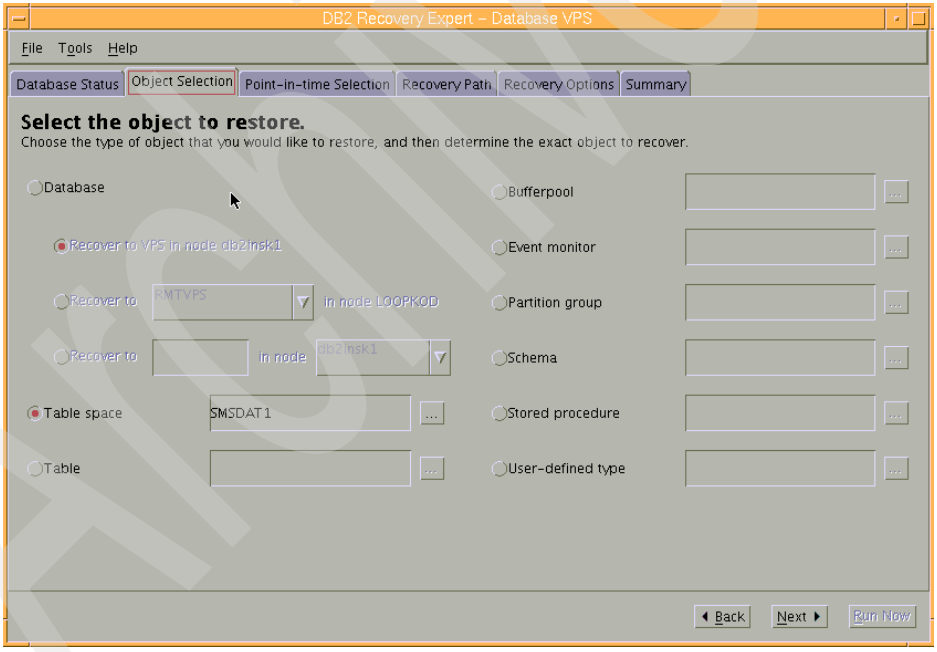


Figure 6-8 DB2 RE object selection: Table space

2. Click **Next** to go to the Point-in-time Selection window (Figure 6-9).

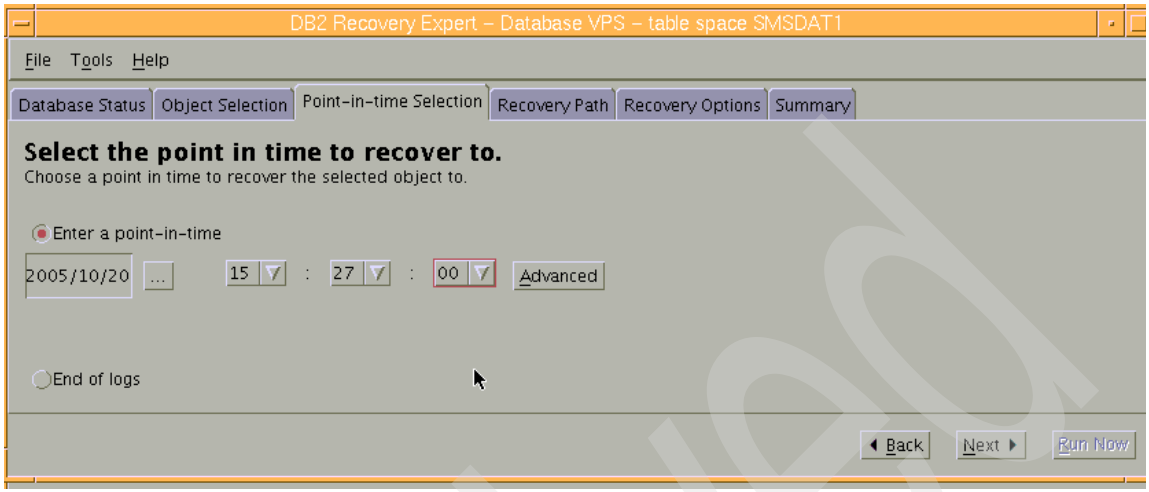


Figure 6-9 Point-in-time Selection

3. Choose a time that takes you between T2 and T3.

> Tip: You can run log analysis to determine the exact times when certain transactions were executed. See Chapter 5, “Log analysis” on page 83 for more information.

4. Click **Next** to see the Recovery Path window (Figure 6-10).

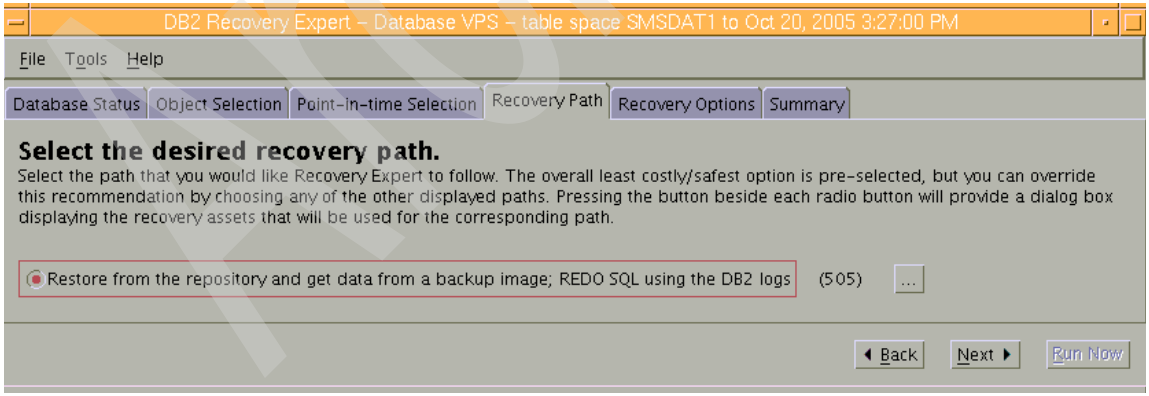


Figure 6-10 Recovery Path window

DB2 RE determines that the undo SQL option is not viable because DDL has been run after the specified point-in-time.

5. Click **Next** to go to the Recovery Options window (Figure 6-11).

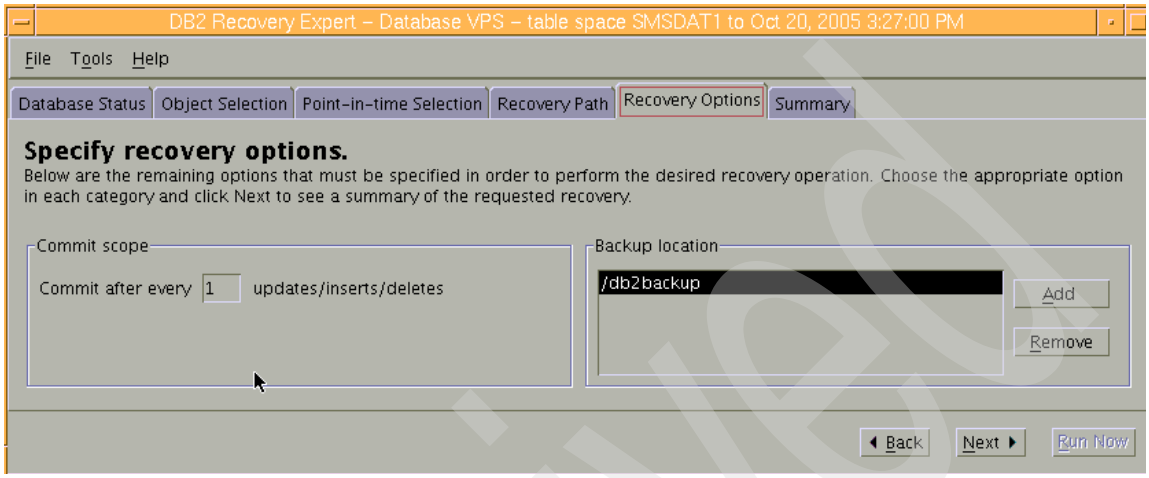


Figure 6-11 Recovery Options

6. Take the defaults in this window.
7. Click **Next** to go to the Summary window and click **Run Now**, see Figure 6-12 on page 137. DB2 RE did not give us Backup Copy options because it is not going to run DB2 LOAD against any of the tables.

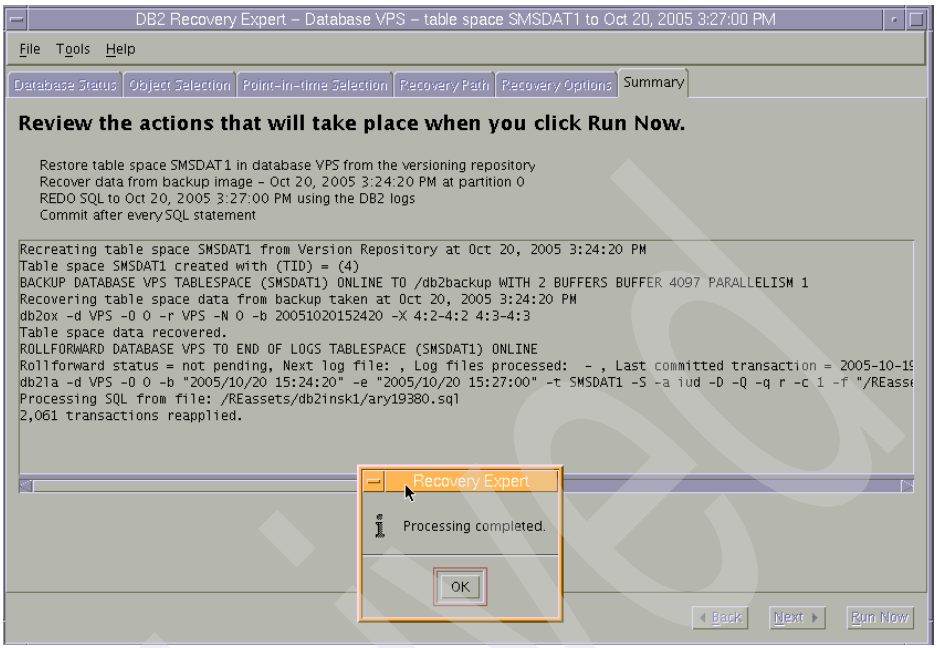


Figure 6-12 Recovery results

Table space SMSDAT1 has been recovered to the desired point-in-time. We check the row counts, as shown in Example 6-9.

Example 6-9 Row count after point-in-time recovery

```
/home/db2insk1->db2 "select count(*) from vps.car"
1
-----------
     117925

  1 record(s) selected.
/home/db2insk1->db2 "select count(*) from vps.customer"
1
-----------
     253456

  1 record(s) selected.
```

VPS.CAR still has 117925 rows, confirming that the delete operation has not been undone. VPS.CUSTOMER has 253456 rows, confirming that its delete operations have been undone. Moreover, the alter table DDL has been undone as well, as shown in Example 6-10 on page 138.

Example 6-10 vps.car table structure

```
/home/db2insk1->db2 describe table vps.car

Column                    Type      Type
name                      schema    name            Length   Scale  Nulls
------------------------- --------- --------------- -------- ------ -----
T_ID                      SYSIBM    INTEGER              4      0  No
T_TYPE                    SYSIBM    VARCHAR             50      0  No
T_MODEL                   SYSIBM    VARCHAR             50      0  No
T_YEAR                    SYSIBM    INTEGER              4      0  Yes
T_COLOR                   SYSIBM    VARCHAR             50      0  Yes
T_PACKAGE                 SYSIBM    VARCHAR             50      0  Yes
T_AC                      SYSIBM    CHARACTER            1      0  Yes
T_POWERWINDOWS            SYSIBM    CHARACTER            1      0  Yes
T_AUTOMATIC               SYSIBM    CHARACTER            1      0  Yes
T_DOORS                   SYSIBM    SMALLINT             2      0  Yes
T_PRICE                   SYSIBM    DECIMAL             10      2  No
T_DATA                    SYSIBM    VARCHAR            500      0  Yes
```

The table space has been recovered as required.

6.2.5 Recovery options without DB2 Recovery Expert

The recovery options without DB2 RE for this scenario are the same as those in 6.1.5, “Recovery options without DB2 Recovery Expert” on page 131.

6.3 Point-in-time recovery of table in a DPF environment

This section discusses how to recover a table in a multi-partitioned DB2 database.

6.3.1 Scenario, time line, and details of activity against the table

In this example, we have the same scenario as in 6.1, “Point-in-time recovery of a table after running DDL” on page 124. The scenario, time line, and activity done against the table is exactly the same but only in a DPF environment.

DPF environment

We have three logical nodes on one server. Node 0 is the catalog node. VPS.CAR is created in DMS table space DMSDAT which is defined to be in DATAGROUP database partition group. DATAGROUP includes nodes 1 and 2. Thus, VPS.CAR data only exist on nodes 1 and 2.

6.3.2 Recovering a table to point-in-time in a DPF environment

To recover a table to a point-in-time in a DPF environment, follow these steps:

1. Connect to the database from DB2 RE, which attaches to node 0 by default.
2. Go to the Object Selection window, and select the SMSDAT1 table space.
3. Select a point-in-time that is before time T2, see Table 6-1 on page 124.

Important:

- In DPF environments, always start with the catalog node.
- You must have run log analysis and updated Versioning Repository before attempting this recovery.

Figure 6-13 shows us the recovery path selected by DB2 RE.

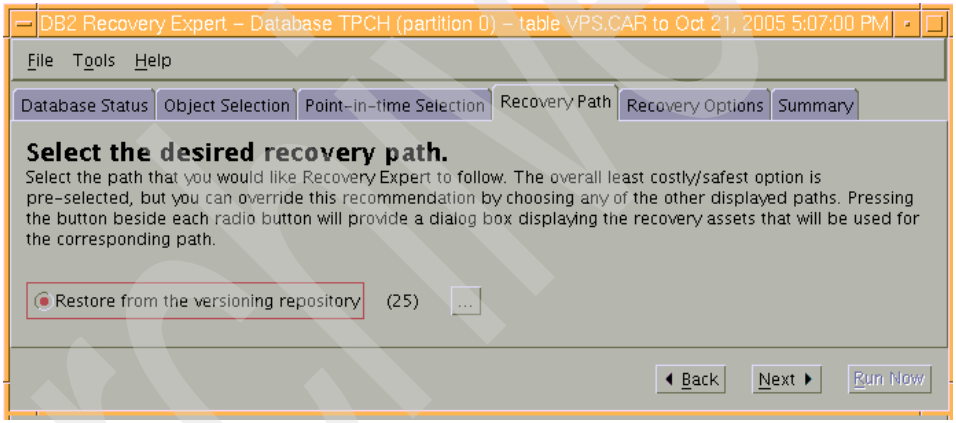


Figure 6-13 Recovery path on catalog node.

DB2 RE is only recovering the table DDL from catalog node because there is no table data on this node. Therefore, the only thing DB2 RE needs to do here is to recover table DDL from Versioning Repository.

4. Click **Next t**o go to the Recovery Options window (Figure 6-14).

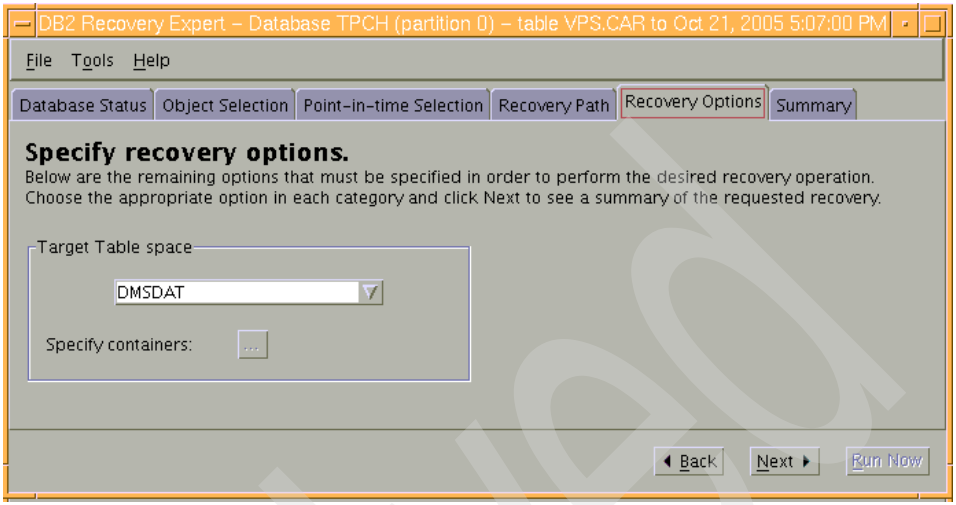


Figure 6-14 Recovery options on catalog node

5. Take the defaults here and click **Next** to go to the Summary window.
6. Click **Run Now** to execute the suggested actions (Figure 6-15 on page 141).

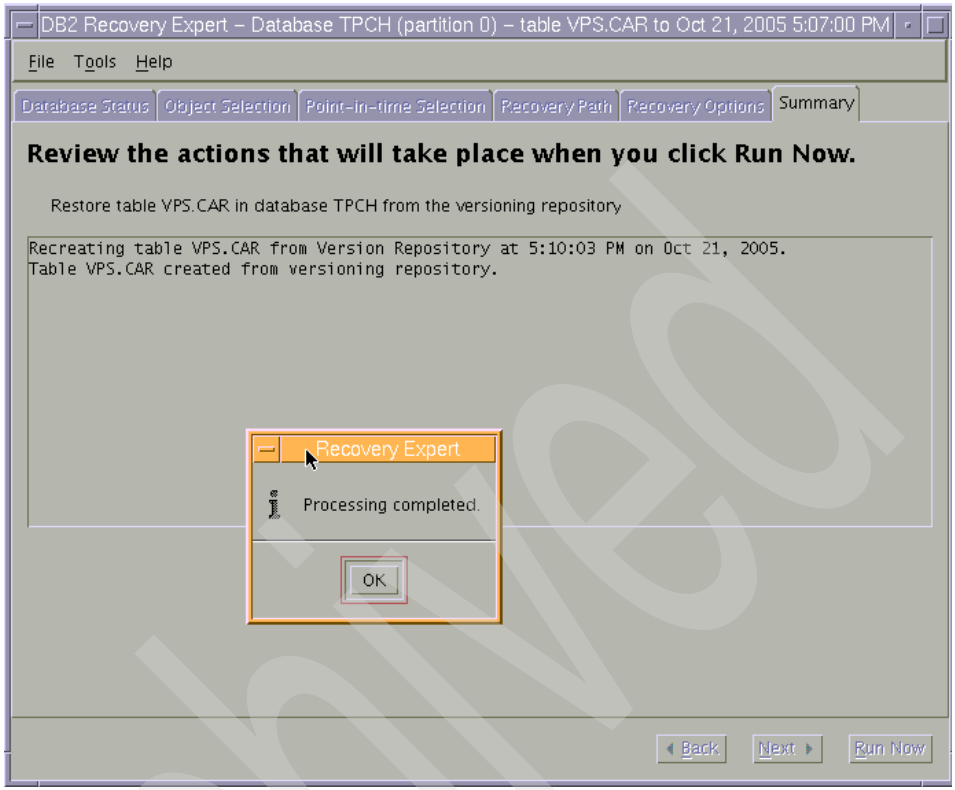


Figure 6-15 Summary of actions on catalog node

We can see that we have recovered the original DDL from the Versioning Repository and restored it.

7. We now move to DB2 partition 1. Choose **File** → **Connect** to open the connection window as shown in Figure 6-16.

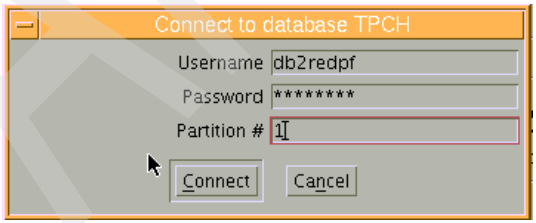


Figure 6-16 Node switching

8. For the Partition #, specify the node to which you want to connect. When you are connected to the database partition, execute log analysis to update the ALID data.

Important: Log analysis is performed against one partition at a time. In a DPF environment, it is important to remember that you have to execute log analysis on each partition that is involved in the recovery procedure, including the catalog node.

You only need to update the Versioning Repository once. It exists only on the catalog node. Thus, you do not have to update it for every node.

9. After log analysis is complete, step through DB2 RE window to obtain the suggested recovery path (Figure 6-17).

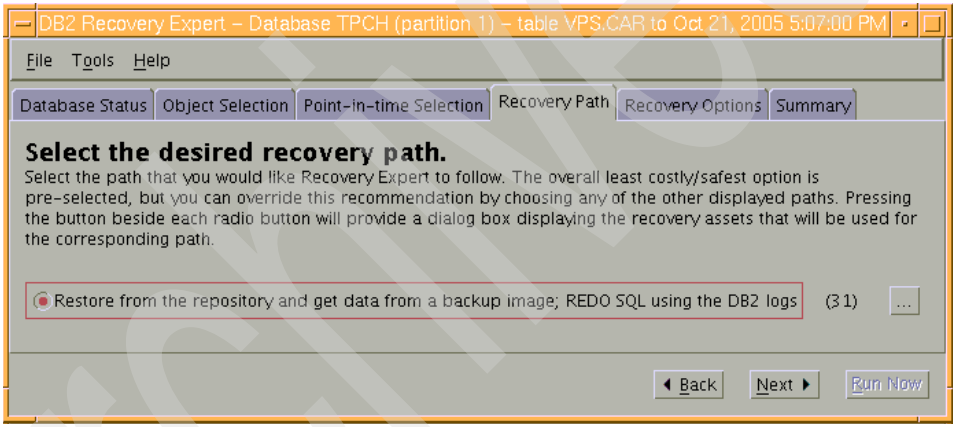


Figure 6-17 Recovery Path on partition 1.

DB2 RE is suggesting to extract data from backup and redo SQL.

10. Step through the remainder of the screens to execute the steps. Figure 6-18 on page 143 shows the results.

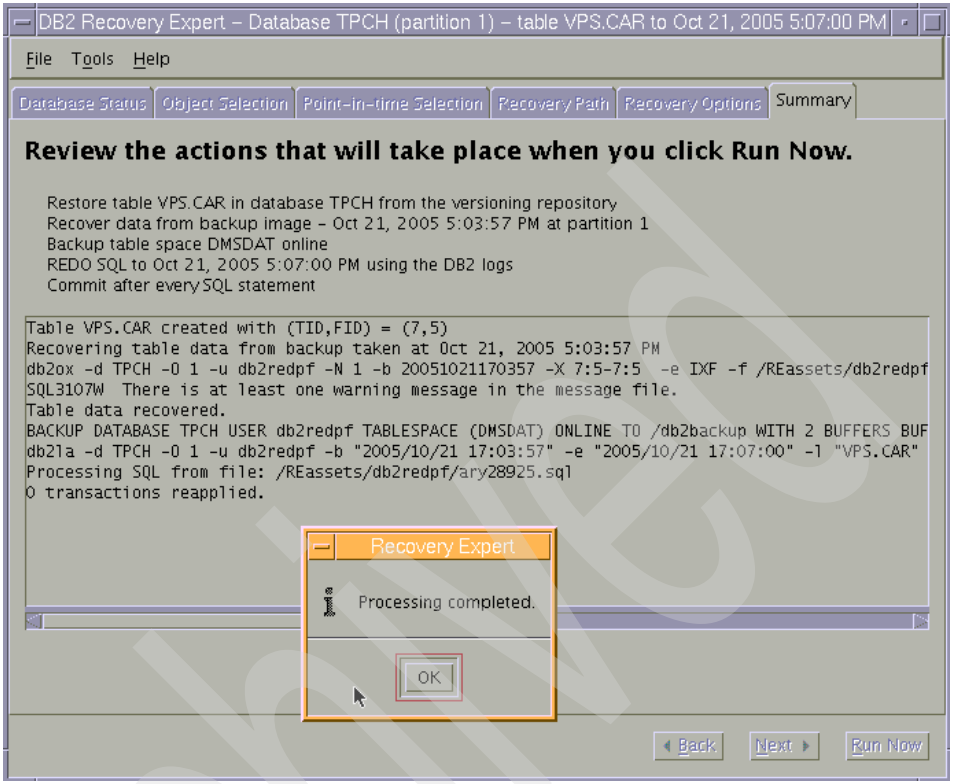


Figure 6-18 Recovery results on Partition 1.

DB2 RE extracted data from the backup using **db2ox**. Then, it loaded the data in to the table and ran a backup of the table space to bring it out of backup pending state. It ran log analysis and determined that no transactions need to be applied.

11. We now switch to node 2 and go through same process, making sure that the ALID is updated first on this partition using log analysis. Figure 6-19 on page 144 shows the Recover Path.

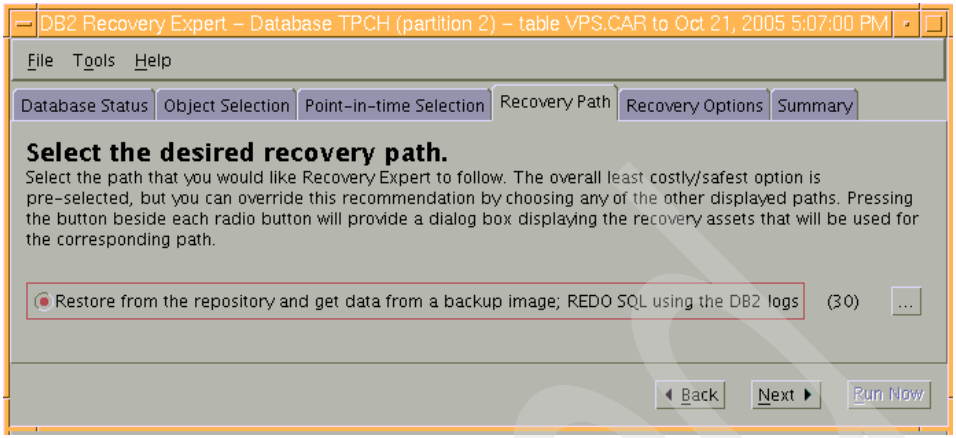


Figure 6-19 Recovery path on partition 2.

12.We step through remainder of the windows. Figure 6-20 shows the results.

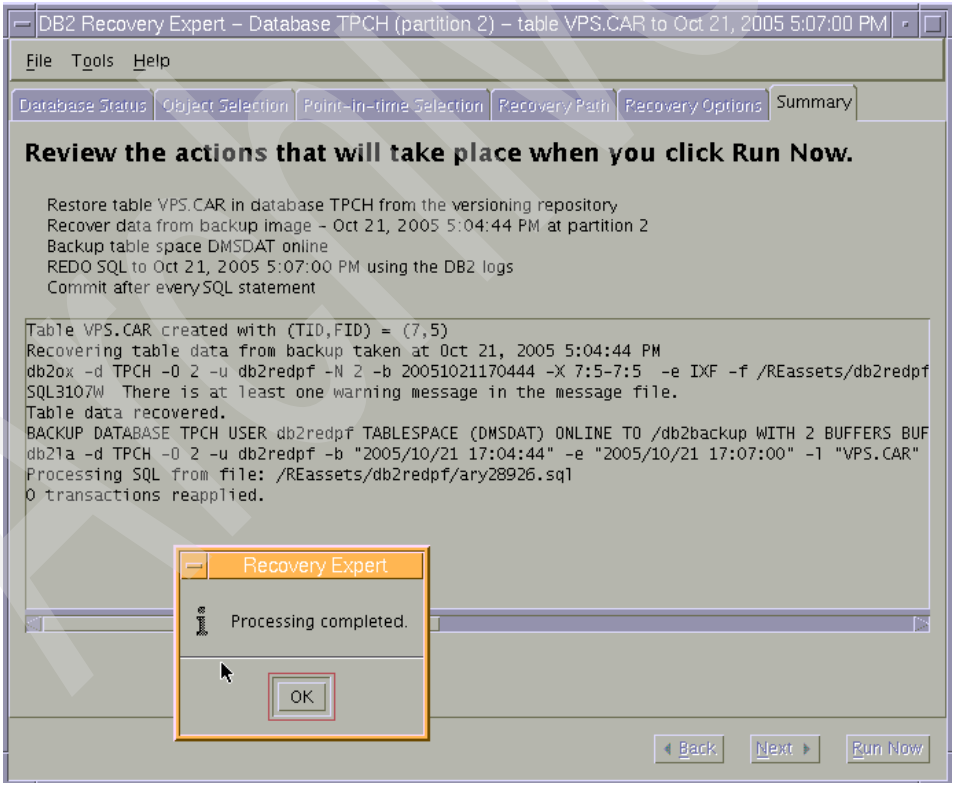


Figure 6-20 Recovery results on partition 2.

DB2 RE followed the same steps as it did on partition 1 to recover the table to a specified point-in-time. Again, we see that zero (0) transactions were redone after extracting data from the backup. This information is correct because we do not want to redo the delete transactions. Example 6-11 shows that all delete jobs have been undone.

Example 6-11 Row count after recovery

```
/home/db2redpf->db2 "select count(*) from vps.car"
1
-----------
     119986

  1 record(s) selected.
```

6.3.3 Recovery options without DB2 Recovery Expert

Recovery without DB2 RE would be time consuming and disruptive. You would have to restore all database partitions and then roll forward the database to a point-in-time before the delete job was run. In a large production environment with tens of partitions, you are looking a very long recovery time. You would also lose all transactions that were done after the given point-in-time. With DB2 RE, you would only lose the transaction for the table space after the given point-in-time.

6.4 Recovering a database to point-in-time when DDL has been run

This section discusses how to recover a database to point-in-time where DDL has been run before the given point-in-time. The prerequisites for this scenario are the same as described in 6.1.1, “Prerequisites for this scenario” on page 124.

6.4.1 Scenario time line

Table 6-3 shows the time line of activity against the database for this scenario.

Table 6-3 Time line for activity against database

Time	Action
T1	Full offline backup taken.
T2	Batch job executed by mistake, which delete operations data from VPS.CAR and VPS.CUSTOMER tables in SMSDAT1 table space and VPS.PART table in DMSDAT1 table space.
T3	Alter table DDL is run against table VPS.CAR

We want to undo the delete operations that were done by the batch job. Multiple tables have been affected in two table spaces. Therefore, we decide to perform a database level recovery.

6.4.2 Details of activity against the database

Example 6-12 shows the row counts before the batch job ran.

Example 6-12 Row count before batch job was run.

```
/db2backup->db2 "select count(*) from vps.customer"
1
-----------
     253456

  1 record(s) selected.
/db2backup->db2 "select count(*) from vps.car"
1
-----------
     119986

  1 record(s) selected.
/db2backup->db2 "select count(*) from vps.part"
1
-----------
     200000

  1 record(s) selected.
```

Example 6-13 shows row counts after the batch job was run.

Example 6-13 Row count after batch job was run

```
/home/db2insk1/yasir->db2 "select count(*) from vps.car"
1
-----------
     117925

  1 record(s) selected.
/home/db2insk1/yasir->db2 "select count(*) from vps.customer"
1
-----------
     250313

  1 record(s) selected.
/home/db2insk1/yasir->db2 "select count(*) from vps.part"
1
-----------
     195001

  1 record(s) selected.
```

Because the time batch job was run at T2, VPS.CAR was altered at time T3 by the following DDL:

```
db2 alter table vps.car add t_condition varchar(30)
```

We need to undo the delete operations done by the batch job.

6.4.3 Recovering a database to a point-in-time

To recover a database to a point-in-time, follow these steps:

1. Connect to the database from DB2 RE.
2. Go to the Object Selection window, and select **Database** for recovery (Figure 6-21 on page 148).

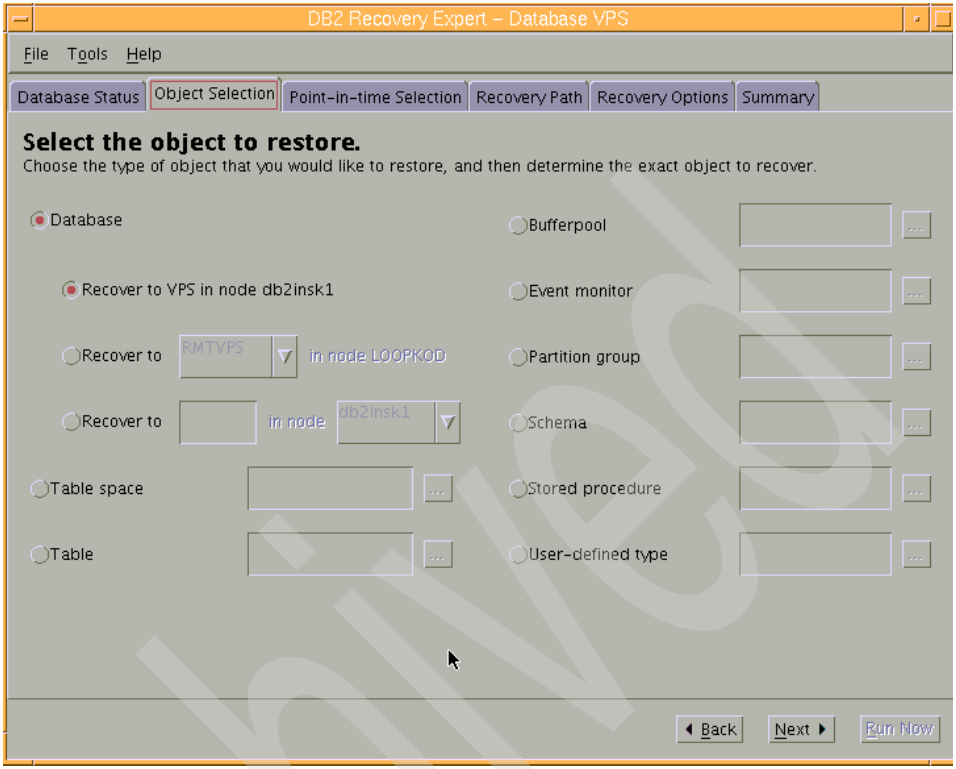


Figure 6-21 DB2 RE Object Selection window

3. Step through the windows and go to the Recovery Path window (Figure 6-22 on page 149).

> **Note:** DB2 RE is suggesting to do the recovery through native DB2 restore and roll forward because it is the most efficient method in this situation.

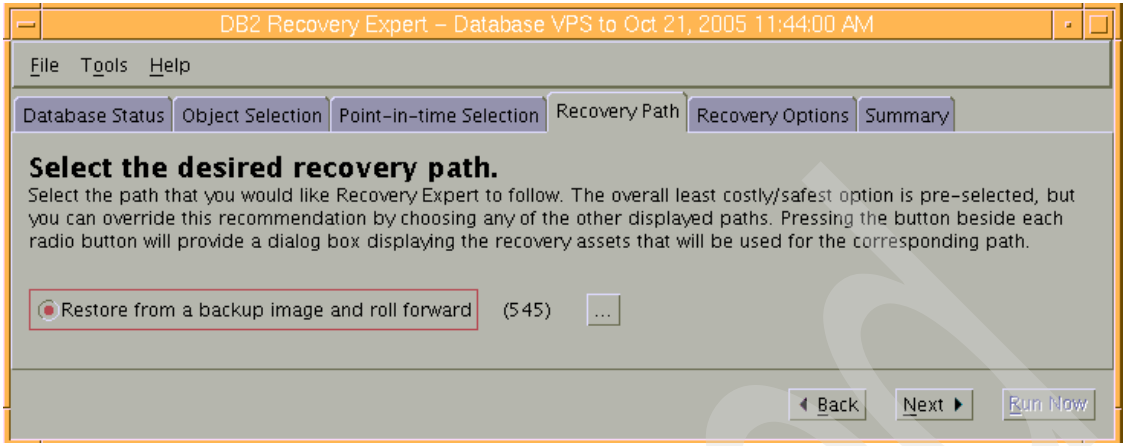


Figure 6-22 Recovery Path suggested by DB2 RE

4. Step through the windows and go to the Summary window.
5. Click **Run Now** to execute the suggested steps (Figure 6-23).

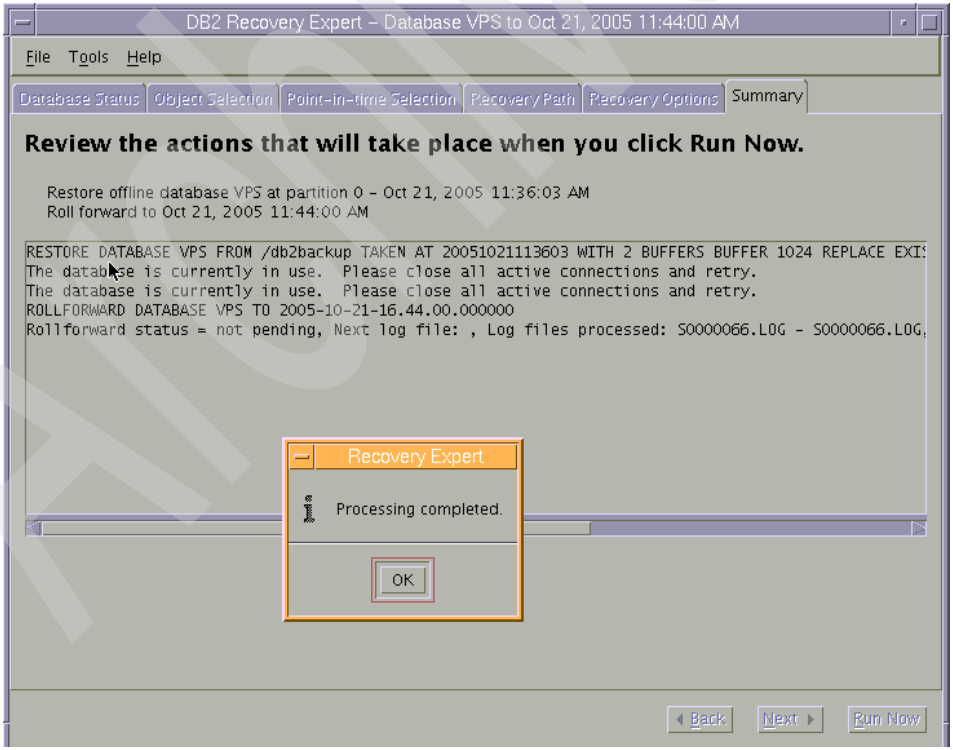


Figure 6-23 Database recovery results

The database has been recovered to the specified point-in-time and made available to users. Example 6-14 shows row counts after the recovery has completed.

Example 6-14 Row counts after database recovery.

```
/db2backup->db2 "select count(*) from vps.customer"
1
-----------
     253456

  1 record(s) selected.
/db2backup->db2 "select count(*) from vps.car"
1
-----------
     119986

  1 record(s) selected.
/db2backup->db2 "select count(*) from vps.part"
1
-----------
     200000

  1 record(s) selected.
```

6.5 Redirected restore using DB2 Recovery Expert

The DB2 documentation describes the need for redirected restore as follows:

> During a database backup operation, a record is kept of all the table space containers associated with the table spaces that are being backed up. During a restore operation, all containers listed in the backup image are checked to determine if they exist and if they are accessible. If one or more of these containers is inaccessible because of media failure (or for any other reason), the restore operation will fail. A successful restore operation in this case requires redirection to different containers. DB2 supports adding, changing, or removing table space containers.

For more information, see the following:

http://publib.boulder.ibm.com/infocenter/db2help/topic/com.ibm.db2.udb.doc/admin/c0006249.htm?

If you are using DB2 RE in a recovery situation and you have to perform a redirected restore, DB2 RE can support that.

If you have all default table space locations, it is trivial to restore a database backup into a new database, effectively making a usable copy of the database. However, if you have defined your own containers for your table spaces and if you try to restore the backup to a new database name, it fails with the message `SQL0294N The container is already in use`. DB2 is trying to use the same container paths again. To get around this issues, you must do a redirected restore.

6.5.1 Prerequisites for the scenario

In this scenario, we are simply making a new copy of a database from an offline backup. Because the original database and its table space containers still exist, the new table space containers must be given new names via redirected restore.

We have one instance, and we want to make a new database in that same instance by restoring from an existing backup of another database. We do not need to roll forward to another point-in-time, the time of the backup is sufficient. Our data tables are in two table spaces, one SMS and one DMS, which have containers on specific paths on the system. We also have the standard default table spaces (SYSCATSPACE, USERSPACE1) in the default locations. The backed up database also has a Versioning Repository for the original database.

6.5.2 Scenario time line

This scenario does not have a time line. It only illustrates how to use the redirected restore option in DB2 RE in case your recovery scenario requires it. In our scenario, we do not attempt to roll forward, although there is no restriction for doing so in a redirected restore.

Redirected restore without DB2 Recovery Expert

You can perform redirected restore in the DB2 Control Center or from the command line. We do not discuss those methods here.

Redirected restore using DB2 Recovery Expert

The lab environment for this scenario is as follows:

- Linux, DB2 UDB V8 FP9a
- Server: Lead
- instance: db2insl1
- database: RECART

To perform a redirected restore using DB2 RE, follow these steps:

1. Start DB2 RE and select the database from which you want to copy, as shown in Figure 6-24.
2. Click **Next** to proceed to Object Selection.

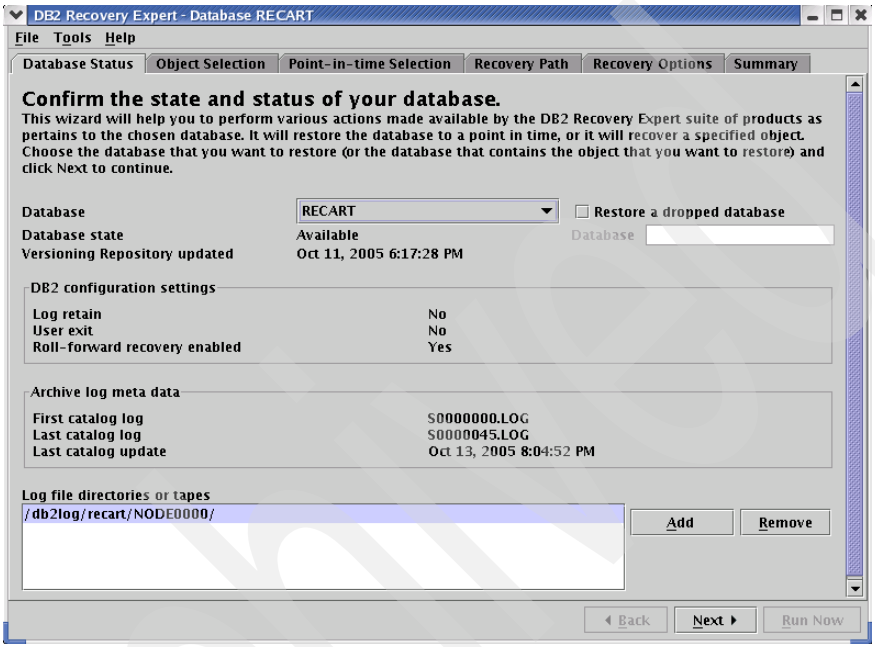


Figure 6-24 Connect to database

3. Choose to restore into a new database, called NEWCART, in the same instance, db2insl1, as shown in Figure 6-25.
4. Click **Next** to proceed to the point-in-time selection.

Note: The database selection area of the Object Selection window includes options for Node selection. The items in the pull-down lists correspond to your local DB2 node directory. DB2 RE does not actually support remote restore or recovery, so you should not assume that you can restore to another system, it will not work. For now, you should *only* select the local node from the list. The local node is always the default, so best just to leave it alone.

While the Log Analysis feature *does* support remote log analysis, you do not need to use the Object Selection screen for that. Object Selection is only for recovery scenarios.

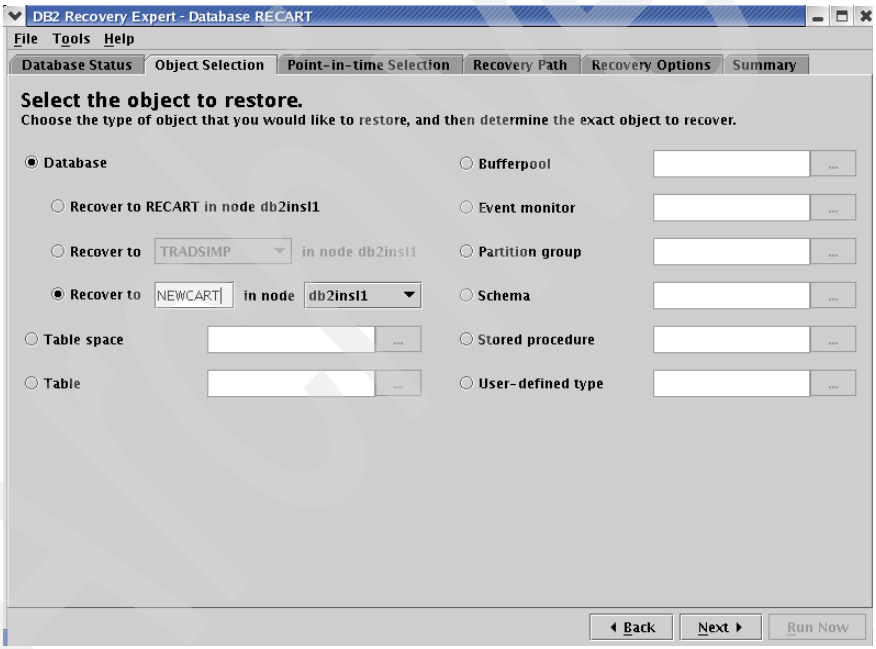


Figure 6-25 Object selection, recover to a new database

5. In Figure 6-26, choose a backup from 11 October at 18:07:59 as the point-in-time.

   The different point-in-time options always resolve back to a specific date and time. You are not necessarily selecting the backup or the version. You can verify the exact backup that is used by drilling down on the Recovery Path window.

6. Click **OK** to accept the point-in-time.

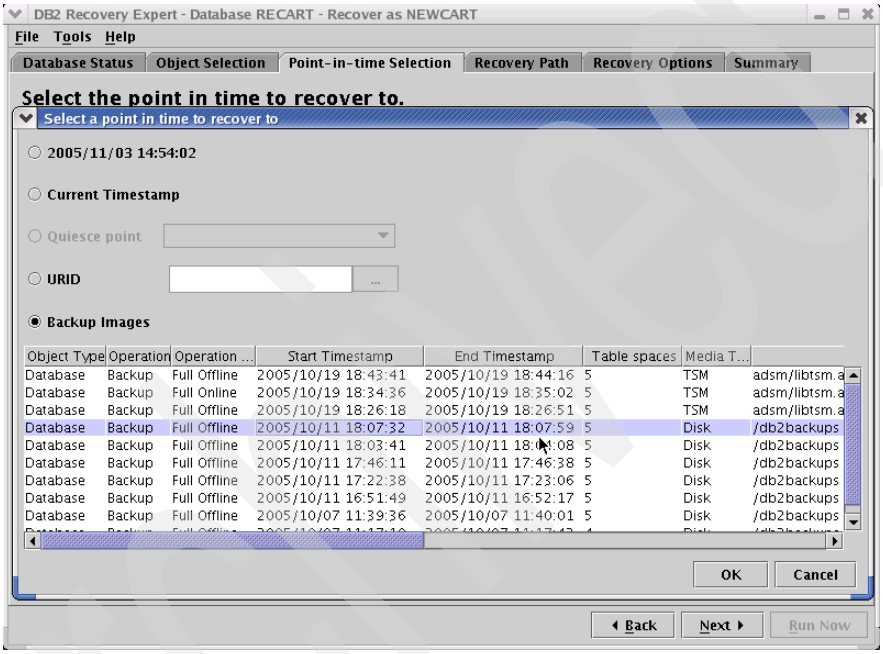


Figure 6-26 Point-in-time selection, choose a backup as point-in-time

In Figure 6-27, we now see the backup's point-in-time has been resolved on the Object Selection window.

7. Click **Next** to proceed to the Recovery Path window.

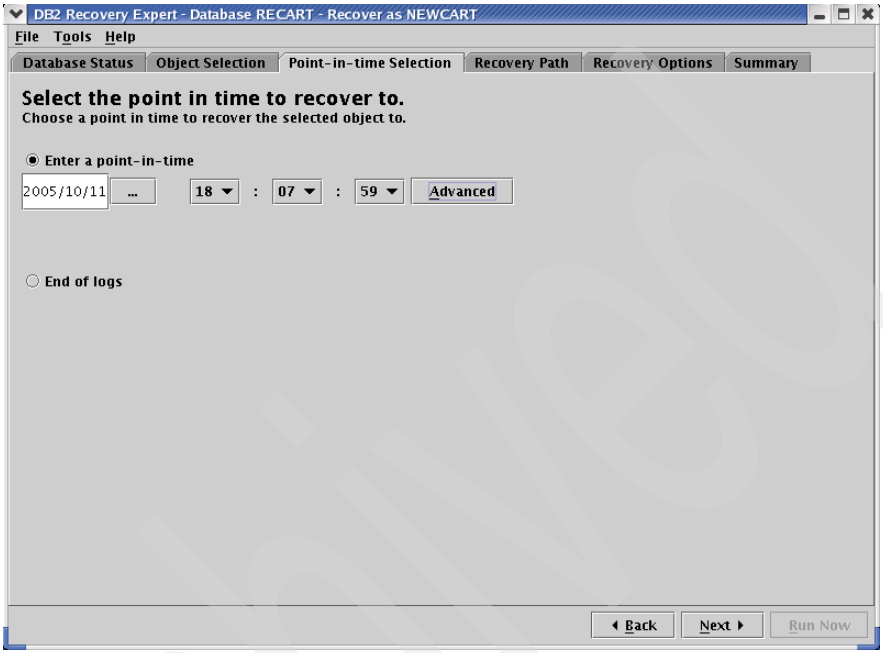


Figure 6-27 Point-in-time selection complete

8. In Figure 6-28, we see the Recovery Path window. We are offered only one recovery path in this case, which is to restore from a backup. Accept the default by clicking **Next**.

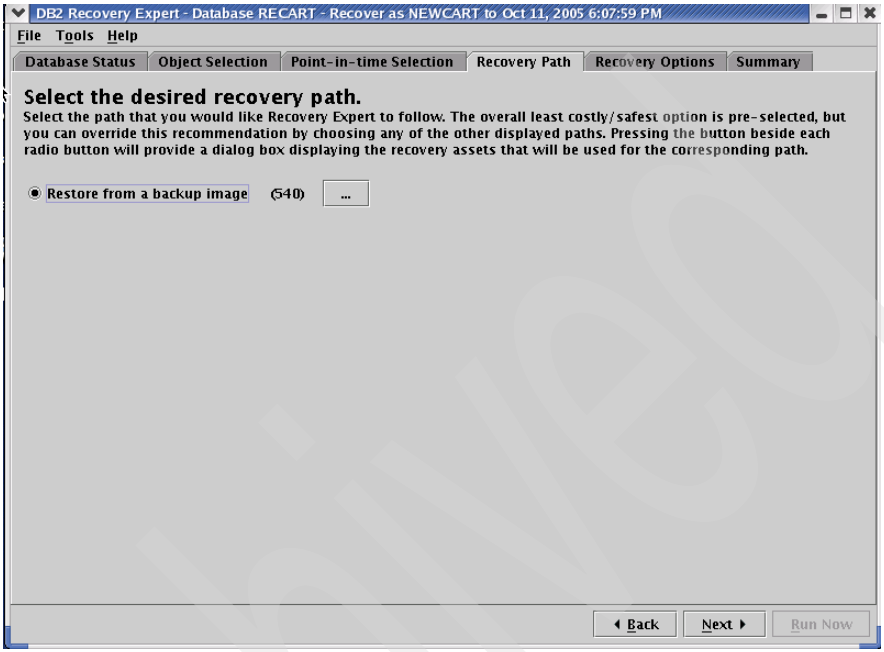


Figure 6-28 Recovery Path, simple restore from a backup

9. In Figure 6-29, we see the Recovery Options window. There are several options here, but we are interested in the redirected restore. Select the box and click the browse icon ("...") to open the redirected restore table space containers window (Figure 6-30 on page 158).

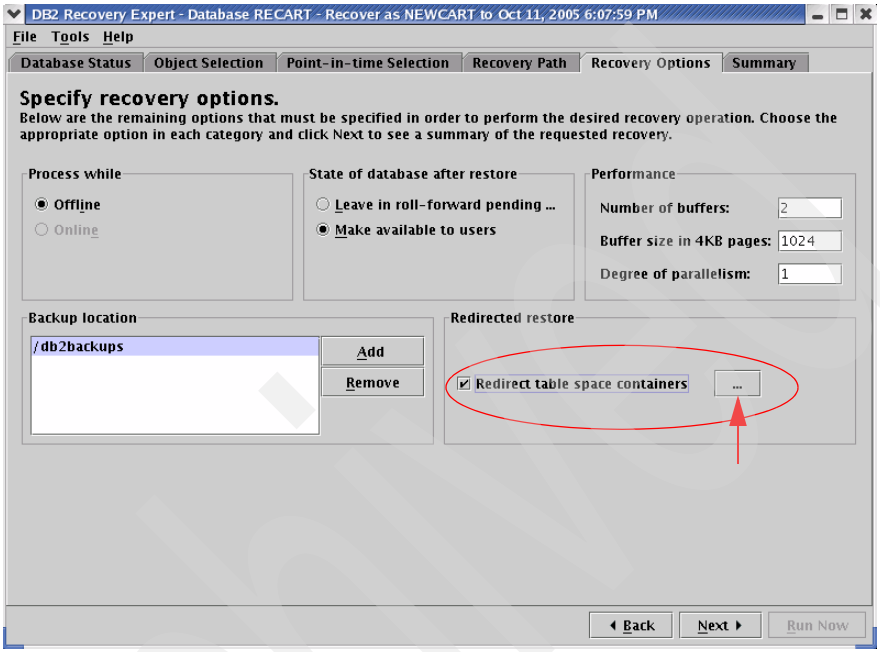


Figure 6-29 Recovery options, choose redirect

In Figure 6-30 on page 158, we see the Redirect table space containers window as it first opens. All the table spaces are listed, showing their current definitions as they are described in the backup. In this case we have five table spaces. We need to redirect only two of them, RECART and USERPRODUCTTS. The other table spaces are defined as a default so they are recreated properly when the new database is created.

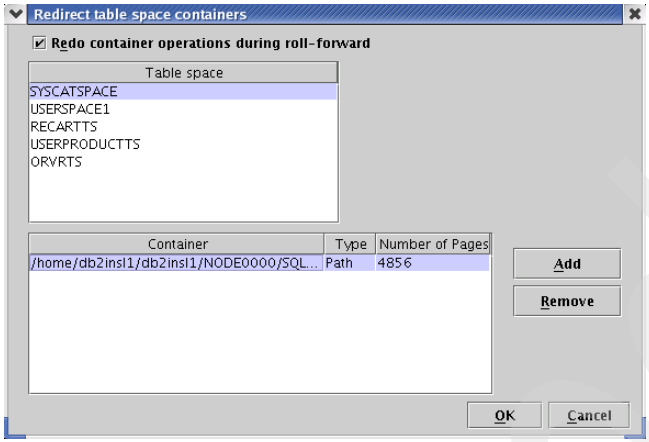


Figure 6-30 Initial Redirect table space containers dialog

> **Tip:** For DB2 RE redirected restore, you must take action on each table space, either by telling DB2 RE to ignore it or to redirect it. You should perform the steps in the sequence described here. In our tests, sometimes the OK button was disabled if we did the steps in a different sequence.

10. Click **Remove** for any table space containers that you do not want to redirect. In this scenario, this is everything in SYSCATSPACE, USERSPACE1, and ORVRTS. See Figure 6-31 on page 159.

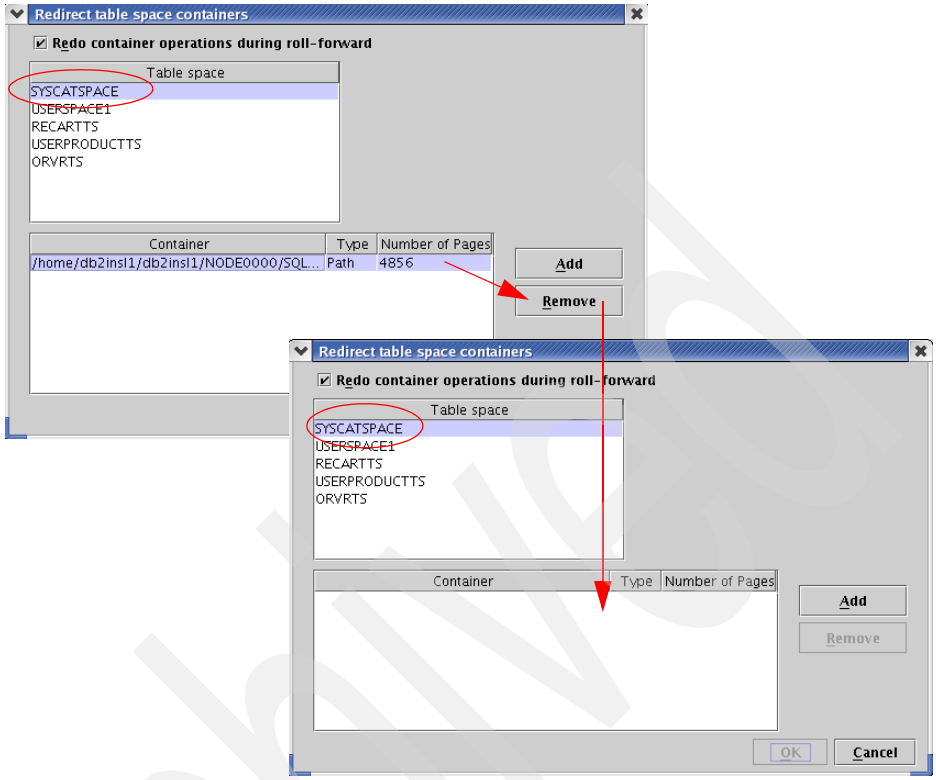


Figure 6-31 Steps to remove container

11. Click **Add** for each table space that you want to redirect. In this scenario, we want to add RECARTTS and USERPRODUCTTS.

    We recommend that you add the new container specification first, and then remove the old one. This gives you chance to compare the before and after.

    We illustrate this in Figure 6-32 on page 160, where we add a new container for the RECARTTS table space, which is SMS. For DMS it is a bit different, because you have to also tell what size to make the file or raw device (we used file container). We show this in Figure 6-33 on page 161.

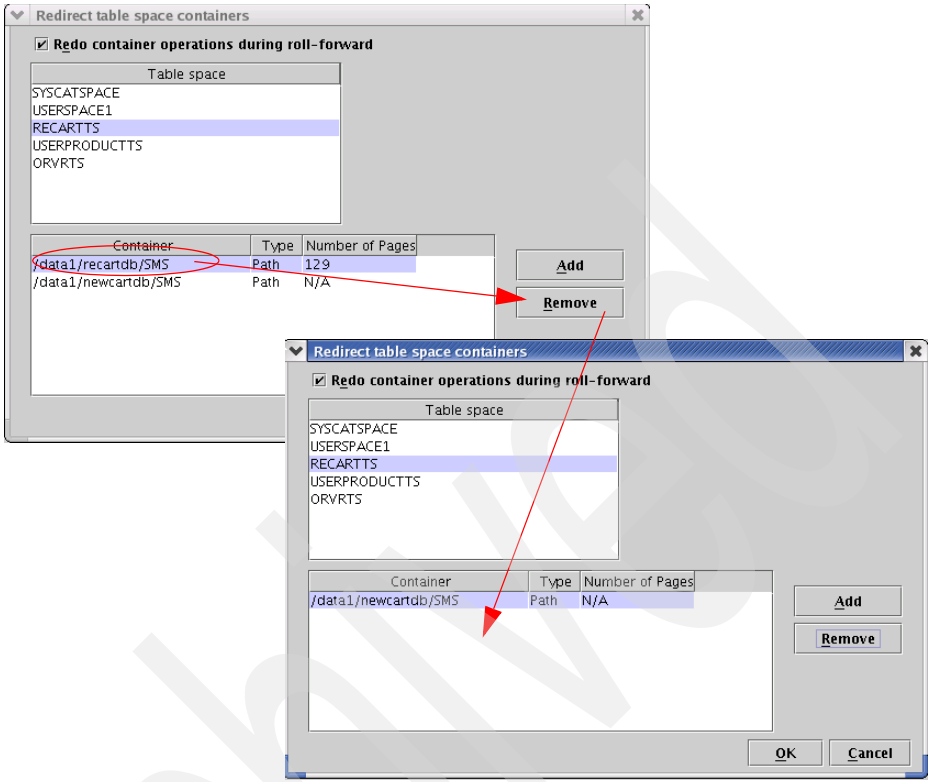


Figure 6-32 Steps to add new and remove old SMS container

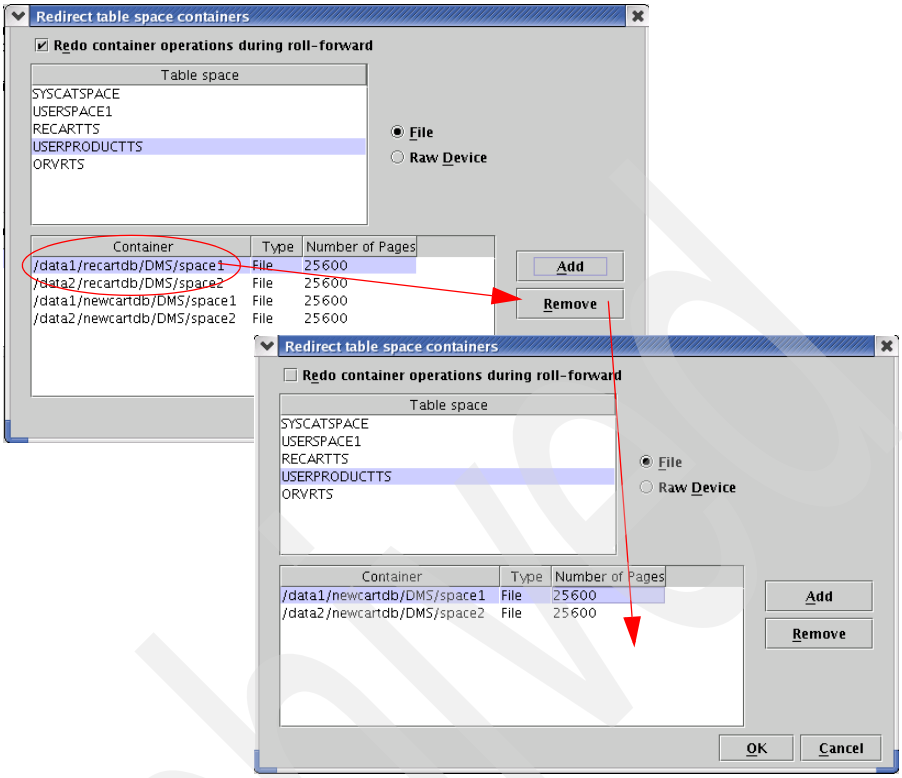


Figure 6-33 Steps to add new and remove old DMS containers

When you use the Add (Path browser) window, shown in Figure 6-34 on page 162, you cannot just type in a new path name that does not exist. DB2 RE might accept your input, but the resulting path is invalid and is not created.

You can create the new directory or path as necessary within the dialog by using the new folder icons (different in Windows and UNIX). If you create the path outside of DB2 RE, be sure that you have the correct permissions on the path. If you create the path from the DB2 RE dialog, the permissions are set correctly.

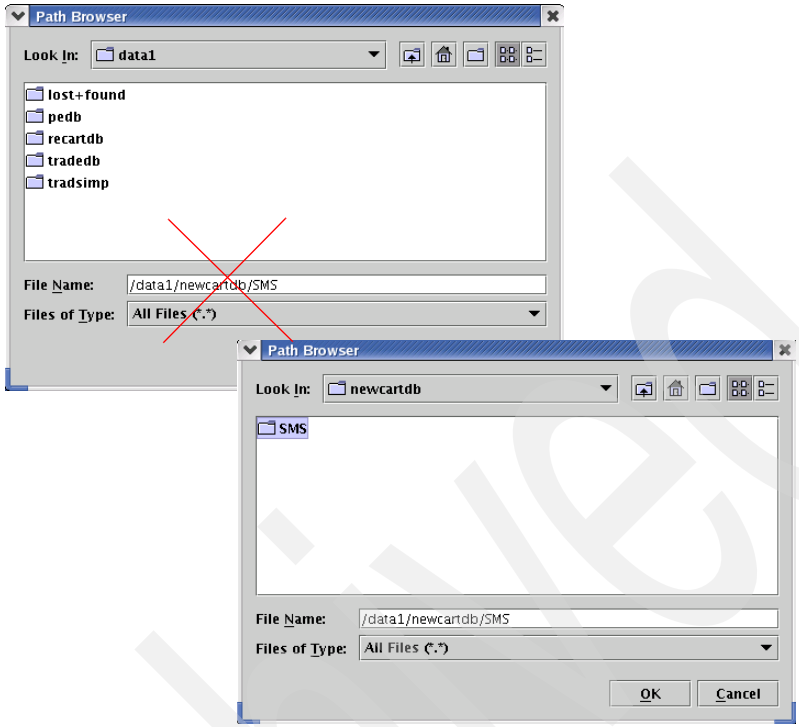


Figure 6-34 New container path must exist

12. When the new containers are defined, now click **Remove** for the old containers. This is shown in Figure 6-32 on page 160 and Figure 6-33 on page 161.

    We deselected the box for Redo container operations during rollforward. This box corresponds to IGNORE or REPLAY ROLLFORWARD CONTAINER OPERATIONS option of the `SET TABLESPACE CONTAINERS` command. We do not want to preserve any DDL in this case, plus we are not rolling forward anyway. You might want to consider this option and its effects if you need to use DB2 RE to do a redirected restore.

13. When you are satisfied with the new table space container paths, click **OK** to return to the Recovery Path window.

14. Click **Next** to proceed to the DB2 RE summary window to review the restore operation (Figure 6-35 on page 163).

> **Note:** The DB2 commands are not shown here until you initiate the restore, so you should be extra careful that all your redirected containers are correct on the previous screens.

15. Click **Run Now** to begin the restore.

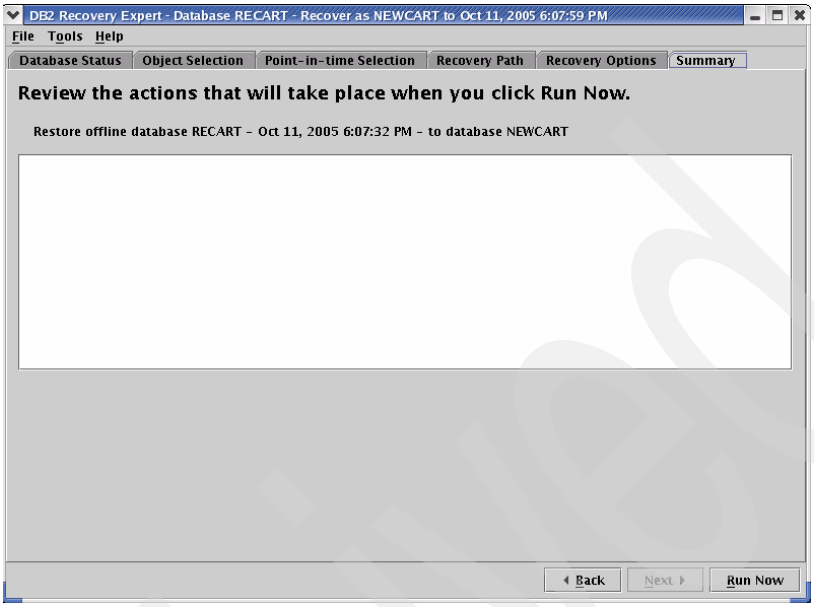


Figure 6-35 Restore summary window

While the restore is running, you see the commands that are executed (Figure 6-36). A progress box also opens but is not shown here.

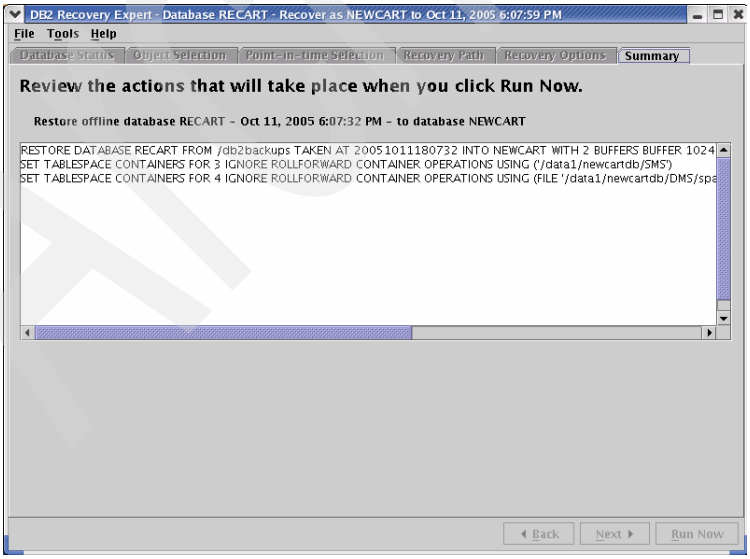


Figure 6-36 Restore in progress

When the restore is finished, a message opens, as shown in Figure 6-37.

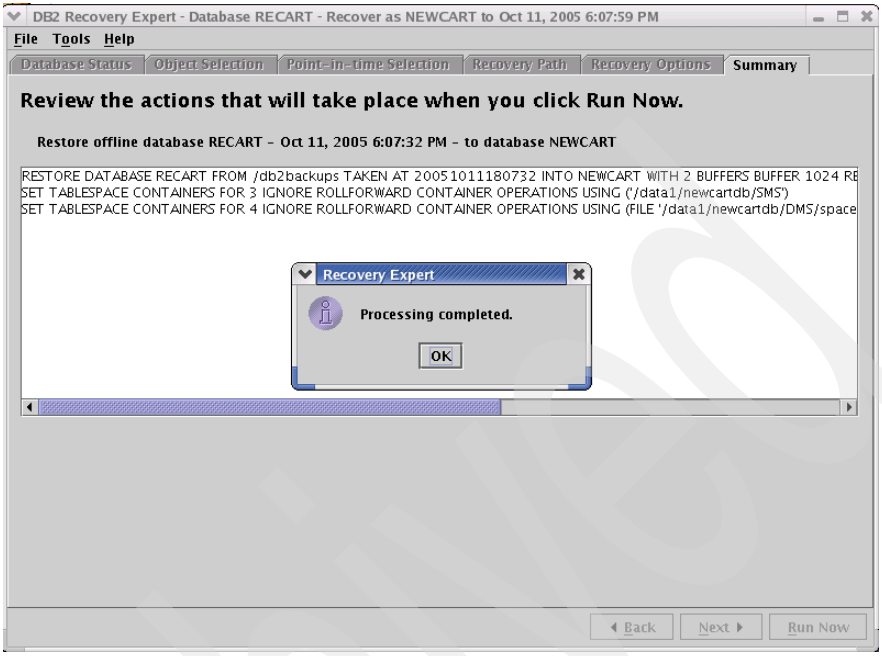


Figure 6-37 Redirected restore is complete

When you create a new database by restoring a backup from another database, you must consider the state of the other objects inside the database. You can ask questions such as:

- Is there a Versioning Repository to rebuild?
- Are there DB2 Grouper sets to redefine?
- Are there other objects that may refer to the old database not the new one?

In our case, we only have a Versioning Repository from the old database, RECART, which contains information that is relative to that database. So, we need to rebuild the Versioning Repository for the new database, NEWCART. We perform the steps for rebuilding the Versioning Repository as described in 4.2.7, “Rebuilding the Versioning Repository” on page 68.

We should make a backup of the new database after rebuilding the Versioning Repository.

Because we want to use the new database with DB2 RE, we also need to rebuild the ALID. The steps to do this are described in 4.4, “Archive Log Inventory Data (DB2 RE metadata)” on page 77.

Example 6-15 shows the table space container detail information to verify that our new containers were defined properly.

Example 6-15 Verify NEWCART table space containers

```
[db2insl1@lead db2insl1]$ db2 connect to newcart

   Database Connection Information

 Database server        = DB2/LINUX 8.2.2
 SQL authorization ID   = DB2INSL1
 Local database alias   = NEWCART

[db2insl1@lead db2insl1]$ db2 list tablespaces

...

 Tablespace ID                        = 3
 Name                                 = RECARTTS
 Type                                 = System managed space
 Contents                             = Any data
 State                                = 0x0000
   Detailed explanation:
     Normal

 Tablespace ID                        = 4
 Name                                 = USERPRODUCTTS
 Type                                 = Database managed space
 Contents                             = Any data
 State                                = 0x0000
   Detailed explanation:
     Normal

 Tablespace ID                        = 5
 Name                                 = ORVRTS
 Type                                 = System managed space
 Contents                             = Any data
 State                                = 0x0000
   Detailed explanation:
     Normal

[db2insl1@lead db2insl1]$  db2 list tablespace containers for 3 show detail

           Tablespace Containers for Tablespace 3

 Container ID                         = 0
 Name                                 = /data1/newcartdb/SMS
 Type                                 = Path
 Total pages                          = 97
```

```
 Useable pages                       = 97
 Accessible                          = Yes

[db2insl1@lead db2insl1]$  db2 list tablespace containers for 4 show detail

           Tablespace Containers for Tablespace 4

 Container ID                        = 0
 Name                                = /data1/newcartdb/DMS/space1
 Type                                = File
 Total pages                         = 25600
 Useable pages                       = 25568
 Accessible                          = Yes
 Container ID                        = 1
 Name                                = /data2/newcartdb/DMS/space2
 Type                                = File
 Total pages                         = 25600
 Useable pages                       = 25568
 Accessible                          = Yes
```

6.6 Point-in-time recovery with referential integrity and DB2 Grouper

DB2 Recovery Expert V2 now has integration with the IBM DB2 Grouper component. DB2 Grouper allows you to discover and define relationships between database objects. DB2 RE can use this knowledge during recovery to detect the relationships and to recover related objects automatically.

For information about how to install the DB2 Grouper component, see Chapter 10, “Installing and using the DB2 Grouper component” on page 305.

While DB2 RE now provides a smoother recovery for related objects, it does not free you completely from thinking about your recovery strategy. For example, there might still be triggers in place that can be affected by running undo or redo SQL. This section shows a simple scenario to introduce the mechanism for how DB2 RE uses DB2 Grouper during a recovery.

6.6.1 Scenario description

In this scenario, we use the Trade6 application, which is a WebSphere application that is available to the public and that is useful for doing WebSphere performance benchmarks. We use the Trade6 application only to demonstrate a somewhat more real-life application. You can learn more about the Trade6 application at:

http://www.ibm.com/software/webservers/appserv/was/performance.html

We perform some simple tasks in the Trade6 application such as registering a new user, buying and selling some securities, and making a user profile change. Then, we want to use DB2 RE to back out to a point-in-time, but the undo will affect several tables. We intend to show how the DB2 Grouper integration will help us do this.

6.6.2 Prerequisites and environment for the scenario

This scenario uses the Trade6 WebSphere application to drive transactions to a DB2 database. Our WebSphere server is at V6.0.1 and runs on a Windows server™ named SIAM. The Trade database is named TRADSIMP and runs on a Linux server named LEAD. We use a Firefox browser on Windows to execute the transactions as a user would.

We modified the database that comes with the Trade6 application in the following ways:

- Named the database TRADSIMP
- Created two DMS table spaces instead of using default
- Created two buffer pools instead of the default
- Altered all tables to include `DATA CAPTURE CHANGES`

We made the following database configuration changes:

```
db2 update db cfg for tradsimp  using logfilsiz 2000
db2 update db cfg for tradsimp  using logprimary 8
db2 update db cfg for tradsimp  using logsecond  5
db2 update db cfg for tradsimp  using applheapsz 1024
db2 update db cfg for tradsimp  using newlogpath /db2log/tradsimp
db2 update db cfg for tradsimp  using logarchmeth1 "DISK:/db2logarch"
```

We made the following instance registry change:

```
db2set DB2_DETAIL_LOGGING=AUTHID
```

6.6.3 Time line for scenario

The times shown in Table 6-4 are the times that we captured from the browser as displayed to us by the Trade application. These are unlikely to match the DB2 transaction times but can be used as a guide for examining the URID when we get to that part of the scenario.

Table 6-4 DB2 RE with DB2 Grouper time line

Time	Event
	Backup taken
17:59:44	Register new user ITSOUSER1
18:08:42	Logoff
18:09:52	Login as ITSOUSER1
18:11:49	Buy 175 shares of s:4
18:14:35	Logout
18:16:55	Login as ITSOUSER1
18:18:27	Update user profile
18:21:58	Buy 2222 share of s:200
18:22:50	Buy 400 shares of s:1
18:23:32	Buy 300 shares of s:0
18:28:01	Sell 175 shares of s:4
18:30:46	Logout

Figure 6-38 shows the state of the portfolio for user ITSOUSER1 just after selling 175 shares of s:4.

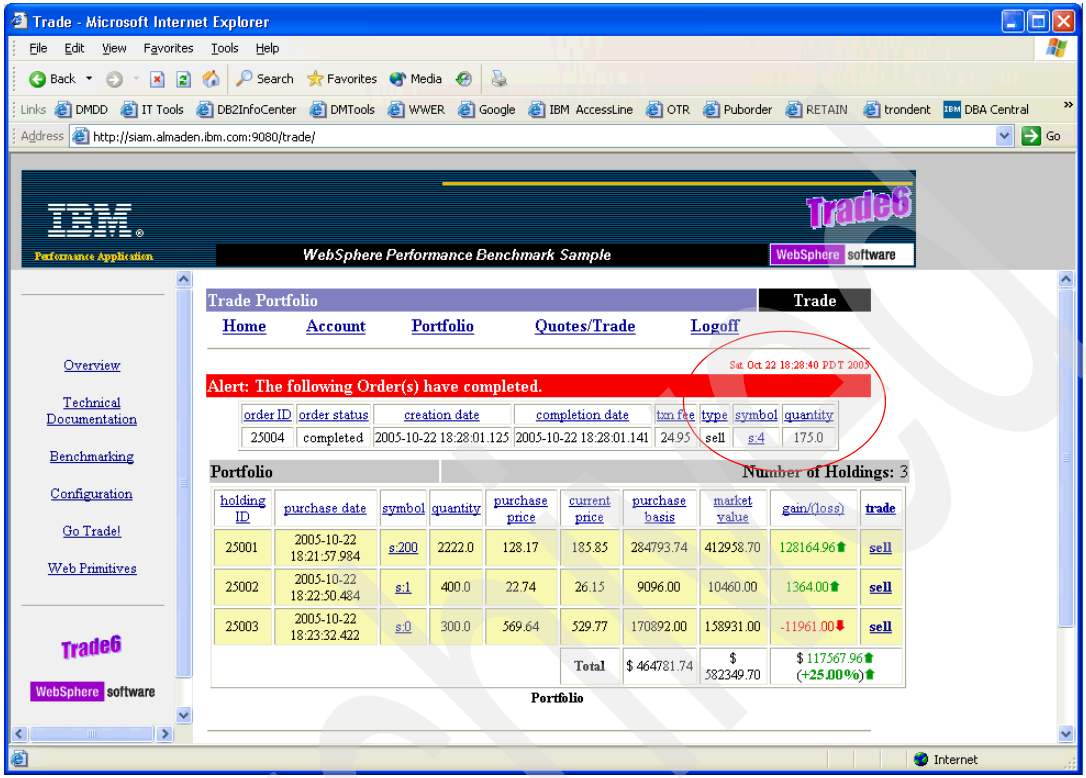


Figure 6-38 ITSOUSER1's portfolio showing sale of s:4

We find that we need recover the database back to the time just before the sale of those 175 shares of s:4. We know that several tables are involved in a sell transaction. Our first thought is to just use log analysis to find the rows to undo. We run log analysis for the time period of 17:45:00 through 18:36:00, and we find more than a few tables were affected, as shown in Figure 6-39 on page 170.

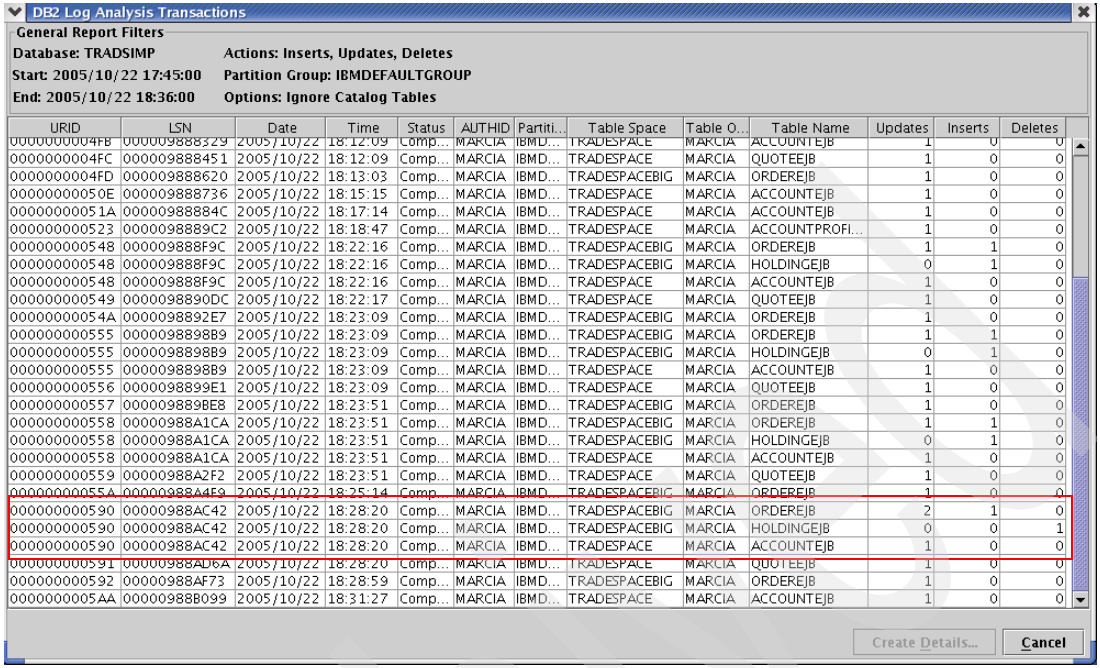


Figure 6-39 Log analysis detail for time window

We do know that we want to start by recovering the ACCOUNTEJB table, and we see that was involved in the transaction URID 0000000590, which is closest to the 18:28:01 we saw in Table 6-4 on page 168 for the time of the sale. But we also notice two other tables were modified in that same unit of work. We decide to try the new DB2 Grouper integration with DB2 RE instead of straight log analysis undo.

Before we can attempt a recovery, we need to create a group set in DB2 Grouper for the TRADSIMP tables. (We describe how to do that in Chapter 10, “Installing and using the DB2 Grouper component” on page 305, so we will not repeat that information here.) The key point for this scenario is that these tables are not related via DB2-enforced referential integrity, we set up the relationships manually in DB2 Grouper.

DB2 Grouper provides three methods of identifying relationships:

- DB2-enforced referential integrity
- Non-enforced (or manual) referential integrity
- Unit of work discovery

DB2 RE currently does not support DB2 Grouper sets from the unit of work discovery, so we do not discuss that here. Even if you do not use detailed

referential integrity in your tables, you may still have tables that would need to be recovered together. It is easy to define these relationships in DB2 Grouper, which makes using DB2 RE easier as well.

The DB2 Grouper set does not need to exist prior to the time of the transaction you want to undo. It only needs to exist prior to running DB2 RE to do the recovery.

Figure 6-40 shows the final DB2 Grouper set that we use during this scenario. For the details on how we created the set, see 10.3, “Using the DB2 Grouper component” on page 324. As you can see there are a number of different relationships between the tables.

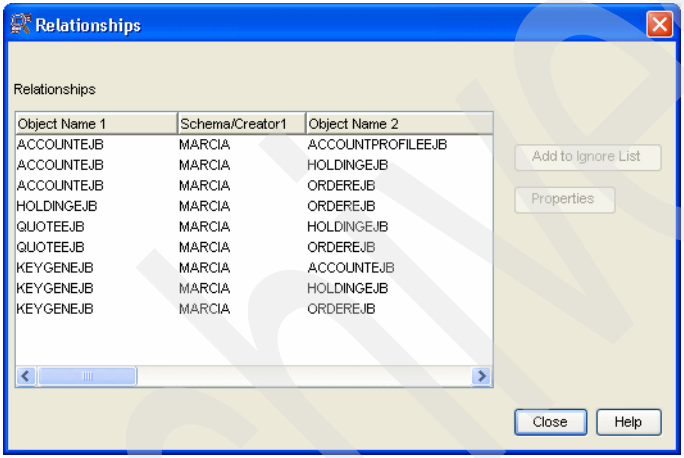


Figure 6-40 TRADSIMP table relationships

6.6.4 Recovery steps

The recovery steps for a DB2 Grouper scenario are the same as for any other type of recovery, until the first table has been recovered. If DB2 RE detects the existence of DB2 Grouper and that your table has related objects, DB2 RE asks if you want to recover the related objects. If you answer yes, you cycle through the same recovery steps for each of the other tables in turn. You always have the option not to recover a table even though it might be related to one already recovered.

In our scenario, we start by recovering the ACCOUNTEJB table. We know this is the parent table and should be recovered first. Follow these steps:

1. Start DB2 RE and connect to the TRADSIMP database, shown in Figure 6-41 on page 172.

2. Click **Next** to proceed to the Object Selection panel.

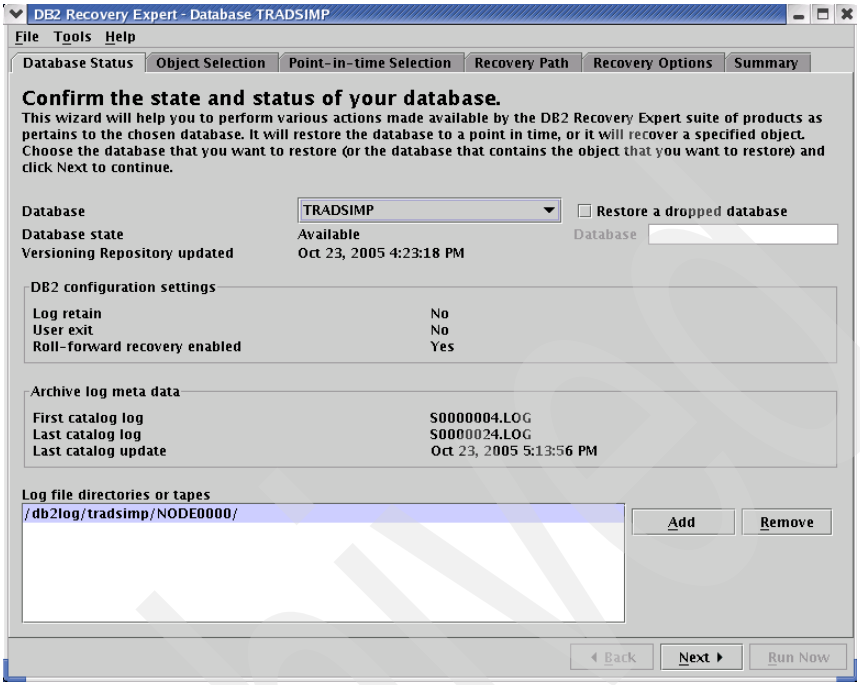


Figure 6-41 Connect to TRADSIMP

3. On the Object Selection window, open the table chooser, and select the ACCOUNTEJB table, shown in Figure 6-42 on page 173. Notice the Show Related button is active because DB2 RE has detected DB2 Grouper installed on the database.
4. Click **Show Related** to see the relationships, shown in Figure 6-43 on page 173.

Note: The relationships shown in Figure 6-43 appear twice, because we ran the group discovery twice without making a new version. If we scroll to the right on that window, we see there is a version end time listed for the outdated entries. This was an oversight on our part when creating this scenario for the book.

5. Click **OK** to dismiss the relationship viewer, and click **OK** to accept the table selection for the ACCOUNTEJB table.

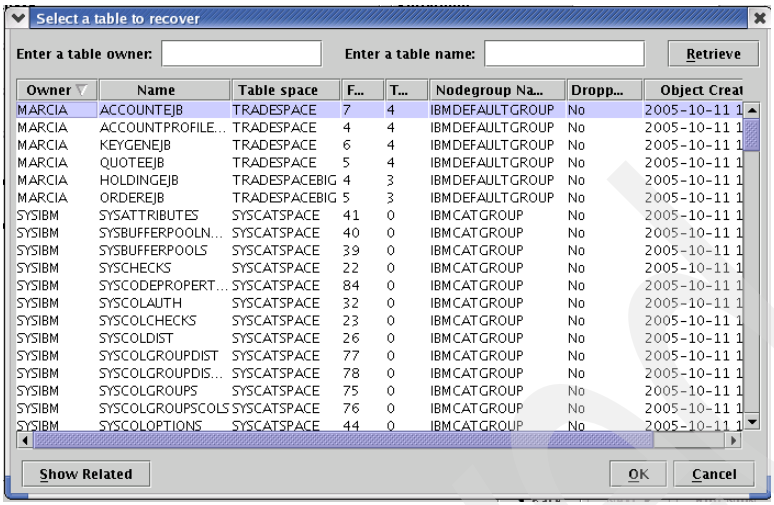


Figure 6-42 Table selection

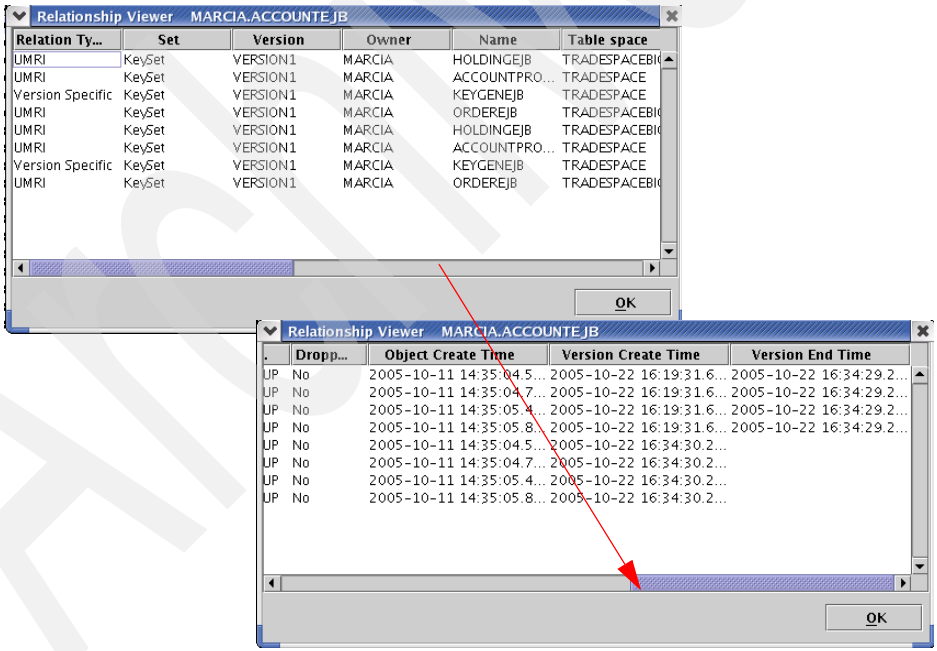


Figure 6-43 Relationship viewer (scrolled right)

6. After selecting the ACCOUNTEJB table, now we are ready to select the point-in-time. Click **Next** to move to the point-in-time window, and then click **Advanced** to open the point-in-time selection window.

We could type in the 0000000590 URID that we found earlier. However, we decide to bring up the URID finder by enabling the URID radio button and clicking the browse icon ("...").

The URID finder looks very much like the log analysis window but it is specific to the one table we selected before, the ACCOUNTEJB table, see Figure 6-44. We know the time window is somewhere between 6:00 p.m. and 6:30 p.m. So, we enter times into the begin and end times with a little cushion on either end. It never hurts to update the ALID, so we select that box as well.

7. Click **OK** to begin the analysis.

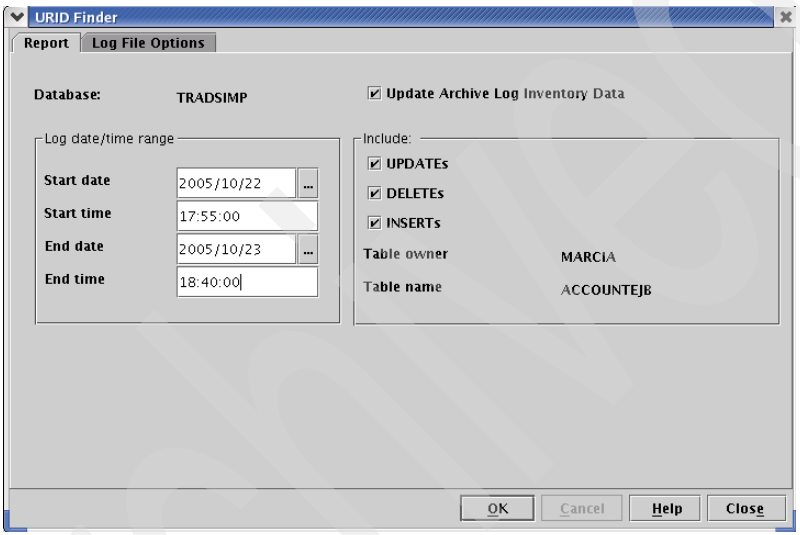


Figure 6-44 URID Finder - before

In Figure 6-45 on page 175, we see the results of the URID search, showing only the update for the ACCOUNTEJB table in URID 0000000590. The time on that URID is 18:28:20.

8. Select that URID, and click **OK** to return to the Advanced point-in-time window.

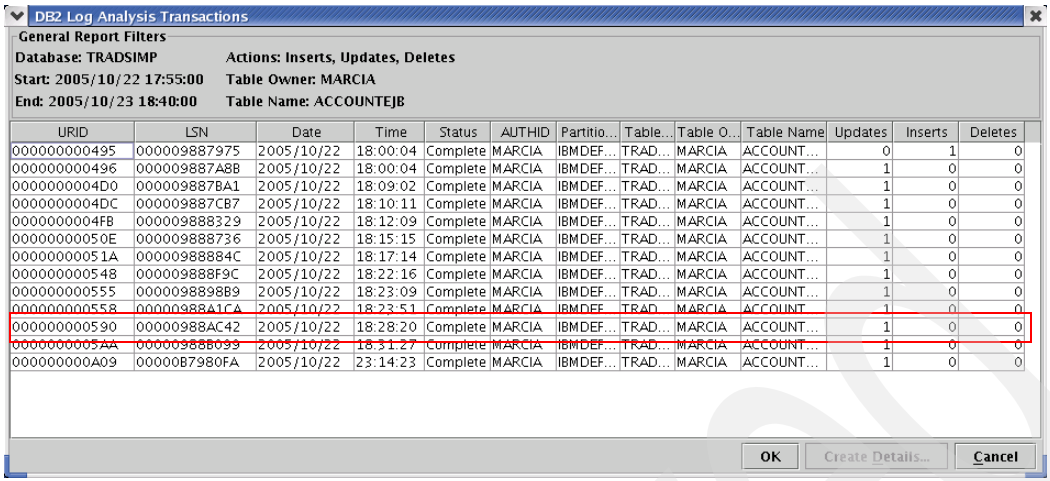


Figure 6-45 URID finder results for ACCOUNTEJB table

Figure 6-46 shows now that our URID is chosen.

9. Click **OK** to return to the point-in-time Selection window. The time stamp of that URID is 18:28:20.

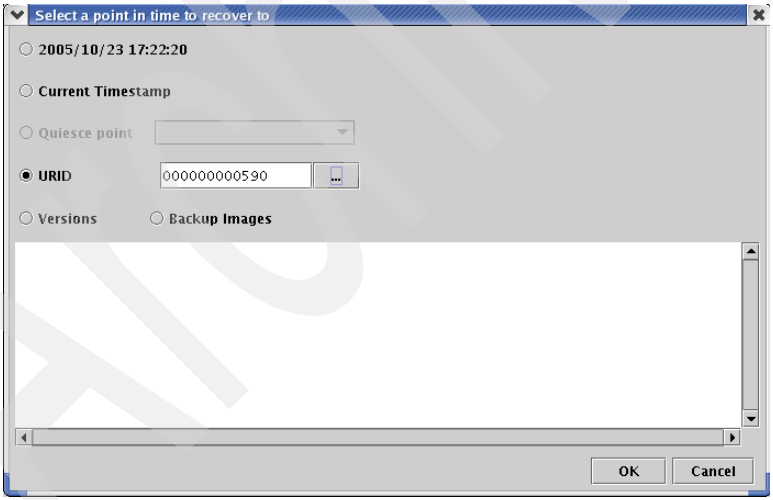


Figure 6-46 URID 0000000590 is chosen

10. Verify the point it time is correct, we see it is 18:28:20 in Figure 6-47 on page 176. Click **Next** to proceed to the Recovery Path window.

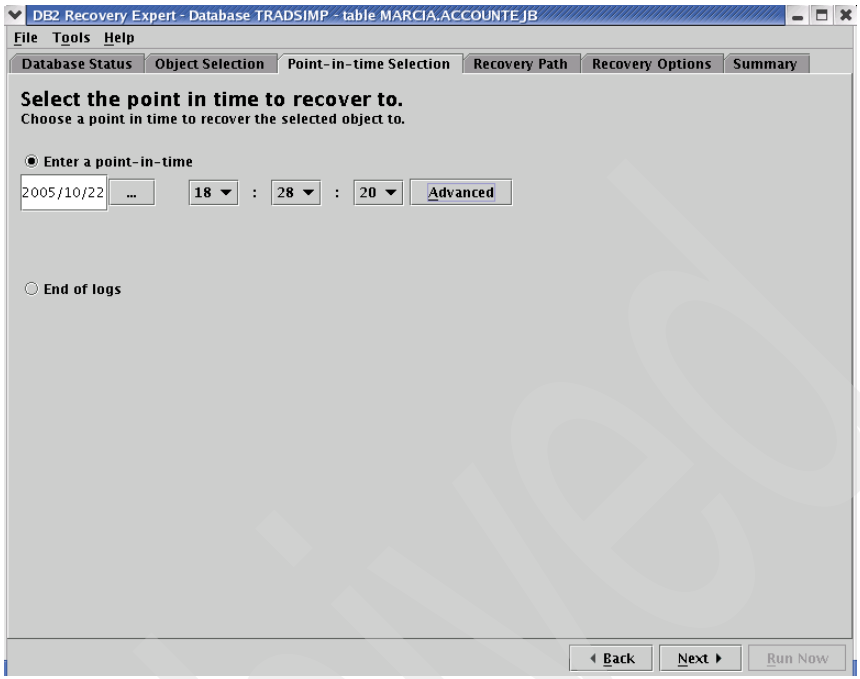


Figure 6-47 URID converted to point-in-time

DB2 RE suggests two recovery paths, we choose the simple undo path, which is the one pre-selected for us, see Figure 6-48 on page 177.

11. Click **Next** to proceed to the Recovery Options window.

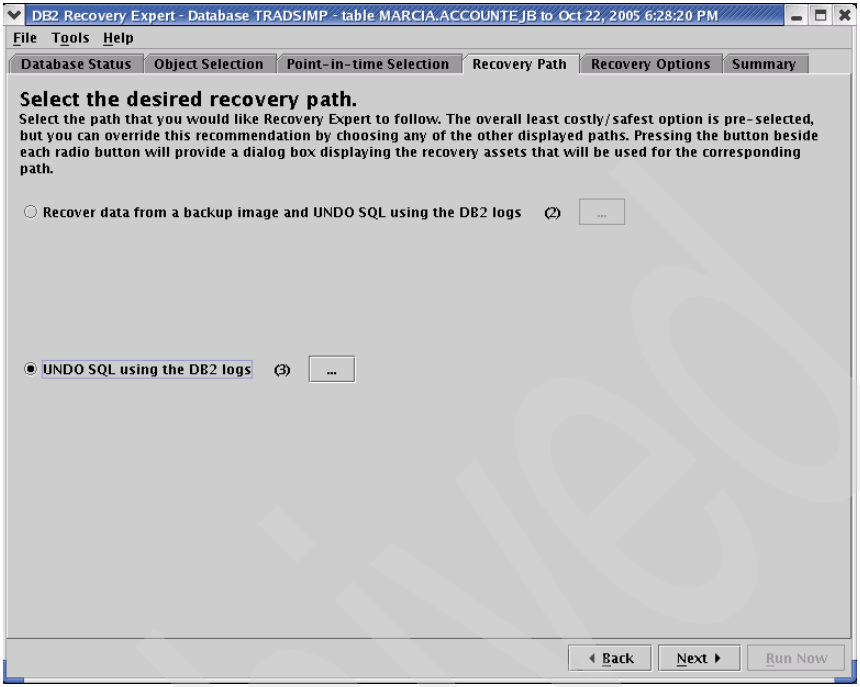


Figure 6-48 Recovery Path choose UNDO

12. Accept the defaults on the Recovery Options window, Figure 6-49 on page 178. Click **Next** to proceed to the Summary window.

Figure 6-49 Recovery Options accept default

The Summary window shows what DB2 RE plans to do to recover the ACCOUNTEJB table. In this case it will undo the SQL to the point-in-time we selected, see Figure 6-50 on page 179.

13. Click **Run Now** to restore the table. Up to here, these steps are the same as any other simple table recovery.

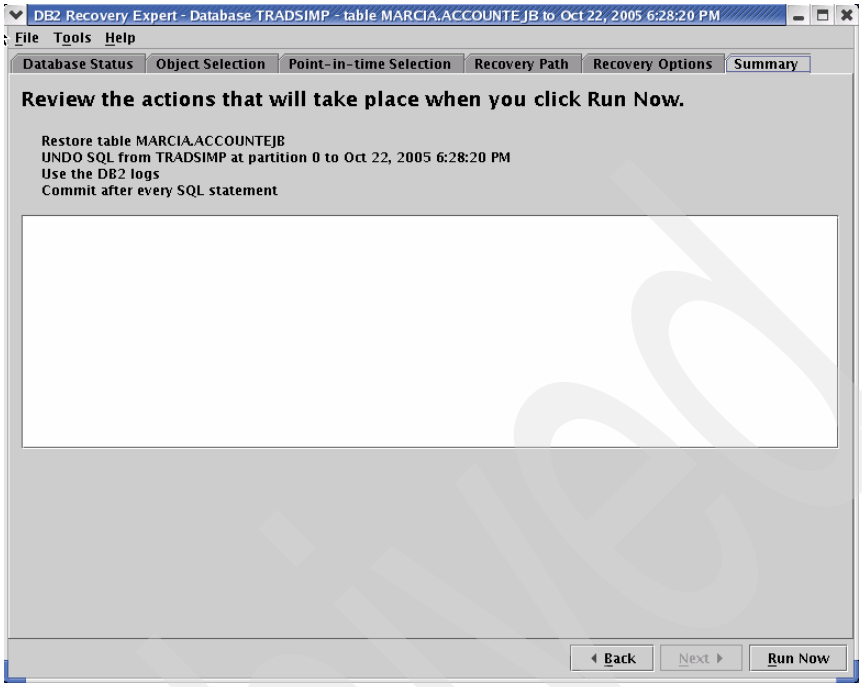


Figure 6-50 Summary of restore for ACCOUNTEJB table

In Figure 6-51 on page 180, we see that three changes to the ACCOUNTEJB table have been backed out. In a non-DB2 Grouper scenario, that would be the end of the recovery. However, we used DB2 Grouper to identify other tables related to ACCOUNTEJB.

14. Click **OK**.

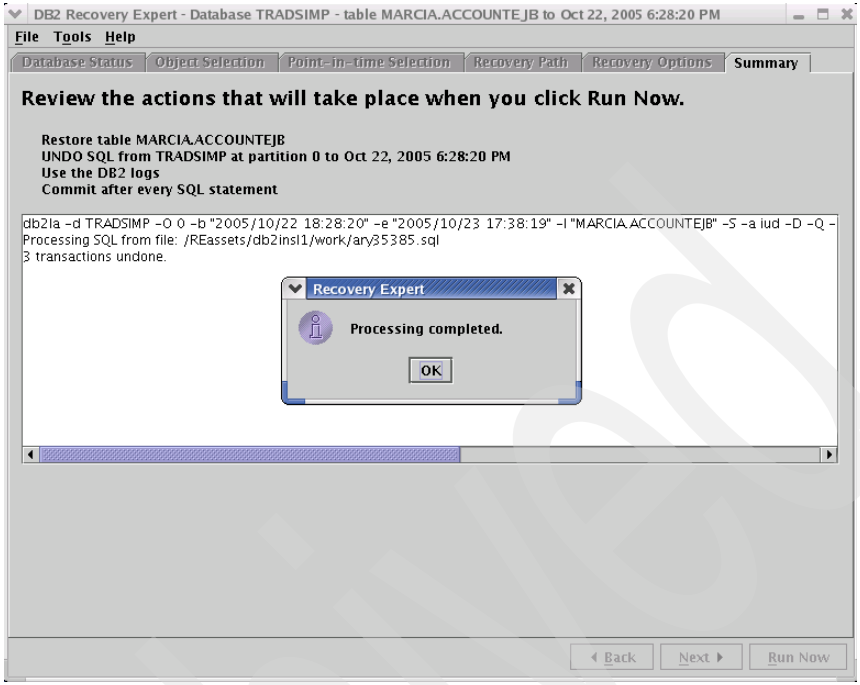


Figure 6-51 Table ACCOUNTEJB is restored to point-in-time

15. In Figure 6-52 on page 181, DB2 RE now asks if you want to recover tables that are related to ACCOUNTJEJB. Click **Yes**.

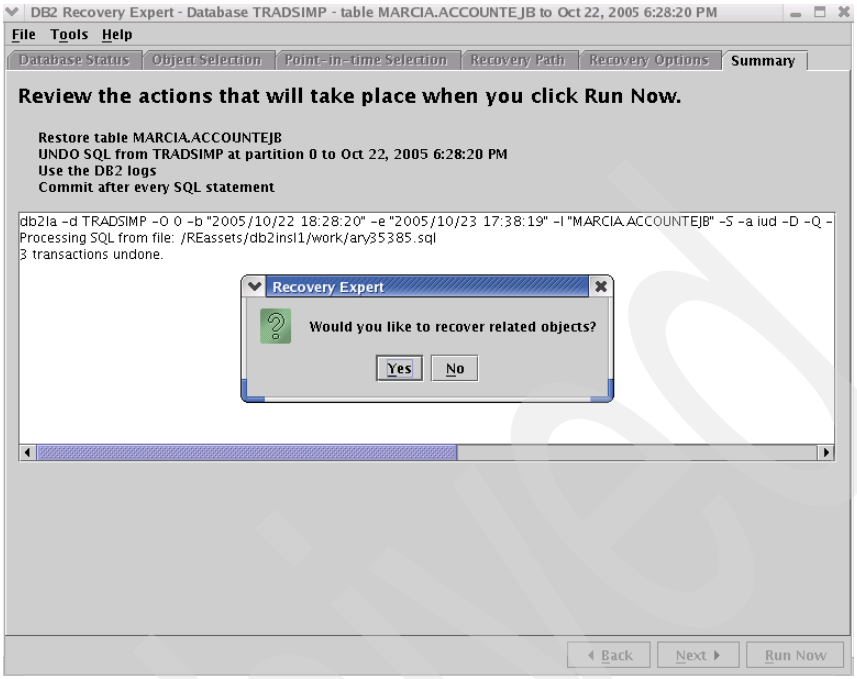


Figure 6-52 Recover related objects after restoring parent table

16. You must choose the group set and version which were previously created in DB2 Grouper.

    We did not give very informative names to our group, this is a lesson learned for the future. You should give a meaningful name to the Group and version. We only have one set and one version in this case, so we choose the one offered to us, see Figure 6-53 on page 182.

17. Click **OK**.

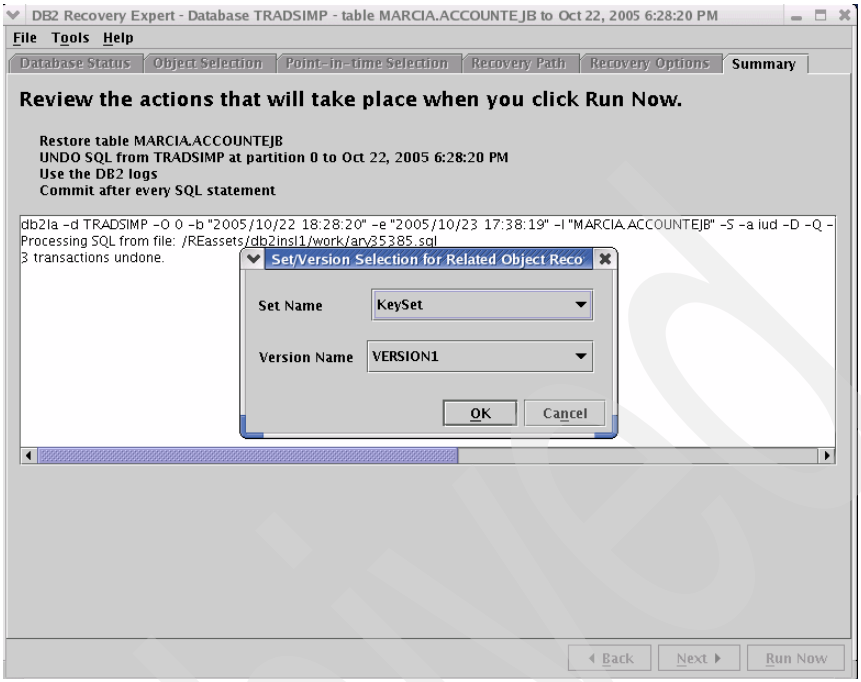


Figure 6-53 Choose DB2 Grouper group set

After you select the DB2 Grouper group, it knows what other tables are involved, and will cycle through the list of tables and ask you if you want to recover each table. It offers one table at a time, and only have one opportunity to answer. As we see in Figure 6-54 on page 183, the next table we can recover is ACCOUNTPROFILEEJB.

18. Click **Yes** to recover the ACCOUNTPROFILEEJB table.

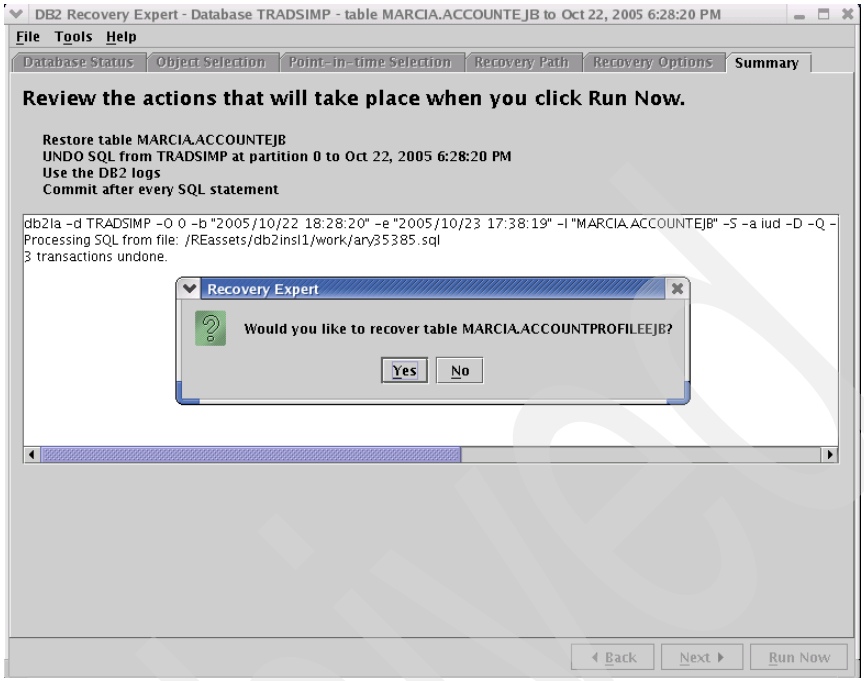


Figure 6-54 Choose to recover the next table in the group, ACCOUNTPROFILEEJB

19. When you select the next object, DB2 RE does not immediately just recover it using the same selections you made for the previous table. Instead, it knows you might want to use different options or path for recovering the table. It keeps the same point-in-time selection, but puts you back on the Recovery Path window for each table in turn. We see in Figure 6-55 on page 184 that we again have two path choices. Choose the simple UNDO path.

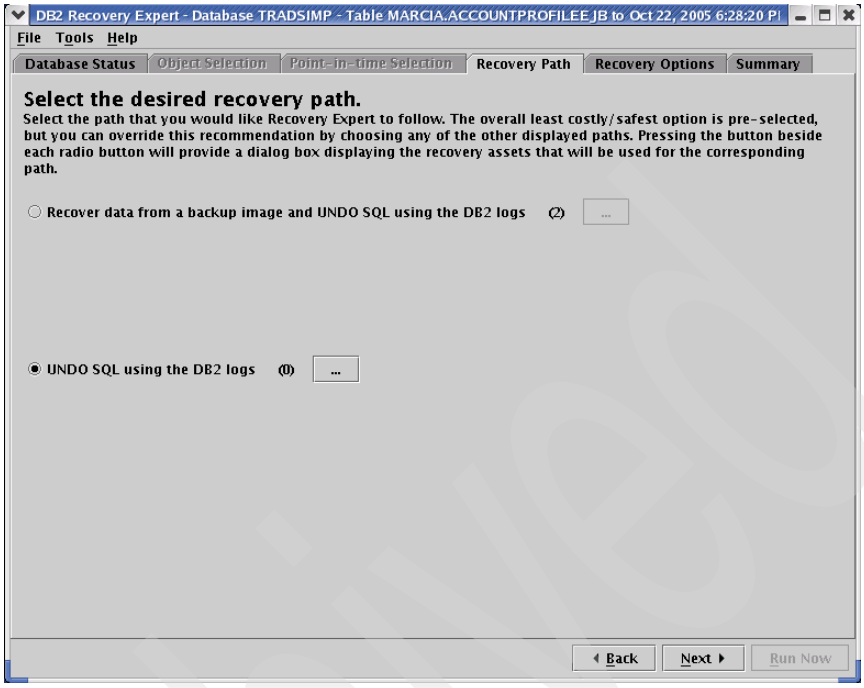


Figure 6-55 Recovery Path for ACCOUNTPROFILEEJB table

20. The rest of the recovery of this table is the same as the previous. Accept the default recovery options and choose to Run Now on the summary page.

    We see in Figure 6-56 on page 185 that there were no rows changed with the UNDO. This is not a problem. Even though the ACCOUNTEJB and ACCOUNTPROFILEEJB tables are related, it does not necessarily mean that every transaction will affect both tables. We might have known that there would be nothing to undo in this case and could have answered NO to the question as posed in Figure 6-54 on page 183.

Tip: Notice in Figure 6-51 and Figure 6-56. the name of the generated undo SQL file is different. This is a result of us setting the property recoverytask.keep_sql=true in the recex.properties file. We strongly recommend setting this to true, so you can see the SQL with each recovery. If you leave the property set to its default value of false, the SQL is overwritten each time you run a recovery. This is particularly important in a DB2 Grouper scenario where you might be recovering several tables in a row and you want to keep track of what changes were made each time. Read about the recex.properties file in 3.4.1, “Customizing the recex.properties file” on page 34.

21. Click **OK** to see the next table that is offered by DB2 RE via DB2 Grouper.

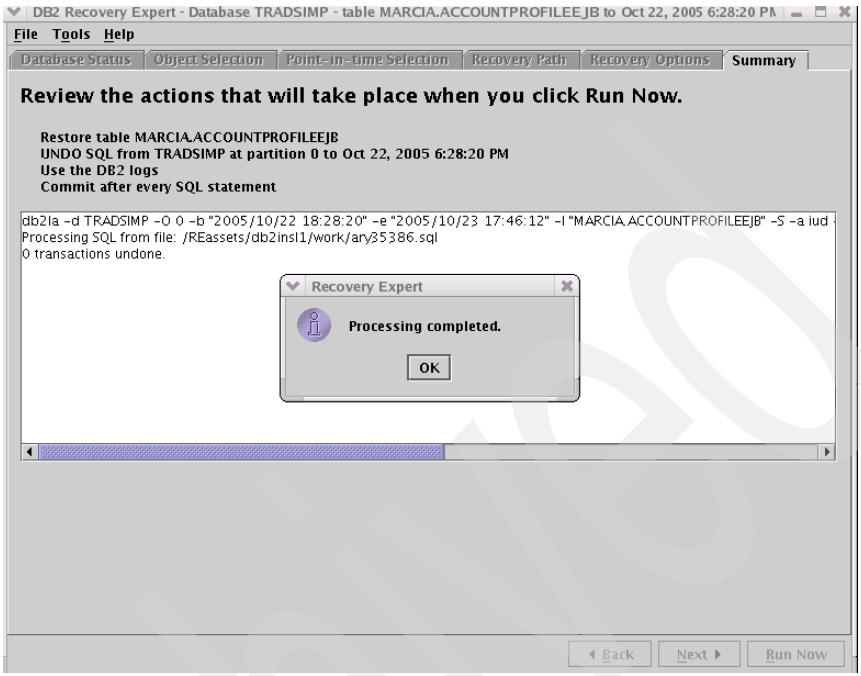


Figure 6-56 Results of recovery for ACCOUNTPROFILEEJB

22. Next we are offered a chance to recover the HOLDINGEJB table, as shown in Figure 6-57. Click **Yes** to recover this table. The HOLDINGEJB table had one transaction undone.

Figure 6-57 Choose to recover HOLDINGEJB table

23. The remaining steps are the same for this and each subsequent table. We do not show the windows here, but these were the tables recovered:

- ACOUNTEJB
- ACCOUNTPROFILEEJB
- HOLDINGEJB
- KEYGENEJB
- ORDEREJB
- QUOTEEJB

Now, the database is restored back to the point-in-time prior to the sale of the 175 shares of s:4 for user ITSOUSER1. The related tables were recovered to a consistent state, so we should be able to login to the application and see our the shares of s:4 are now back in our portfolio. This is shown in Figure 6-58.

Undoing the sale of the s:4 was not the objective of this table recovery. It only illustrates the consistent state of all the data after undoing to a point-in-time prior to the sale.

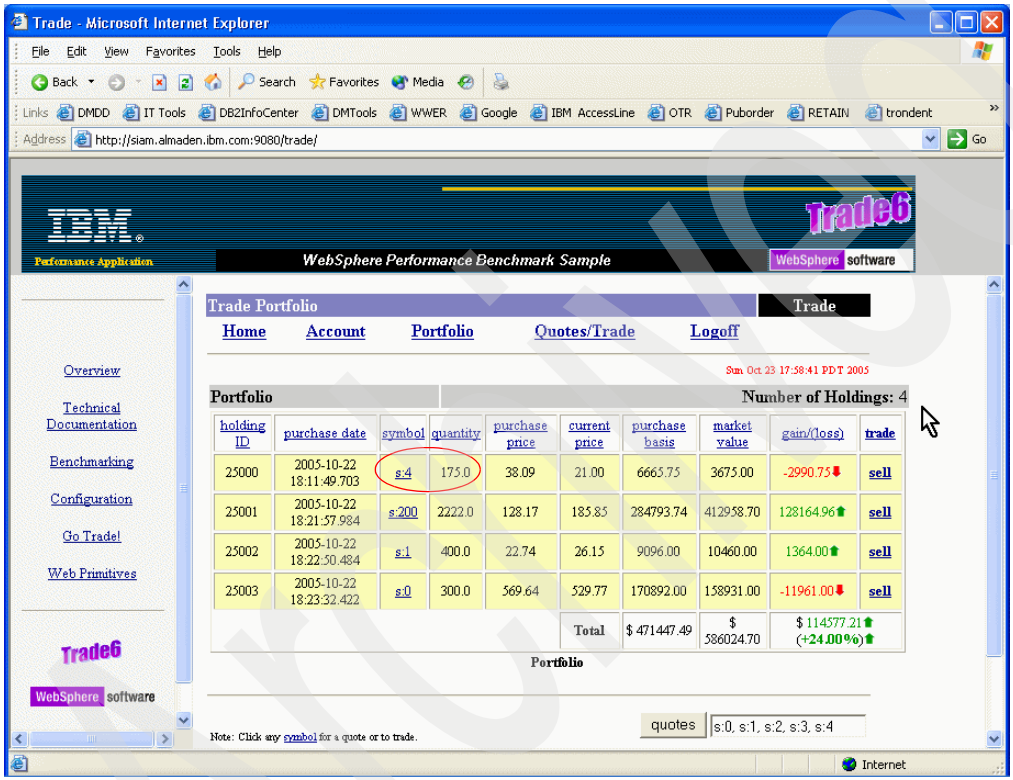


Figure 6-58 ITSOUSER1's portfolio after recovering tables to point-in-time

Note: We devised this scenario to illustrate how DB2 Grouper can recover related tables to a point-in-time to maintain a consistent data state. This particular application is a simulated financial application where you can buy and sell shares of a security. In real-life, undoing a sale such as we did would be a very bad idea for obvious legal reasons. A real application would include a reversal-type of transaction to back out mistakes such as this properly. We used this Trade application merely for demonstration purposes.

7

Recovering dropped objects

This chapter explains how to recover objects that are dropped from a table, table space, or database. It includes the following sections:

7.1 Prerequisites for recovering dropped objects

For any recovery or analysis operation using DB2 Recovery Expert (DB2 RE), you must have `LOGRETAIN` enabled in your database configuration.

To recover a dropped table or table space, you also need a full, offline table space or database backup as well as an updated Versioning Repository and Archive Log Inventory Data (ALID).

7.1.1 Updating the Versioning Repository and taking full, offline table space or database backup

DB2 RE uses the Versioning Repository to determine the definitions of dropped objects. Therefore, you must update the Versioning Repository before the time that the table was dropped. DB2 RE uses full, offline backup images to extract the data for dropped objects. Thus, for a recovery to be possible, a full offline backup should have been taken *after* the time the dropped object was created and *before* the table was dropped. Object recovery in Example 7-1 is not possible because the object definition is available in the Versioning Repository but there is no table data in a full offline backup.

Example 7-1 Example of situation where object recovery is not possible

```
1. Full offline backup taken
2. Table A created
3. VR updated
4. Table A dropped
```

Object recovery in Example 7-2 is not possible because there is no object definition available in the Versioning Repository.

Example 7-2 Second example where object recovery is not possible

```
1. Table A created
2. Full offline backup taken
3. Table A dropped
4. VR updated
```

7.1.2 Updating the ALID or metadata

You should update the ALID if you also plan to apply redo SQL after the table has been created from data in backup image and DDL in the Versioning Repository. If you do not want to apply any redo SQL against the table, then you do not have to update the ALID.

Tip: We recommend that you add Versioning Repository update just prior to every full offline backup in your operations.

Note: Dropped objects can be recovered *only* from full offline backups.

7.2 Recovering a dropped table

This section describes the process of recovering a dropped table.

7.2.1 Scenario description

In this scenario, the VPS.CAR table has been dropped from database, and we need to recover it with minimal disruption to remainder of database.

Scenario time line

Table 7-1 shows the time line for this scenario.

Table 7-1 Dropped table Scenario time line

Action	Time
T1	
T2	Versioning Repository updated
T3	Offline backup taken
T4	Delete rows in VPS.CAR
T5	Drop table VPS.CAR

Situation before the table is dropped

Example 7-3 shows the number of rows before table is dropped.

Example 7-3 VPS.CAR table at backup time

```
/home/db2insk1->db2 "select count(*) as total_rows from vps.car"
TOTAL_ROWS
-----------
     119986

  1 record(s) selected.
```

Some records are deleted, and Example 7-4 shows number of rows after the delete operations.

Example 7-4 VPS.CAR table after deletes have been done

```
/home/db2insk1->db2 "select count(*) as total_rows from vps.car"
TOTAL_ROWS
-----------
     117756

  1 record(s) selected.
```

7.2.2 Recovering a dropped table using DB2 Recovery Expert

To recover a dropped table using DB2 RE, follow these steps:

1. Start DB2 RE and connect to the VPS database, as shown in Figure 7-1.

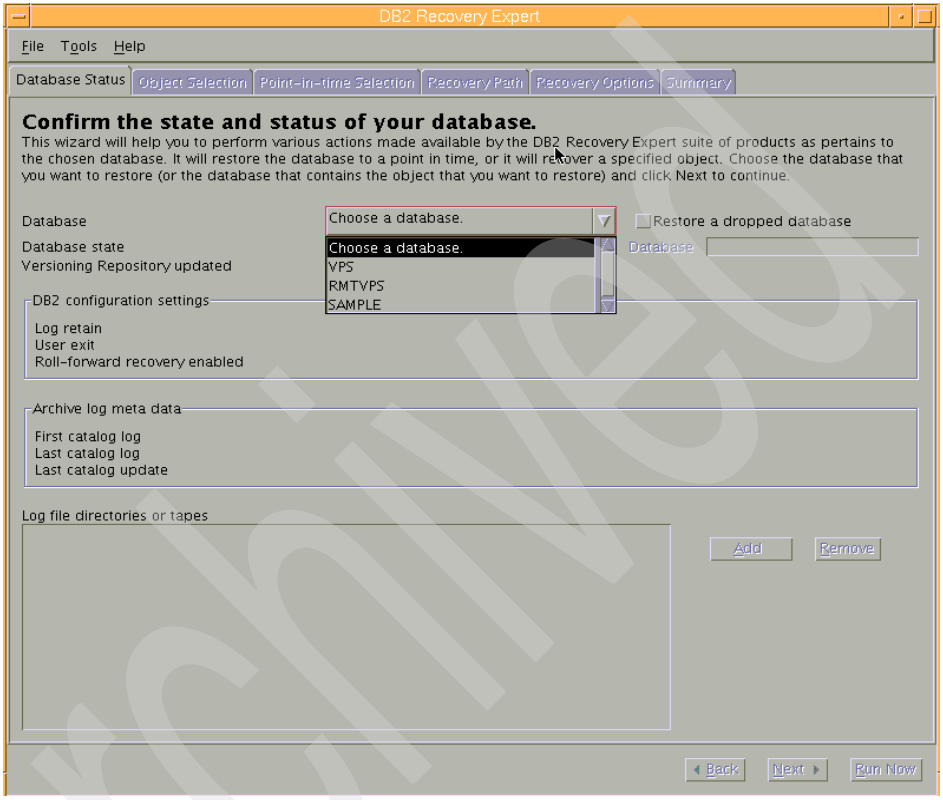


Figure 7-1 DB2 RE Main Screen

2. When connected to database, go to the Object Selection tab and select **Table**

3. Enter the table name VPS.CAR as shown in Figure 7-2.

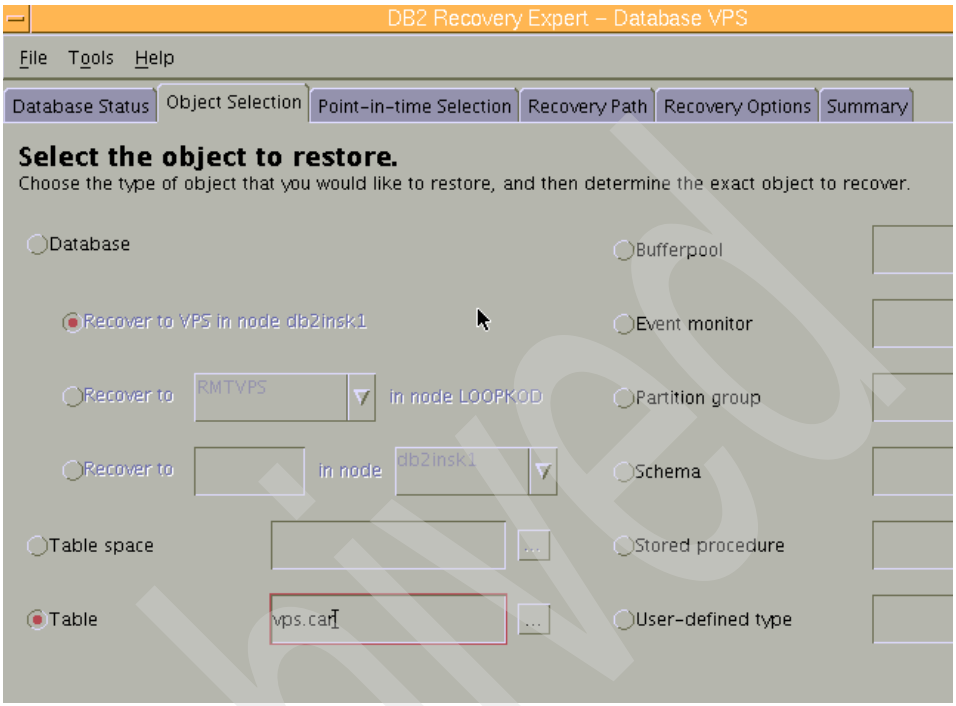


Figure 7-2 Object Selection for recovery

If you do not know the name of dropped table, you can also click the browse icon ("...") and then **Retrieve** to retrieve a list of tables (Figure 7-3 on page 193).

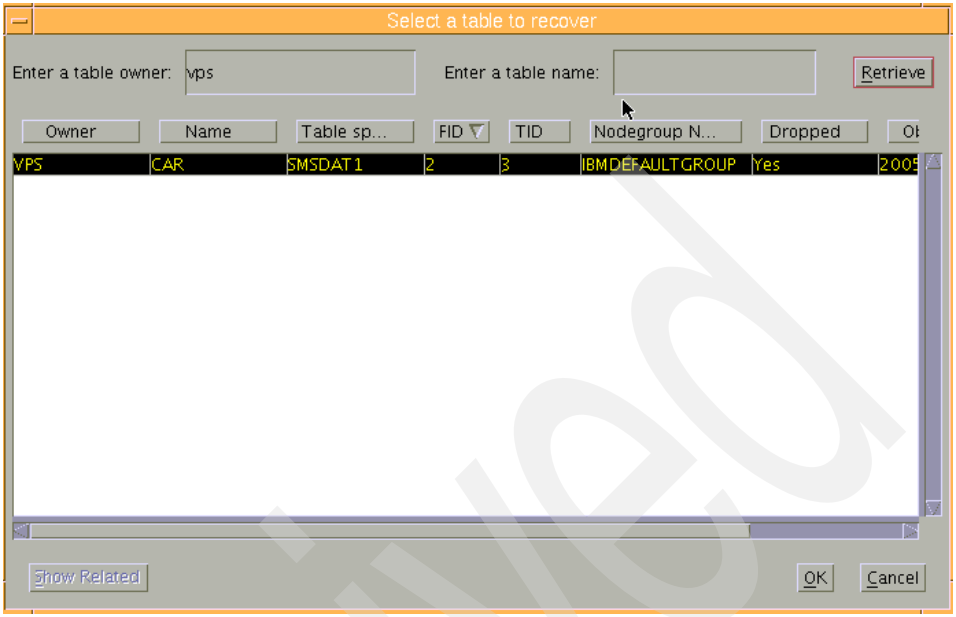


Figure 7-3 Table selection for recovery

We can see under the Dropped column that the VPS.CAR table has been dropped.

4. Click this entry and click **OK**.

This action takes us to the Point-in-time tab (Figure 7-4). Here we enter a point-in-time that corresponds to time T4 in Table 7-1 on page 189. The End of logs option is inactive. We do not want to go to the end of the logs, because we want to stop at a point-in-time just prior to the drop table transaction.

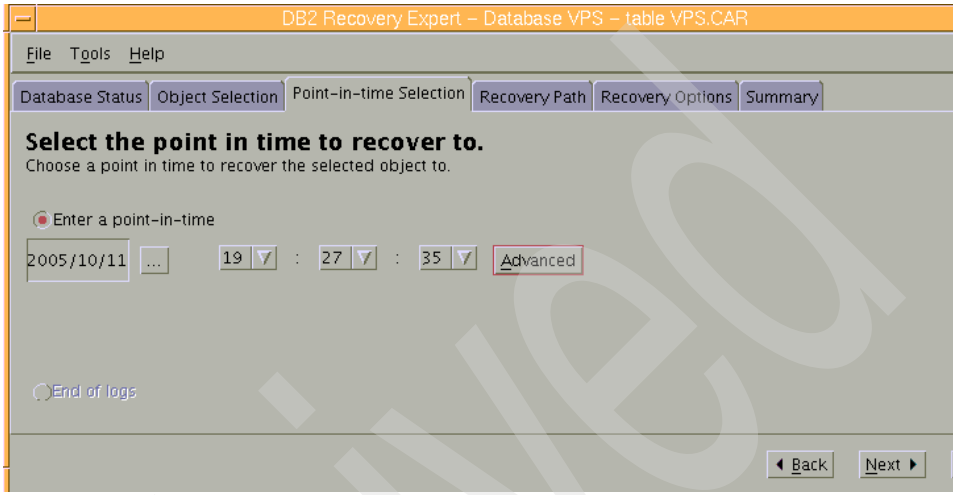


Figure 7-4 Point-in-time selection for recovery

5. Click **Next** to move to the Recovery Path tab (Figure 7-5). This window shows us all the recovery paths that are available to us. In this case, there is only one option available. The number in parentheses, 485, is a relative cost. If there were multiple options available, DB2 RE would have selected the fastest option (the option with lowest relative cost value).

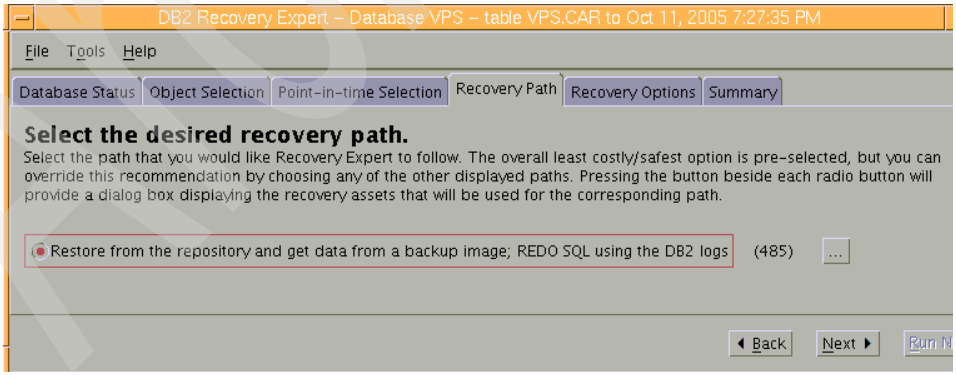


Figure 7-5 Recovery Path window

6. Optionally, you can click the browse icon ("...") to see more details about Recovery Assets (Figure 7-6 on page 195).

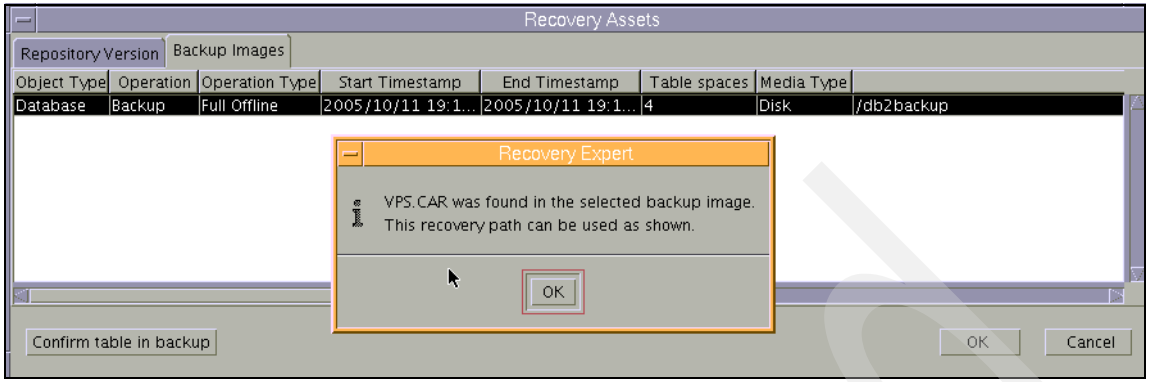


Figure 7-6 Recovery Assets detail

7. Click **Confirm table in backup** to verify that the dropped table is available in the selected backup image. Click **OK** to go back to Figure 7-5 on page 194. Click **Next** to go to the Recovery Options window (Figure 7-7).

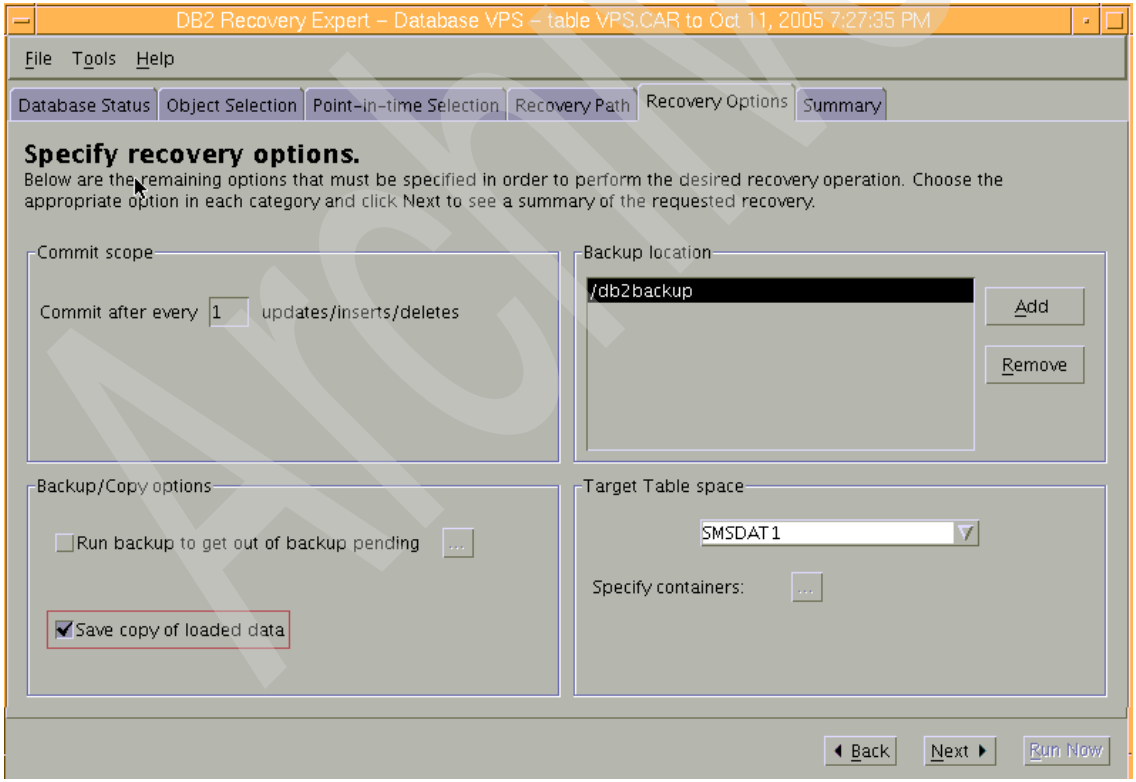


Figure 7-7 Recovery Options

In this window, you have the following options:

- Saving a copy of the loaded data

  When the data is extracted from the backup image, DB2 RE runs the DB2 **LOAD** command to load data into the table. (For more information about this command, see *Administration Guide: Implementation V8*, SC09-4820.) This option refers to the COPY YES option of the **LOAD** command. When selected, it saves a copy of the loaded data in the directory that is specified in the backup location.

- Running a backup to get out of backup pending state
- If you do not select **Save copy of loaded data**, then the **LOAD** command drops to the default of COPY NO. When this option is used, the table space in which the table resides is left in backup pending state. This option runs a table space backup immediately after the load to bring table space to normal state.

8. Click **Next** and it takes us to Figure 7-8.

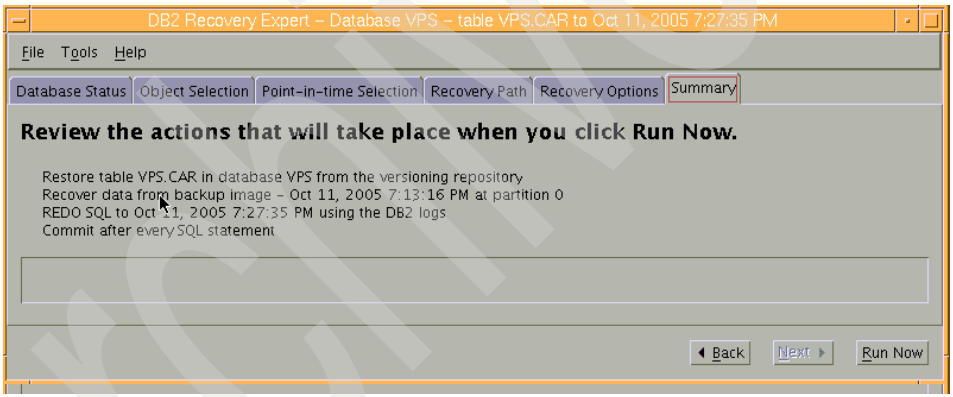


Figure 7-8 Summary window

In this window, DB2 RE outlines all the action it takes in sequence to resurrect the dropped table.

9. Click **Run Now** to execute these actions (Figure 7-9 on page 197).

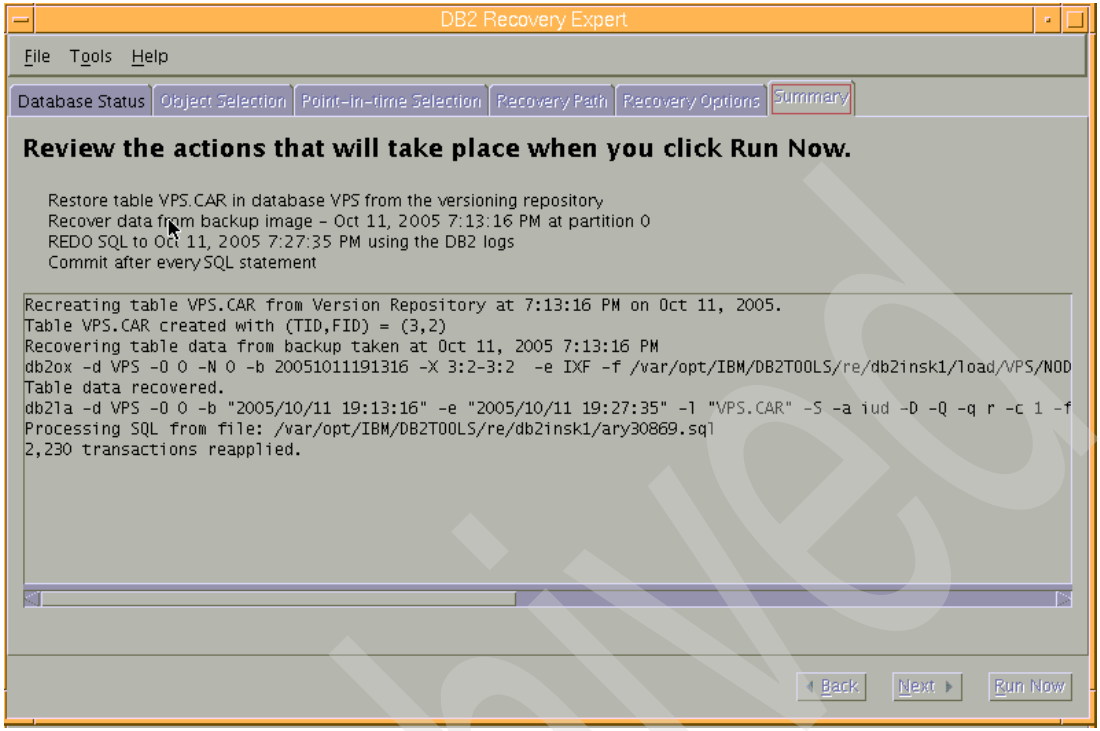


Figure 7-9 Results window

We can see that the table is created from the Versioning Repository. Then, **db2ox** is run to extract data from the backup image. When the data is loaded into a table, we need to redo the transaction that was executed after the backup but before the specified point-in-time, which is just prior to the time that the table was dropped. In this case, **db2la** is invoked to create the redo SQL. We see that in the end 2,230 transactions were applied. We can do some cross checking here by checking the difference in counts in Example 7-3 on page 190 and Example 7-4 on page 190. The difference is 2230 rows, which matches with number of redo transactions that were created and applied by DB2 RE.

7.2.3 LOB and long field considerations

Dropped tables with LOB or long field data in a different table space can also be recovered as long as both regular data and LOB and long field are in the same backup image. So, you can recover from full, offline backups and table space backups where both data and LOB table space are in the same backup image. However, you cannot recover a table when regular data is in one table space backup image and LOB and long field data is in another table space backup.

As described in “Log analysis against table with LOB data in different table space” on page 118, log analysis cannot produce redo SQL for tables with LOB data in separate table space. This implies that for such tables you can only recover the table from the backup image. You cannot create, apply, and redo transactions to bring them up to a point-in-time after the backup was taken.

7.2.4 Dropped table recovery options without DB2 Recovery Expert

Using standard DB2 functions, you can recover a dropped table if your table space has `DROPPED TABLE RECOVERY` enabled. (For more information, see *Administration Guide: Implementation V8*, SC09-4820.) However, you have to restore the entire table space, not a single table. This restriction can be an issue when you have multiple tables per table space. Using DB2 RE, with or without Dropped table recovery ON, you can still recover a single table until the last committed transaction.

Tip: We recommend that you enable `DROPPED TABLE RECOVERY` for your table spaces, because it causes an entry to be written to the recovery history file when a table is dropped. Otherwise, if a table is dropped accidentally, it is much more difficult to determine exactly when it was dropped.

To enable `DROPPED TABLE RECOVERY` for a table space, you can use the command:

```
db2 alter tablespace <table space> dropped table recovery on
```

When `DROPPED TABLE RECOVERY` is enabled, this command gives you the time when an object was dropped:

```
db2 list history dropped table all for <database name>
```

7.3 Recovering a dropped table space

This section describes the steps to recover a dropped table space.

7.3.1 Scenario description

A System Managed Space (SMS) table space SMSDAT1 has been dropped in our database. We need to recover this table space completely. In other words, we need to recover it to the last committed SQL against the table space prior to the drop table space operation. SQL are run against tables, so this would be the last committed transactions for one of the tables in the table space.

Scenario time line

Table 7-2 shows the time line for this scenario.

Table 7-2 Scenario time line

Action	Time
T1	Versioning Repository updated
T2	Full Offline Backup taken
T3	Rows deleted from table VPS.CAR
T4	Table space SMSDAT1 is dropped
T5	Recover SMSDAT1 table space to a time after T3 and before T4.

Situation before table space is dropped

Example 7-5 shows the situation at time between T3 and T4.

Example 7-5 Number of rows before the delete SQL

```
/home/db2insk1->db2 "select count(*) as total_rows from vps.car"
TOTAL_ROWS
-----------
     119986
  1 record(s) selected.
```

Example 7-6 shows the situation just prior to table space drop.

Example 7-6 Number of rows in VPS.CAR table just before table space is dropped

```
/home/db2insk1->db2 "select count(*) as total_rows from vps.car"
TOTAL_ROWS
-----------
     117756

  1 record(s) selected.
```

7.3.2 Recovering a dropped table space with DB2 Recovery Expert

To recover a dropped table space, follow these steps:

1. Start up DB2 RE GUI and connect to VPS database, as shown in Figure 7-10.

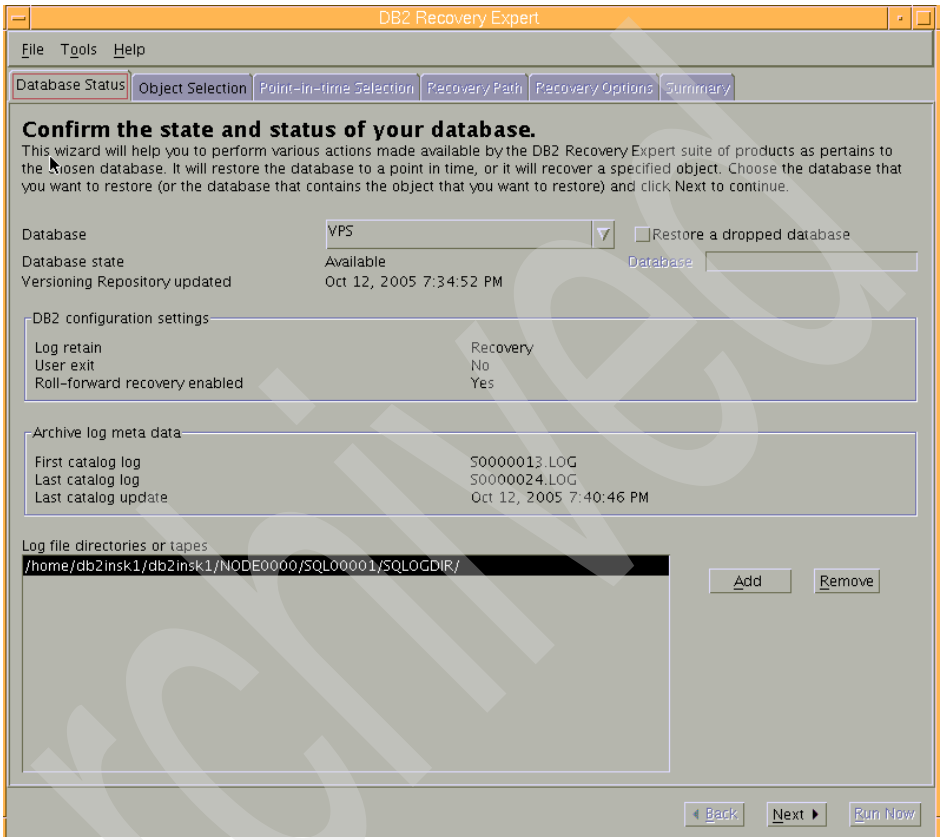


Figure 7-10 DB2 RE main screen

2. Click **Next** to go to the Object Selection tab, as shown in Figure 7-11 on page 201.

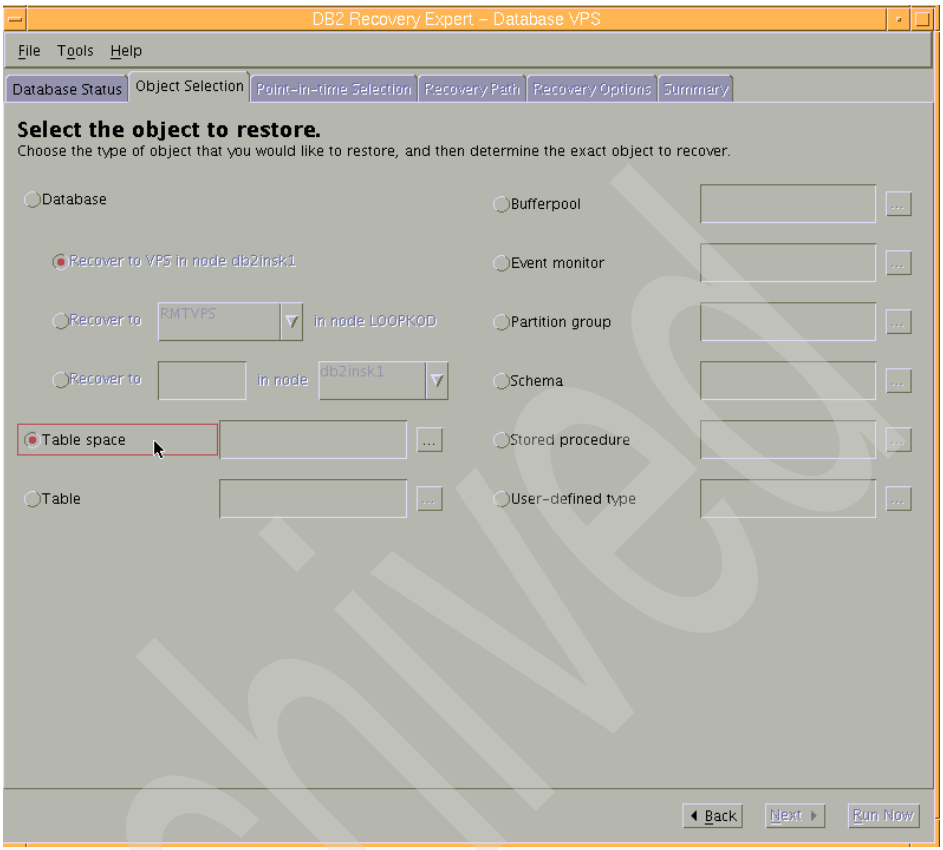


Figure 7-11 DB2 RE object selection

3. Select **Table space**.

4. Click the browse icon ("...") next to the Table space option to get to Figure 7-12.

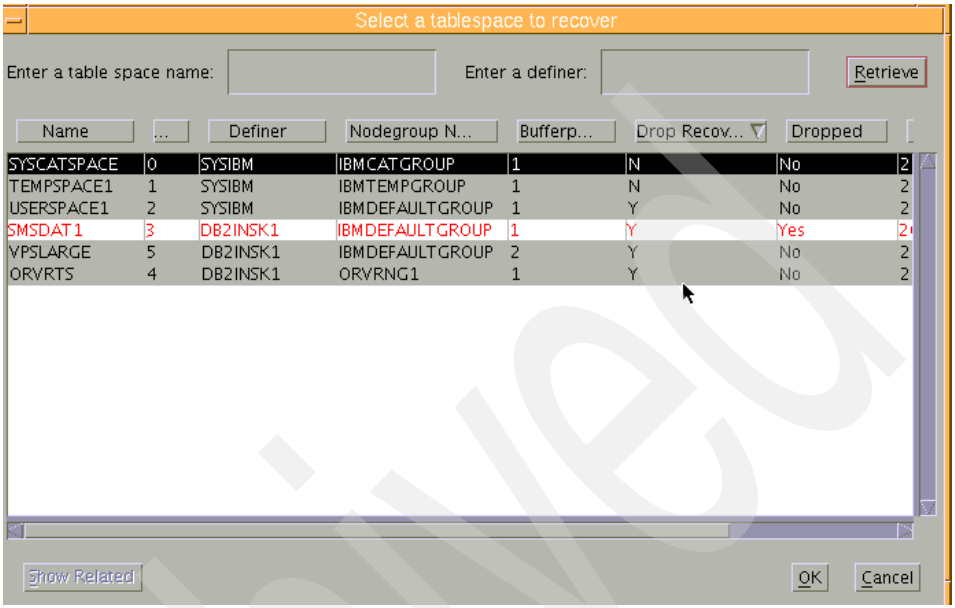


Figure 7-12 DB2 RE table space selection

We can see that SMSDAT1 is listed and the Dropped column shows Yes for SMSDAT1 table space.

5. Select the dropped table space, click **OK**.

6. Click **Next** to go to Point-in-time Selection tab (Figure 7-13).

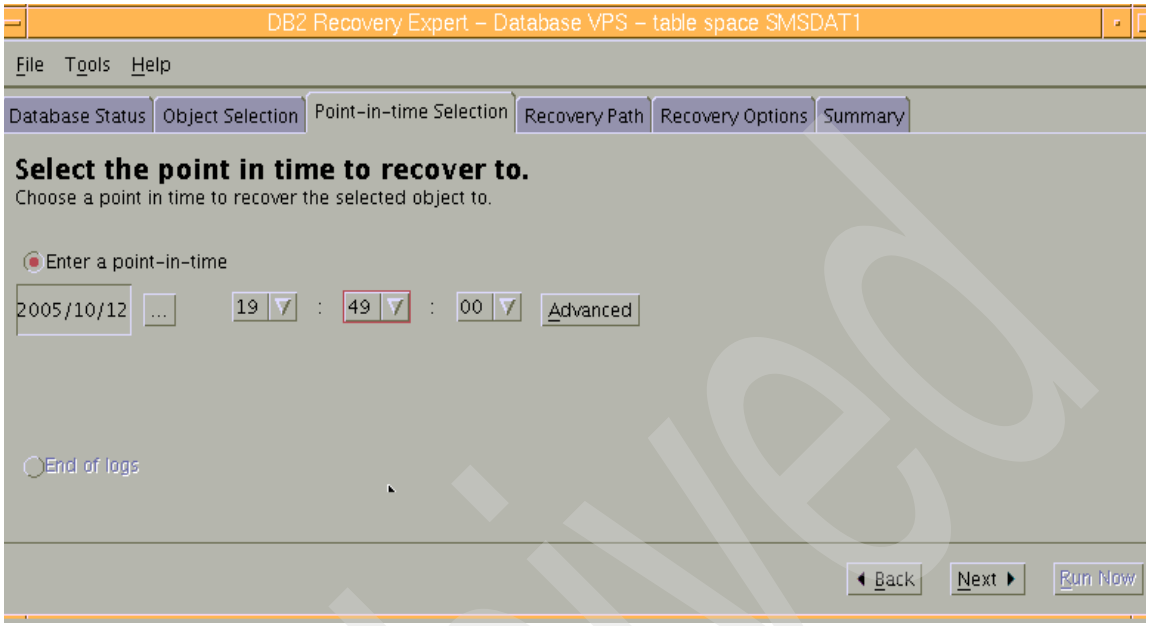


Figure 7-13 Point-in-time Selection

We enter a point-in-time just prior to dropping the table space. If you are not sure what time stamp to use, check the history file to find the time when the table space was dropped, as follows:

```
db2 list history all for <database name>
```

Example 7-7 shows a dropped table space entry in history file.

Example 7-7 List history output

```
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log  Backup ID
-- --- ------------------ ---- --- ------------ ------------ --------------
 O  P  20051012194920
-------------------------------------------------------------------------------
 Contains 1 tablespace(s):
 00001 SMSDAT1
-------------------------------------------------------------------------------
   Comment: DROP TABLESPACE
Start Time: 20051012194920
  End Time: 20051012194920
    Status: A
-------------------------------------------------------------------------------
 EID: 4 Location:
DDL: drop tablespace smsdat1
-------------------------------------------------------------------------------
```

7. Click **Next** to go to Recovery Path window (Figure 7-14).

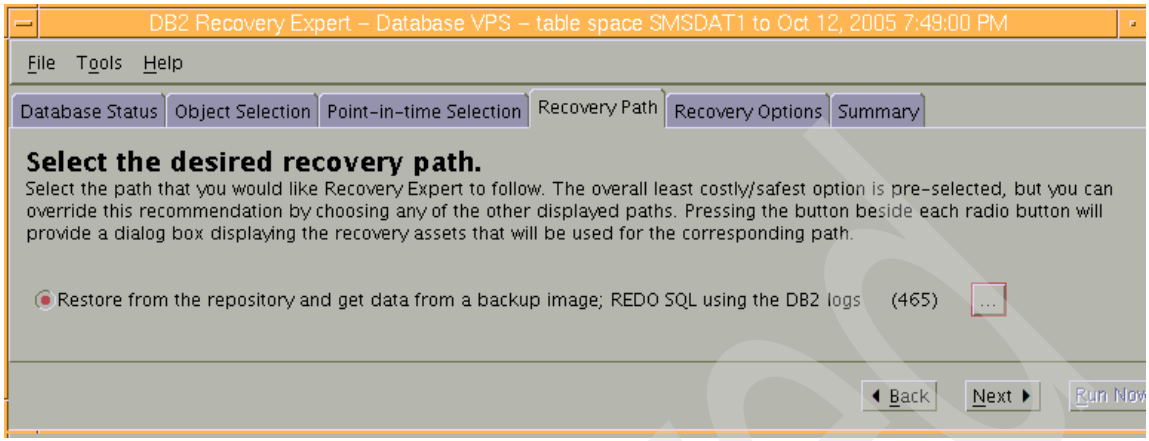


Figure 7-14 Table space recovery paths

8. There is only one option available here. Click **Next** to go to Recovery Options window (Figure 7-15).

Figure 7-15 Table space recovery options

9. Take the defaults and click **Next** to go to the Summary window (Figure 7-16).

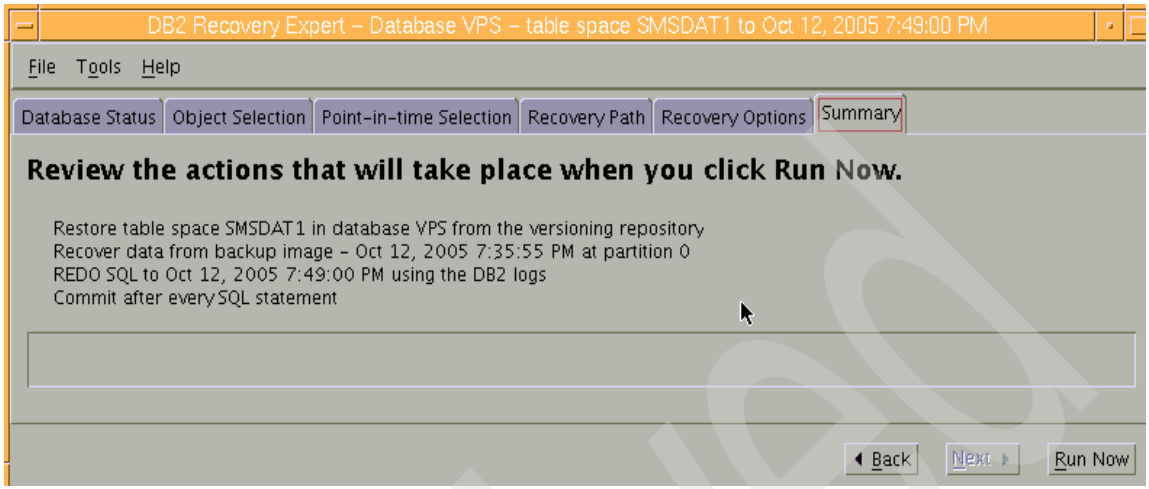


Figure 7-16 Table space recovery window

This window provides a summary of that actions that DB2 RE takes to do the table space recovery:

- Recreates table space using DDL stored in Versioning Repository.
- Restores the `SMSDAT1` table space into newly created table space using **`db2ox`**.
- Runs log analysis to create redo SQL for transactions since the backup image
- Executes this SQL, committing after very SQL statement.

10. Click **Run Now** to execute these actions.

Figure 7-17 provides the detailed actions taken with commands.

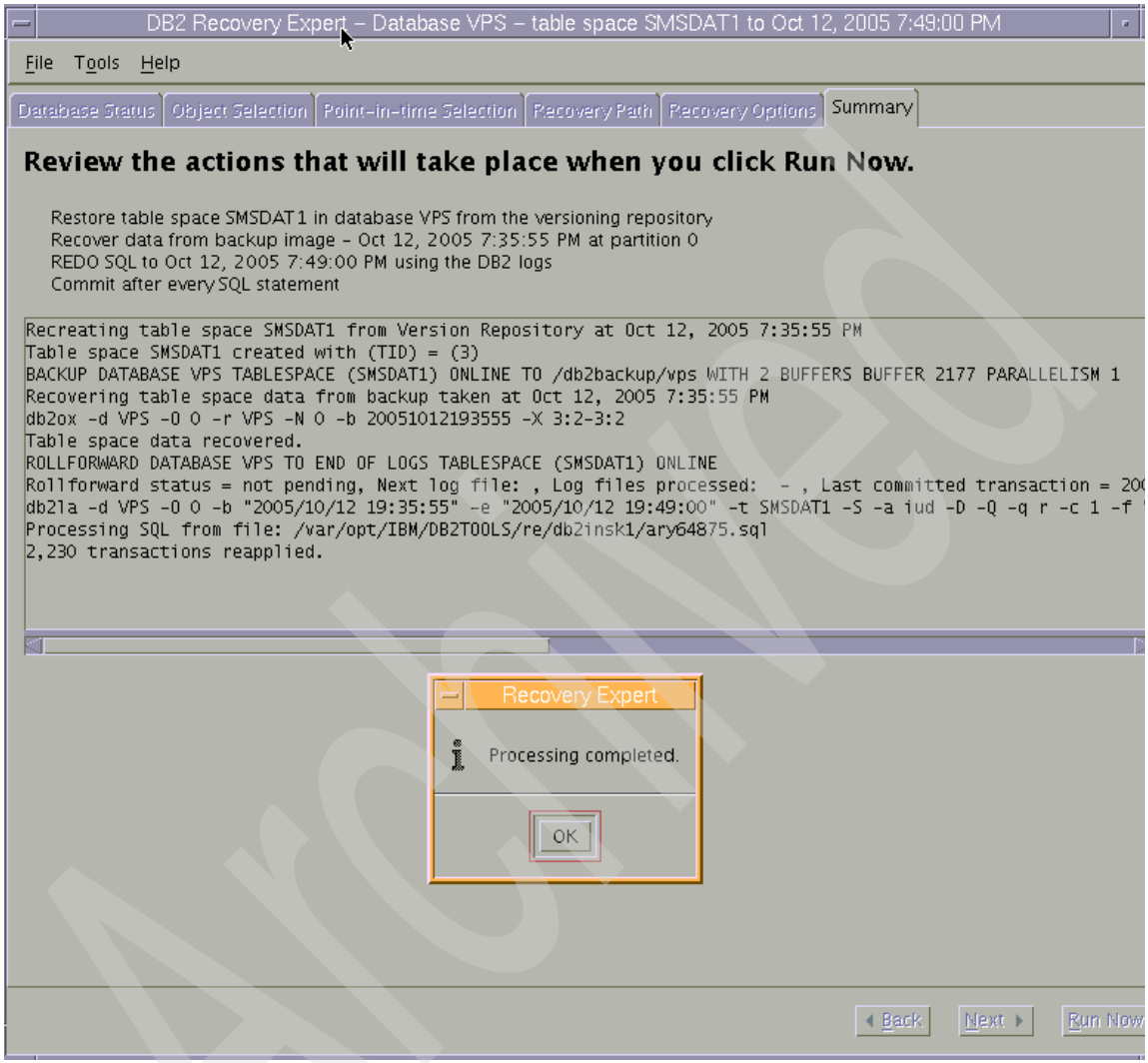


Figure 7-17 Table space recovery results

The dropped SMSDAT1 table space has now been recovered to a point-in-time between T3 and T4. We can do some cross checking and see that the last message in Figure 7-17 says that 2,230 transactions have been applied. This number was the number of rows that were deleted at time T3. We can determine this by taking the difference between the number rows shown in Example 7-5 on page 199 and Example 7-6 on page 199.

7.3.3 Recovering dropped table space without DB2 Recovery Expert

Recovering a dropped table space using native DB2 UDB methods can be very disruptive and costly.

DB2 UDB does not allow restore of a table space that has already been dropped. The table space has to exist in the database before we can do a table space level restore into it. Therefore, a table space restore from a table space backup is not possible even if you have table space data available in a backup image. This means that the only option you have to recover table space is to restore the entire database and rollforward to a point-in-time before the table space was dropped. This can be disruptive as the entire database would become unavailable for the duration of recovery. Restoring a very large database can take several hours. In DPF environments, you would have to restore database on all nodes.

Moreover, with database restore and rollforward, you must stop rollforward before the table space was dropped. There could have been many transactions done against other tables and table spaces since the time of drop table space. Once you stop at this point-in-time, you loose all transactions that were executed against the database after this point-in-time.

7.3.4 Restrictions with table space recovery

At this time, DB2 RE does not support recovering a dropped table space that has tables with indexes , LOB, or long data in a separate table space.

7.4 Recovering a dropped database

When a database has been dropped, you can restore it from the DB2 command line, the Control Center, or the DB2 RE GUI. This section explains how to recover a dropped database using these methods.

7.4.1 Prerequisites for recovering a dropped database

To recover a dropped database, the following conditions must be met:

- A full, online or offline database backup must be available.

  If restoring from an online backup, you should also have the minimum set of logs associated with the online backup available.

- `LOGRETAIN` must be set to `Recovery` if you wish to rollforward through logs.

You can find the logs that are required for a given online backup by issuing the following command:

```
db2 list history backup all for <database name>
```

Example 7-8 shows the output of an online backup using this command.

Example 7-8 List history output for an online backup

```
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log  Backup ID
 -- --- ------------------ ---- --- ------------ ------------ --------------
  B  D  20051018201659001   N    D  S0000193.LOG S0000195.LOG
 ----------------------------------------------------------------------------
  Contains 2 tablespace(s):

  00001 USERSPACE1
  00002 SMSDAT1
 ----------------------------------------------------------------------------
    Comment: DB2 BACKUP VPS ONLINE
 Start Time: 20051018201659
   End Time: 20051018201720
     Status: A
 ----------------------------------------------------------------------------
  EID: 11 Location: /db2backup
```

In this example, we can see the Earliest Log and the Current Log that are associated with this online backup. Log files from S0000193.LOG to S0000195.LOG must be available to restore from this backup.

7.4.2 Recovering a database using DB2 commands

You can use DB2 **RESTORE** command to recover a dropped database from either an offline or online backup. There are a lot of options available with the **RESTORE** command. For more information about this command, see *Data Recovery and High Availability Guide and Reference V8*, SC09-4831. The following is an example of a typical **RESTORE** command:

```
db2 restore database <database name> from <backup path> taken at <date/time of
backup> without prompting
```

After the restore is finished, the database is left in rollforward pending state. If you restore the database from an offline backup, then you can decide to stop here or you can rollforward. If you restore from an online backup, then you need to rollforward to a minimum point-in-time. For this rollforward, the associated log files must be available, as stated in 7.4.1, “Prerequisites for recovering a dropped database” on page 207.

You can rollforward database with the following command:

```
db2 rollfoward database <database name> to end of logs
```

Note: Logs in the active log path are also dropped when a database is dropped. Therefore, you must have logs archived in an archive location if you wish to do a rollforward. You need to copy the logs to active log path of the restored database to be able to do a rollforward. If you do not wish to copy the logs into database log path, you can place them anywhere on the system and use `overflow logpath` clause in the **`rollforward`** command to point to them.

After you perform the rollforward, you can bring the database out of rollforward pending state by issuing the following command:

```
db2 rollfoward database <database name> stop
```

At this point, the database should be in a normal and connectable state.

7.4.3 Recovering a database using DB2 Recovery Expert

DB2 RE can work with both DB2 backups and backups taken with Fast Backup. For information about restoring a database using Fast Backup, see 9.5, “Scenario B: Recovering a database on the multiple network storage volumes” on page 286.

To recover a database using the DB2 RE GUI, follow these steps:

1. Start up DB2 RE GUI and connect to VPS database.
2. Choose **Restore a dropped database**, as shown in Figure 7-18 on page 210.

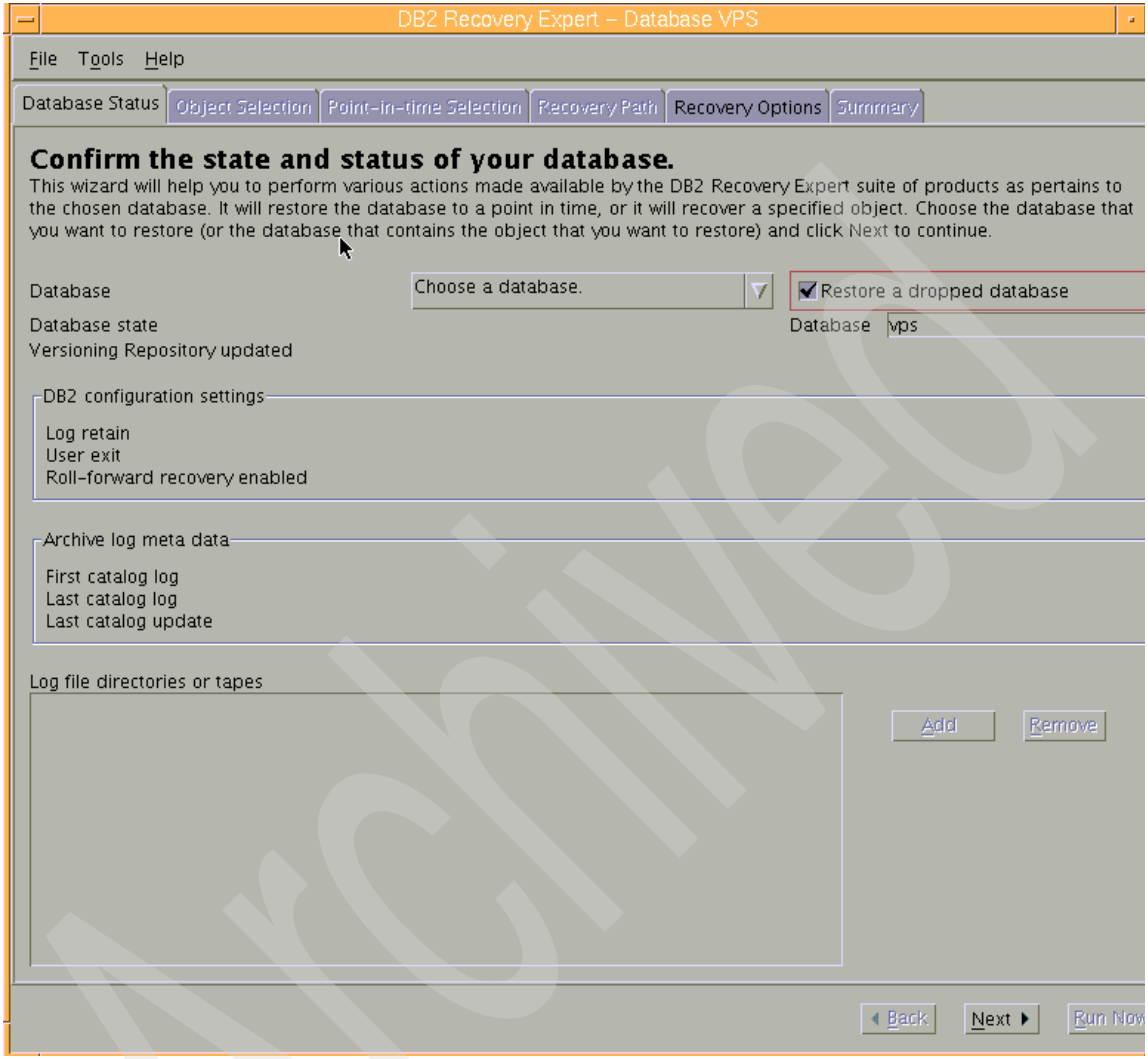


Figure 7-18 Recover dropped database: Main window

3. After you enter the database name, the Recovery Options tab becomes available. Click **Next** to go to this tab (Figure 7-19 on page 211).

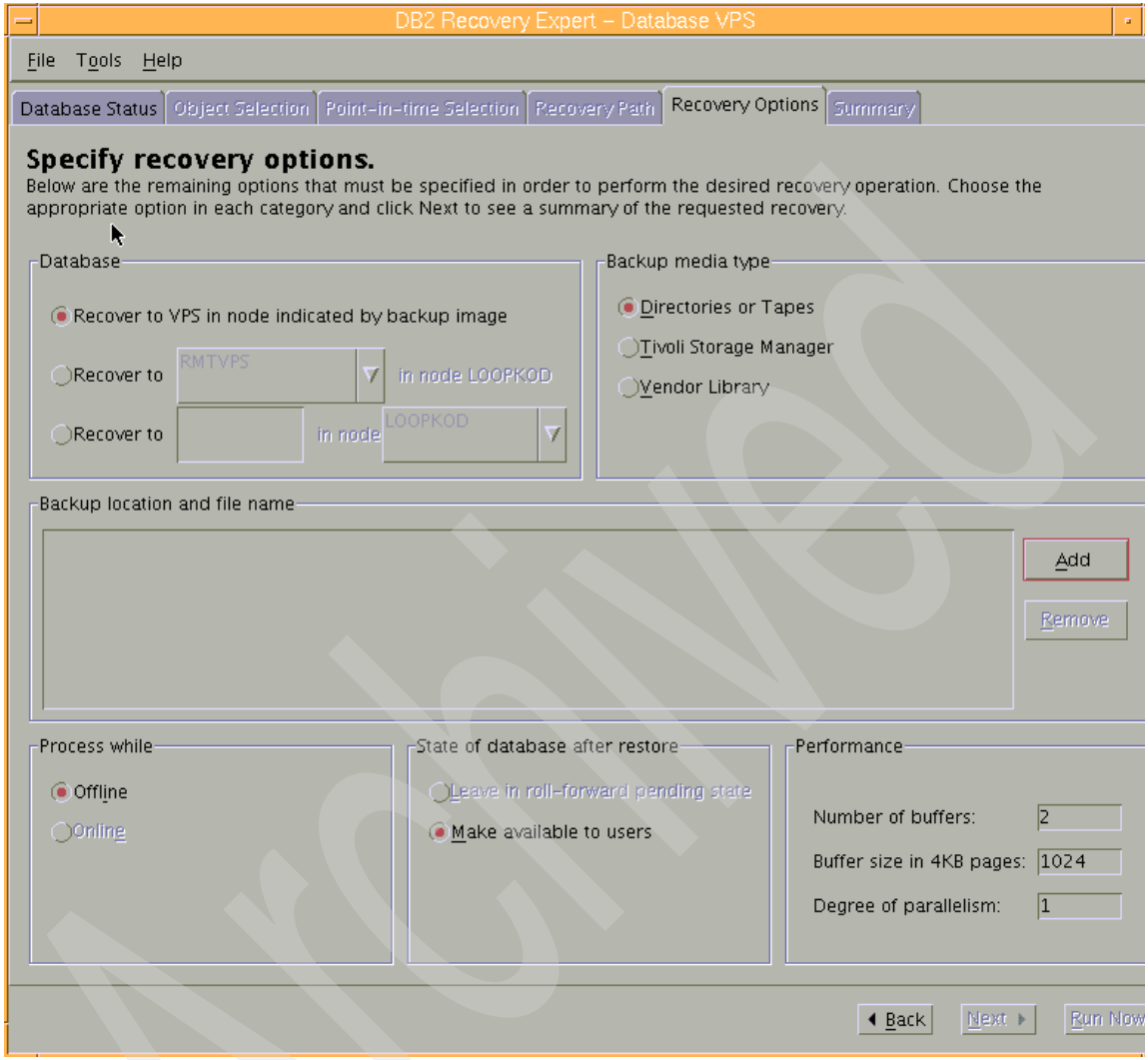


Figure 7-19 Database recovery options

In this tab, you have the following options:

– Process while

  The Online option is inactive because the database does not yet exist. The Online option does not apply here.

– State of database after restore

  DB2 RE does not allow you to select Leave roll-forward pending state. Make available to users is selected automatically, which indicates that

DB2 RE will add the `WITHOUT ROLLING FORWARD` option to the DB2 **`RESTORE`** command. Thus, DB2 RE does not allow restoring from a full online database backup, because you must rollforward through a minimum set of logs after restoring from an online backup. You can restore from full offline database backups, but you would not be able to perform a rollforward operation.

4. Click **Add** to specify backup location (Figure 7-20).

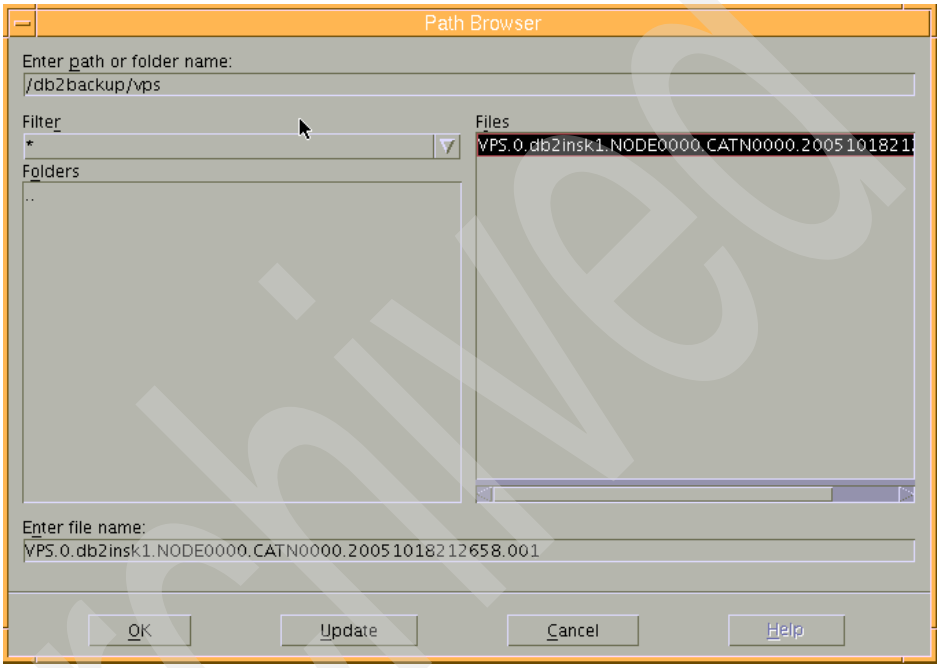


Figure 7-20 Database recovery path browser

5. Select the backup image and click **OK**. This brings us to Figure 7-21 on page 213.

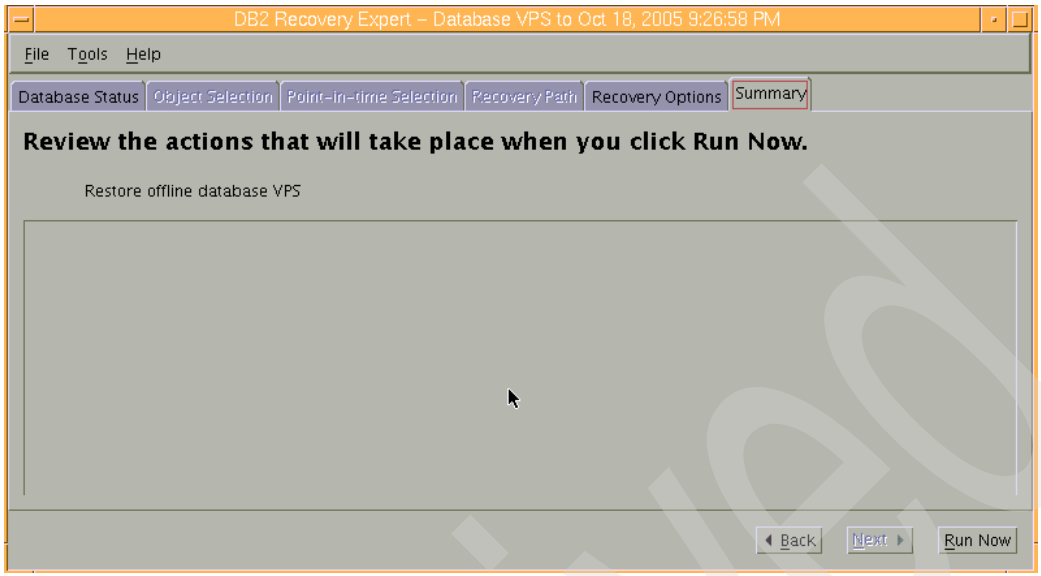


Figure 7-21 Database recovery summary

6. Click **Run Now** to execute the **RESTORE** command. Figure 7-22 shows the results window.

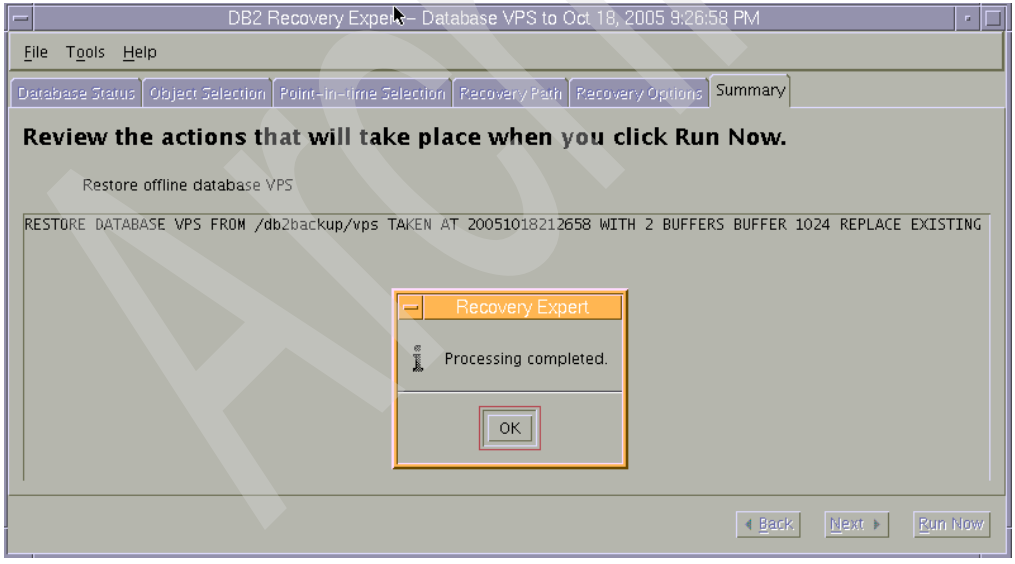


Figure 7-22 Database recovery results

The dropped database is restored and available for users.

8

Cloning a database, table space, or table

This chapter describes how to clone a database and how to clone tables or table spaces from within a database or to another database. It looks at the Object Translation command, which DB2 Recovery Expert uses in the second cloning process for tables and table spaces. It then introduces a new sample tool to simplify the table cloning process when used outside the GUI.

This chapter includes the following sections:

8.1 Cloning a database

There are certain situations during an application's life when you might want to take a copy of a database for analysis purposes without disturbing the original database. You can *clone* a database using various DB2 tools, such as **db2look**. However, DB2 Recovery Expert (DB2 RE) makes it easier to transfer the most up-to-date version of the data without impacting the availability of the data.

DB2 RE calls the native DB2 restore command to run the database cloning process, so you may find this easier using DB2 RE rather than native DB2 commands or the DB2 Control Center. However for cloning at a more granular level, tables or table spaces, DB2 RE enables you to extract tables to a previous point-in-time.

Note: DB2 RE uses standard DB2 processes to clone a database by calling DB2 to run a database restore or redirected restore. DB2 RE does not use object translation for cloning a database.

This cloning process is best described using a scenario. This section discusses in detail a scenario that clones a database.

8.1.1 Details of this scenario

For this scenario, we have a performance database named PETEST1 where we are running a series of performance tests prior to moving our application into production. We have discovered some *hot-spots* in the application where performance is unacceptable. However, we do not want to interrupt performance testing to tune the database and possibly make some physical changes to the database. The answer is to clone the complete PETEST1 database to another identical copy called PETEST5. DB2 RE can do this in a few simple steps.

8.1.2 Prerequisites for the scenario

To clone a database using DB2 RE, the following prerequisites must be met:

- A full offline backup of the database to be cloned is required.
- The target database must be on the same machine but can be a different DB2 instance.
- `LOGRETAIN` must be set to `RECOVERY` mode.

We ran an update and a count on a specific table in the performance database PETEST1 to make sure that the latest data was being restored to the cloned database. Initially, the row count for the RACER table is 40,100 rows. After

inserting 500 rows the total has increased to 40,600, as shown in Example 8-1. This is the count against which we can check in the cloned database to ensure that it has rolled forward beyond the backup copy.

Example 8-1 Generate activity on the source database

```
C:\Program Files\IBM\SQLLIB\BIN>db2 select count(*) from racer

1
-----------
      40100

  1 record(s) selected.


C:\Program Files\IBM\SQLLIB\BIN>db2 insert into racer select * from race fetch
first 500 rows only
DB20000I  The SQL command completed successfully.

C:\Program Files\IBM\SQLLIB\BIN>db2 select count(*) from racer

1
-----------
      40600

  1 record(s) selected.
```

8.1.3 Creating a clone of a database

To create a clone of the PETEST1 database, follow these steps:

1. Start the DB2 RE GUI, from Windows XP in this scenario, by selecting **Start** → **All Programs** → **DB2 Recovery Expert 2.1** → **Recovery Exper**t.
2. Connect to the source database (Figure 8-1).

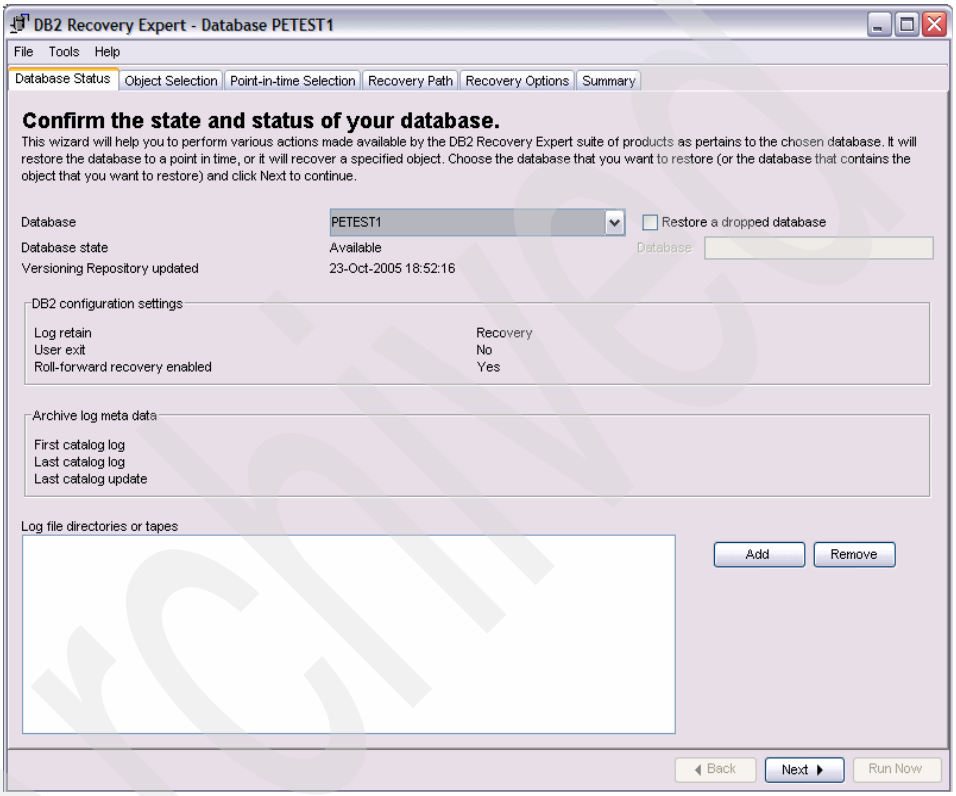


Figure 8-1 Connect to source database PETEST1

We can see from this first panel when the Versioning Repository was last updated. We recommend that the Versioning Repository is updated before any recovery process is attempted, especially if it is possible any DDL has been executed. See Figure 8-2 on page 219 for Versioning Repository update from the tool bar.

Figure 8-2 Update Versioning Repository

Also on the first panel we can see the Last catalog update. This refers to the DB2 RE Archive Log Inventory Data (ALID). Again we recommend updating the ALID regularly because it will save time when an actual recovery is run. If you do not run the ALID update first, then it is run in update mode during a recovery scenario.

3. From the tool bar, click **Tools** → **Log Analysis** (Figure 8-3).

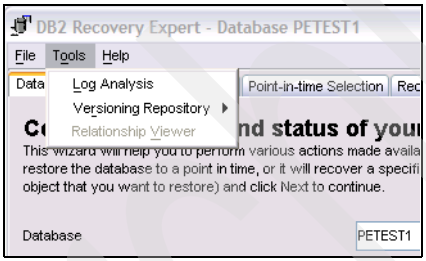


Figure 8-3 Invoke Log Analysis from the Tool Bar

This takes you to the Log Analysis Tool window, as shown in Figure 8-4.

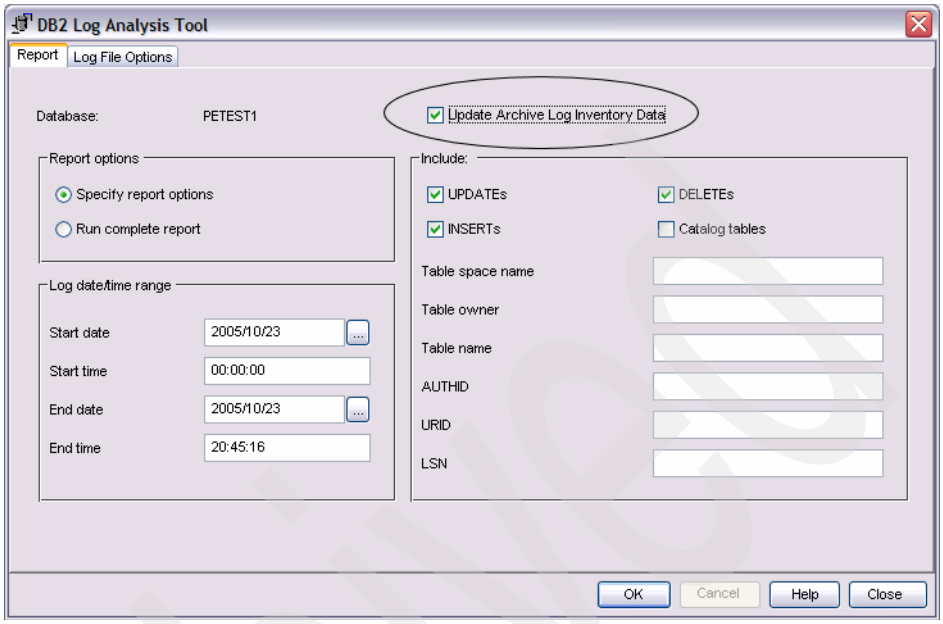


Figure 8-4 Update the ALID

4. Select **Update Archive Log Inventory Data** and click **OK** to run the update.

   The ALID is updated irrespective of the other parameters that you enter in this window. These parameter are only to customize the reports. The ALID is an incremental update and is brought up-to-date from when it was last run to include the last committed unit of work in the database.

The Log Analysis General Report window opens (Figure 8-5). You can view various reports by selecting the item lists in the left pane. However, the report was generated in this case as a by-product of updating the ALID. An alternative could have been to update the ALID from the command line.

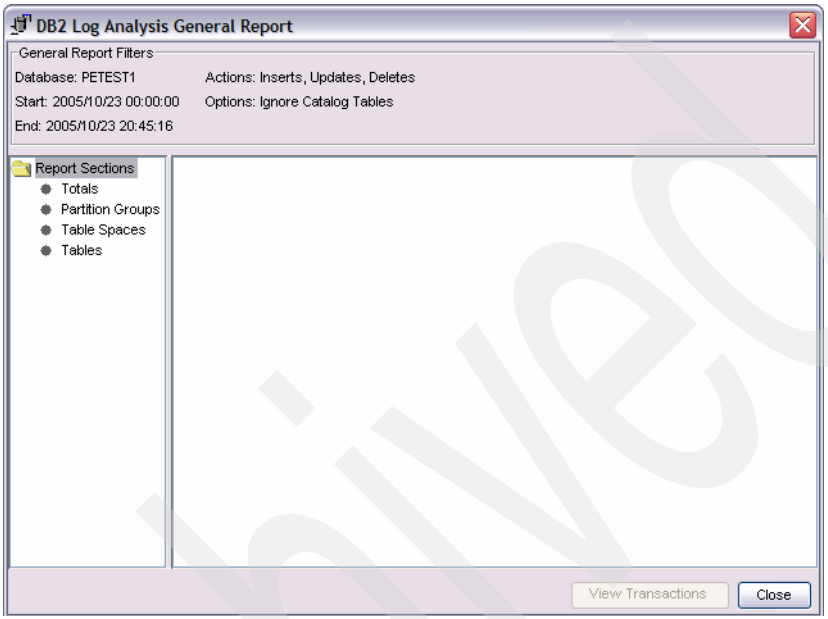


Figure 8-5 Log Analysis General Report screen

5. Click **Close** to close the General Report window and click **Close** to close the Log Analysis window. This takes you back to the DB2 RE main window.
6. Click **Next** to navigate to the Object Selection window.
7. In this window, click **Recover to** and enter the name of the target database, which in this scenario is PETEST5 (Figure 8-6 on page 222).

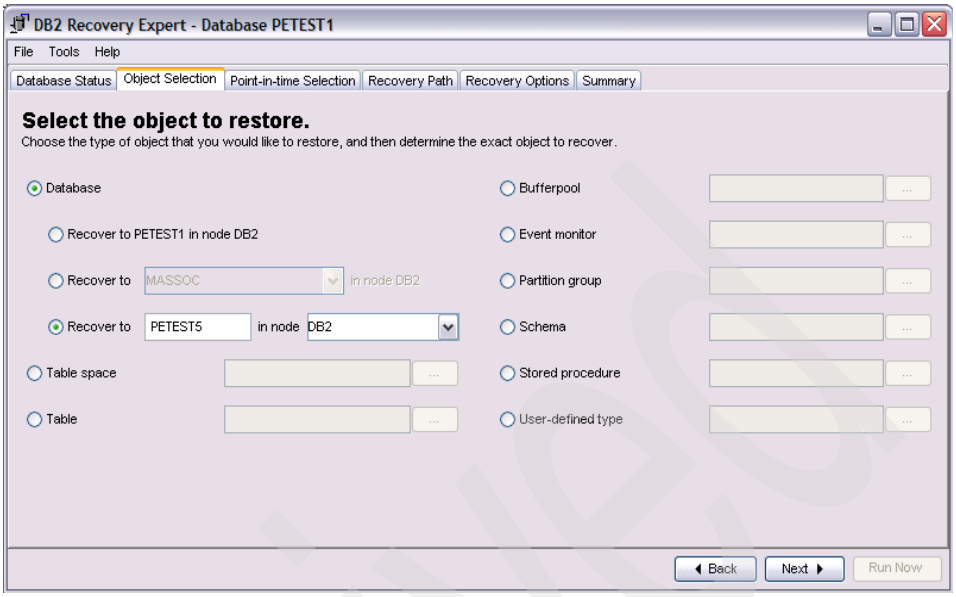


Figure 8-6 Enter the target database name

8. Click **Next** to go to the Point-in-time-Selection panel (Figure 8-7).

Figure 8-7 First recover point-in-time panel

9. Select **Enter a point-in-time** to choose how you want to rollforward, and then click **Advanced** (Figure 8-8 on page 223).

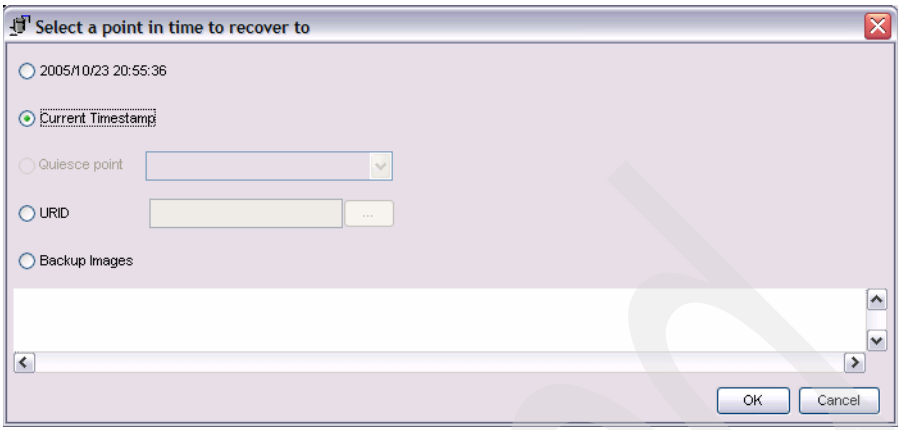


Figure 8-8 Select a recovery point-in-time

10. Select **Current timestamp** and click **OK** to go back to the Point-in-time Selection window (Figure 8-7 on page 222).

11. Click **Next** to advance to the Recovery Path window.

    At this point, you might get a window warning you that the ALID is not up-to-date (Figure 8-9). You can click **OK** to bypass this message, because it is just a warning that recovery can take longer if many ALID updates are required.

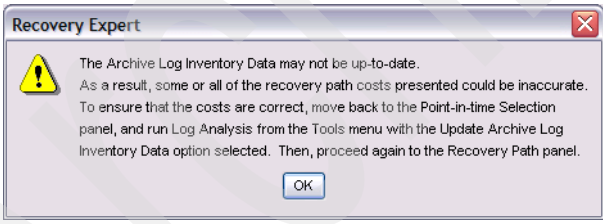


Figure 8-9 ALID warning message

After clearing the warning, you see the Recovery Path window. As we might expect, the path that DB2 RE chooses is to restore from a backup image and rollforward, as shown in Figure 8-10.

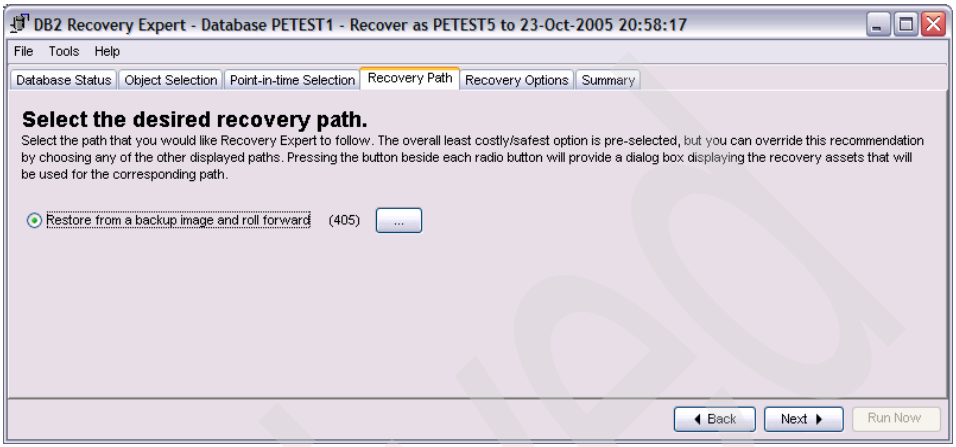


Figure 8-10 Recovery Path available

12. Click **Next** to navigate to the Recovery Options window (Figure 8-11 on page 225).

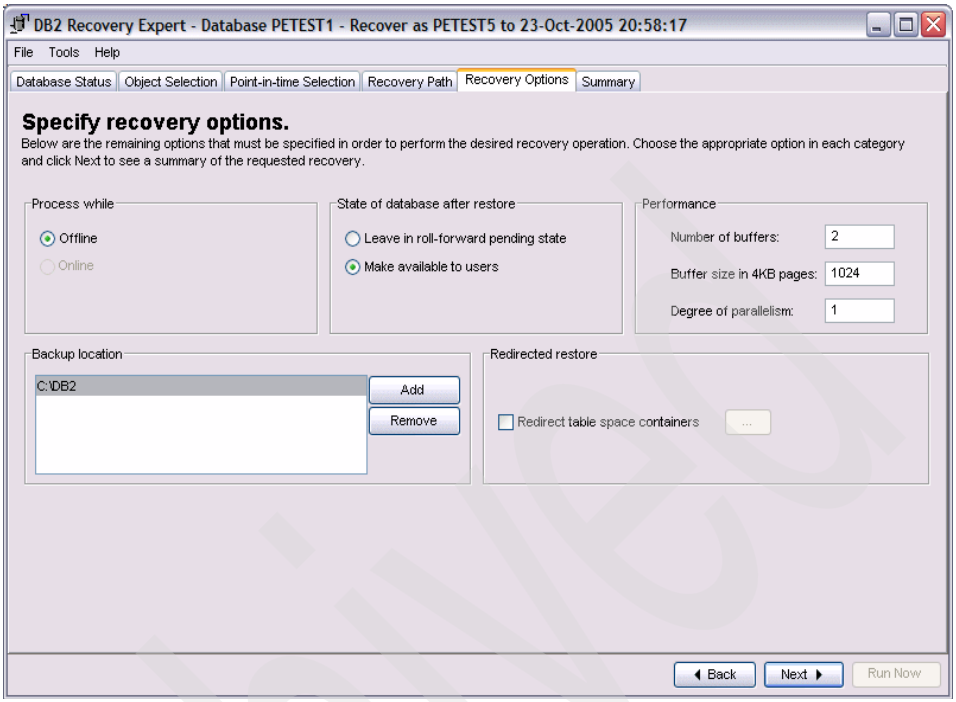


Figure 8-11 Recovery Options panel

In this scenario, we take all the defaults because we are using default table spaces so that DB2 RE can resolve the directory path where they go automatically.

The target database is processed while offline, and we choose to make the new database available to users.

13. Click **Next** to go to the summary panel (Figure 8-12 on page 226).

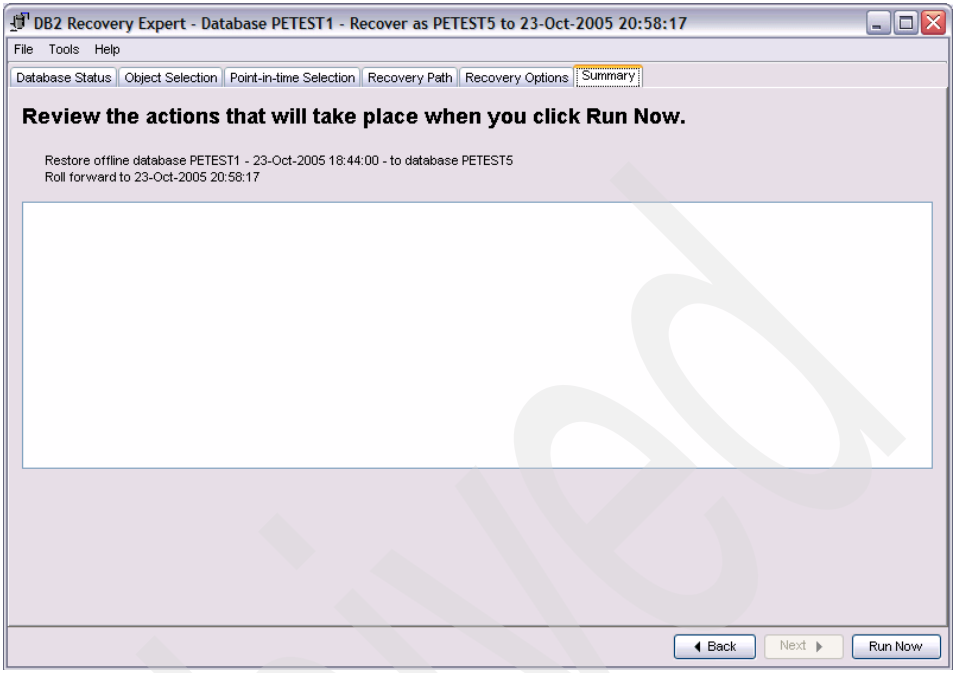


Figure 8-12 Summary panel before update

This is the last chance to review what DB2 RE is about to execute. It shows the date of the backup to which it is restoring and the date and time to which the logs are rolled forward.

14. Click **Run Now** to create the clone database PETEST5.

The final panel of the scenario confirms the actions DB2 RE has taken (Figure 8-13 on page 227).

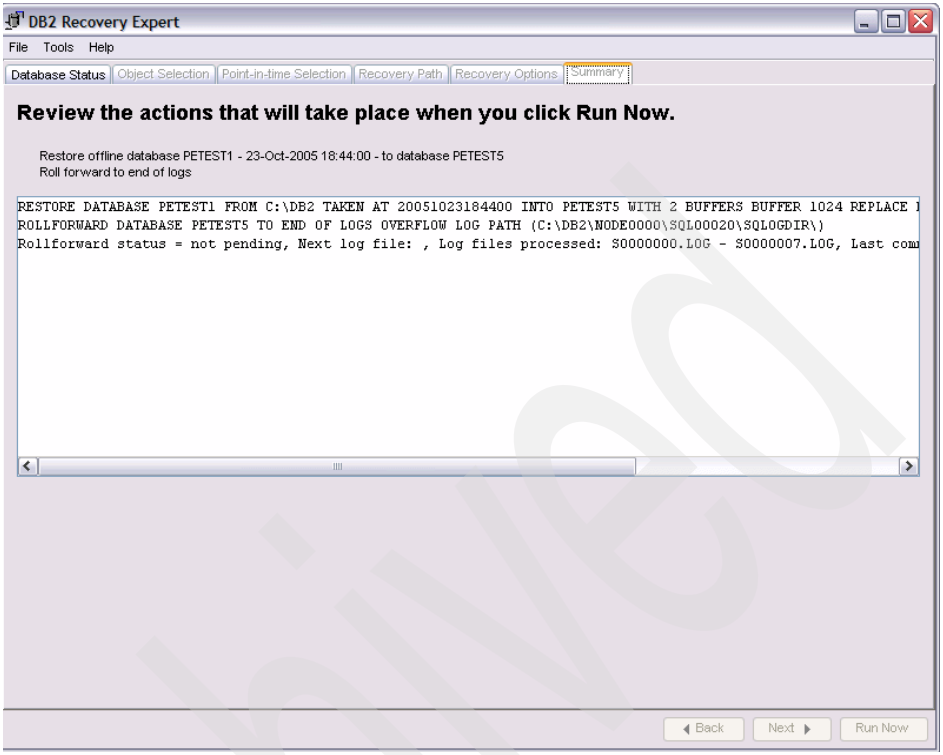


Figure 8-13 Summary of actions involved in cloning a database

We can see the backup copy that was used and the log files that were used in the creation of the new database. We can check that the new database has been created and that it reflects the latest updates of a total of 40,600 rows in the RACER table (Example 8-2).

Example 8-2 Connect to new database and verify data

```
C:\Program Files\IBM\SQLLIB\BIN>db2 connect to PETEST5
   Database Connection Information
 Database server        = DB2/NT 8.2.2
 SQL authorization ID   = KEVINTAY...
 Local database alias   = PETEST5
C:\Program Files\IBM\SQLLIB\BIN>db2 select count(*) from racer

1
-----------
      40600
  1 record(s) selected.
```

8.2 Introducing the Object Translation tool db2ox

You can use the DB2 RE Object Translation tool, **db2ox**, for the following tasks from the command line:

- For a full table space recovery from a backup

  This task clones or recovers a table space to the state of the specified backup.

- To extract the data for one or more tables from a backup

  This task extracts table data from the specified backup. The data can be rolled forward to a different point-in-time using the Log Analysis tool **db2la** from the command line. This cannot be done in native DB2 at the table level.

The **db2ox** command is also called from the GUI as part of the work flow to extract data from backups as part of the recovery process.

This section explains the process of extracting table data and the method of translating the table space and table identifiers to match the target when necessary.

8.2.1 Object Translation Tool constraints

Following are limitations of **db2ox**:

- Table space recovery or cloning can only occur on the same machine, within the same instance.
- The data file extracted for cloning a table has to be shipped manually if the target is on another machine.
- The target objects must all be created manually before executing the **db2ox** command.

  This includes the table space and all its tables within the table space for table space cloning and all tables and their indexes for table cloning.

- The backups from which the data is extracted must be offline or the data integrity cannot be guaranteed.
- Object Translation does not support any referential integrity and does not interface with Grouper.

  You must be responsible for maintaining integrity between tables.

- Object Translation does not support multiple instances.
- The **-X** translation map is highly recommended instead of the **-o** owner, **-n** name, **-T TID** and **-F FID** parameters

This is especially true in a multiple partition environment due to the restriction of the **-o** and **-n** parameters being applicable only to the database catalog partition.

Note: We do not recommend using **db2ox** for table space recovery manually from the command line. It is more pragmatic to let the GUI handle calling **db2ox** for this process, as one of the recovery steps. You should only use the **db2ox** command from the command line to extract table data and to clone table space.

8.2.2 Cloning using the object translation tool

A full offline backup image of the source database is required as the starting point for cloning using the object translation tool. When using **db2ox**, table spaces and tables are usually referred to by their table space identifiers (TIDs) and their table file identifiers (FIDs). These can be found in the DB2 catalog using the query that is shown in Example 8-3.

Example 8-3 Displaying the TIDs and FIDs

```
db2 select tbspace, name, tid, fid from sysibm.systables where name =
'DEPARTMENT'
TBSPACE             NAME                TID    FID
------------------- ------------------- ------ ------
USERSPACE1          DEPARTMENT               2      4

1 record(s) selected.
```

8.3 Object translation help text

There are 17 parameters that you can use when executing the Object Translation tool. Although this number of parameters can seem overwhelming, it is not as complicated as it sounds. Some of these parameters are mandatory, some are mutually exclusive, and others we recommend that you never use. If you execute the **db2ox** program with the help **-h** parameter, you see the following command options:

```
C:\Documents and Settings\Administrator>db2ox -h
DB2 Object Translation Tool.  Version 2.
Usage: db2ox -d database -b "YYYY/MM/DD HH:MM:SS" [-r targetdb]
 [-O nodenum] [-u username [-p[ password]]] [-N target_nodenum]
 [-o tableowner] [-n tablename] [-T TID] [-F FID]
 [-e { IXF | DEL } ] [-f output_directory]
```

```
 [-X OLDTID1:OLDFID1[-NEWTID1:NEWFID1][ OLDTID2:OLDFID2[-NEWTID2:NEWFID2]]
...]
 [-B backup_base_path] [-v] [-h]
```

Table 8-1 lists command line options for **db2ox**.

Table 8-1 DB2 RE Object Translation tool input parameters

Parameter	Description
-b	Specifies the date and time of source backup image. *Option cannot be omitted.*
-B	Specifies base path of backup images for database. If omitted, **db2ox** will try to locate backup images automatically.
-d	Specifies the database name, tables of which would be used for object translation. *Option cannot be omitted.*
-e	Specifies that **db2ox** must perform table extraction. Supported formats for table extract are DEL and IXF.
-f	Specifies output directory name in extract mode. If omitted, the current working directory is used.
-F	Specifies the numeric identifier of the table to extract in single table mode.
-h	Displays help on the product.
-n	Specifies the name of the table to extract in single table mode.
-N	Specifies the node number of the target database for table space recovery. If omitted, the default node number is used.
-o	Specifies the schema for the table to extract in single table mode.
-O	Specifies the node number of the source database. If omitted, default node number is used.
-p	Specifies the password for connecting to DB2. If a value is not specified the user is prompted for the password. If omitted, no password is used.
-r	Specifies the target database name for table space recovery. If omitted, the source database is used.
-T	Specifies the numeric table space identifier for the table to extract in single table mode.
-u	Specifies the user for connecting to DB2. If omitted, an implicit connection is established.

Parameter	Description
-v	Displays the version of the product.
-X	Specifies the object identifier map. For table extract mode only the old TID:FID pairs are required. For table space recovery the new TID:FID pair is required for translation.

You can use **db2ox** to clone or to extract a table space and all of its tables to the same or to a target database. Only a small subset of the possible parameters are relevant for this purpose. Typically, you should use the command in the form as shown in Example 8-4.

Example 8-4 Generic form of the db2ox command in table space recovery mode

```
db2ox -d source_database -b source_backup_timestamp -r target_database -X
"oldTID1:oldFID1-newTID1:newFID1 oldTID2:oldFID2-newTID2:newFID2"
```

In the case of a multiple partitioned table, execute **db2ox** against each partition by appending the command with **-N *partition_number***.

When the **-e** and **-f** parameters are not coded, you are running in table space extract mode. You must always use the **-e** and the **-f** parameters together for table extraction mode. More specifically, we recommend that you set the **-e** file format always to IXF. Currently the DEL format is not supported.

We also recommend that you not use the **-o** owner, **-n** table name, **-T** table space identifier, and **-F** table identifier parameter combinations, but instead always use the **-X TID/FID** translation map parameter. This is because the **-o** and **-n** combination can be used only with the database catalog partition and **-T** and **-F** can be used only in single table mode.

Example 8-5 shows the syntax of using Object Translation in table extract mode for three tables done simply using the translation map parameter.

Example 8-5 db2ox used in table extraction mode for multiple tables

```
C:\RE\db2ox\>db2ox -d sample -b 20051014143145 -e ixf -f C:\RE\db2ox\department
-X "2:3 2:4 2:5"
```

Note: When using **db2ox** for table space extraction, *do not* use the **-e** parameter because this parameter is related to file output, which is not applicable to table space extraction. Use the **-r** parameter to specify the target database. This can be the same as the source database.

When extracting a table, use the **-e** and **-f** parameters to create a data file as output. This file is then used to populate the target table.

8.4 Cloning a table space or table using db2ox

This section demonstrates the steps of using **db2ox** to clone a table space and tables.

8.4.1 Cloning a table space

The following steps are required to clone a table space using the object translation tool (**db2ox**):

1. Confirm that you have a full offline or table space backup of the source database, and determine the timestamp to use in the **db2ox** execution.
2. Extract the DDL for the table space and tables required from the source database.
3. Execute the DDL in the target database to create the new objects.
4. Determine the TIDs and FIDs of the source tables and the TIDs and FIDs of the new target tables, which you code in the **-X** parameter translation map.
5. Create a backup image of the target database so **db2ox** can resolve the transformations between source and target objects. This can be a full database backup or an online table space backup. The target backup is just used as a template.
6. Run **db2ox** against the source database backup image.
7. Rollforward the target database.

Restriction: When you clone a table space, *all* the tables in the source table space must be extracted and *all* the tables must exist in the target table space before running **db2ox**.

8.4.2 Scenario to clone a table space that contains three tables

This scenario uses the RELAY table space that contains three tables in the source database PETEST1 and creates a clone of the complete table space to

the target database RETEST1. The data in the table space can be cloned from any offline backup providing the table structures are still current. We go through the seven steps for this scenario:

1. To go back to a previous point-in-time, select an existing backup of the source database from the date for which you want the target data to be created.

   For this scenario, we create a new backup, run the backup command from the DB2 CLP:

   ```
   BACKUP DATABASE PETEST1 TO "C:\DB2" WITH 2 BUFFERS BUFFER 1024 PARALLELISM
   1 WITHOUT PROMPTING
   ```

   The time stamp of the backup is 20051030214354, which is used for the **-b** parameter of the **db2ox** command. The **-b** parameter identifies the backup of the source database.

   You do not specify the backup image of the target database in the command, but the target backup must exist.

2. Extract the DDL for the table space RELAY and its tables RACE, RACER, and VAN1 from the source database PETEST1.

   Extract the DDL by running the **db2look** command against the source objects or by running the Generate DDL option from the DB2 Control Center.

3. Connect to the target database RETEST1 and execute the DDL to create the RELAY table space and its three tables.

4. Determine the TIDs and FIDs of the source and table objects.

   Connect to the PETEST1 database and run the following SQL (Example 8-6) from the Control Center or the DB2 CLP:

Example 8-6 Determine the TIDs and FIDs of the source and target databases

```
SELECT name, tbspace, TID, FID FROM SYSIBM.SYSTABLES where tbspace = 'RELAY';
```

   The TIDs and FIDs for the source objects are displayed in Figure 8-14. These are used to code the translation map in the -X parameter.

NAME	TBSPACE	TID	FID
RACE	RELAY	5	2
RACER	RELAY	5	3
VAN1	RELAY	5	4

Figure 8-14 Display the TIDs and FIDs for source objects

   Disconnect from the source database, connect to the target database RETEST1, and re-run the SQL in Example 8-6. The TIDs and FIDs for the

target objects are displayed in Figure 8-15. These are the corresponding IDs to which the source IDs are mapped in the translation map.

NAME	TBSPACE	TID	FID
RACE	RELAY	3	2
RACER	RELAY	3	3
VAN1	RELAY	3	4

Figure 8-15 Display the TIDs and FIDs for target objects

When coding the translation map, the **-X** parameter on the **db2ox** command map the source TID:FID with the target TID:FID.

By coding the **-X** parameter translation map correctly, it is not necessary to code the **-F FID**, **-T TID**, **-o** owner, or **-n** name parameters.

Tip: Only code the **-X** parameter instead of using the **-F**, **-T**, **-o**, or **-n** parameters. The **-X** parameter translation map displays source to target mappings in a much clearer format.

5. Take a table space backup of the newly created target table space RELAY in RETEST1.

   Example 8-7 is a sample command for taking the target backup.

Example 8-7 Backup command for target table space

```
BACKUP DATABASE PETEST1 TABLESPACE ( RELAY ) TO "C:\DB2" WITH 2 BUFFERS BUFFER
1024 PARALLELISM 1 WITHOUT PROMPTING;
```

   The target tables can be empty before the backup is taken, because the tables are populated by running the **db2ox** command.

   The time stamp of the target backup is not coded in the **db2ox** command.

   If the backup of the target database does not exist, the following error message is returned:

```
C:\Program Files\IBM\SQLLIB\BIN>db2ox -d pedemo -b 20051020171957 -r
petest1 -X "11:2 11:3"
ARY0104E Target backup image does not exist
```

6. Run the **db2ox** command to populate the target tables. For this scenario the correct command is given in Example 8-8.

Example 8-8 The db2ox command to populate target tables

```
C:\Documents and Settings\Administrator>db2ox -d petest1 -b 20051030214354 -r
retest1 -X "5:2-3:2 5:3-3:3 5:4-3:4"
```

If the **db2ox** command has completed successfully, *no message is returned* in the command line.

To check whether **db2ox** has worked correctly, check the target table space status. Connect to the RETEST1 database and issue the list table space command. The target table space RELAY is in rollforward pending status, as shown in Example 8-9. Now, we can move to the next and final step to execute the rollforward command.

Example 8-9 Check target table space status after db2ox command has run

```
C:\Program Files\IBM\SQLLIB\BIN>db2 list tablespaces

Tablespace ID                        = 3
 Name                                 = RELAY
 Type                                 = System managed space
 Contents                             = Any data
 State                                = 0x0080
   Detailed explanation:
     Roll forward pending
```

7. Rollforward the target table space.

   If you attempt to access one of the new target tables at this point, you see the SQL error message shown in Figure 8-16.

Figure 8-16 SQL error for rollforward pending

Rollforward the target table space RELAY to end of logs as shown in Example 8-10 to return the table space to normal, which makes it accessible to users.

Example 8-10 Issue rollforward command for target tables pace

```
C:\Program Files\IBM\SQLLIB\BIN>db2 "rollforward db retest1 to end of logs and
complete tablespace (RELAY) online"

                                Rollforward Status
```

```
 Input database alias                   = retest1
 Number of nodes have returned status   = 1

 Node number                            = 0
 Rollforward status                     = not pending
 Next log file to be read               =
 Log files processed                    =  -
 Last committed transaction             = 1970-01-01-00.00.00.000000

DB20000I  The ROLLFORWARD command completed successfully.
```

A count on the three cloned tables shows that they have been created to the point-in-time of the backup, not to the current time of the source database, as shown in Example 8-11.

Example 8-11 Record count of target tables

```
C:\Program Files\IBM\SQLLIB\BIN>db2 select count(*) from RACE
1
-----------
      40000

  1 record(s) selected.

C:\Program Files\IBM\SQLLIB\BIN>db2 select count(*) from RACER
1
-----------
        100

  1 record(s) selected.

C:\Program Files\IBM\SQLLIB\BIN>db2 select count(*) from VAN1
1
-----------
          0

  1 record(s) selected.
```

The target tables are now available and ready to access. They are at the point-in-time of the backup copy taken of the source database coded by the timestamp in the **-b** parameter of the **db2ox** command.

8.4.3 Error scenarios with db2ox in table space recovery mode

There are several circumstances in which you might see error messages when trying to use **db2ox**. This section lists these known circumstances .

LOGRETAIN not set on for target database

If you have not set the LOGRETAIN parameter for the target database to RECOVERY, you see this error message:

```
ARY0348E SQL2651N The log records associated with the database cannot be
asynchronously read.
```

Target database must have corresponding table space

The table space you are cloning must have a corresponding table space in the target database. Otherwise, the following message is returned:

```
C:\Program Files\IBM\SQLLIB\BIN>db2ox -d petest2 -b 20051020171957
-r retest2 -X "11:2 11:3"
ARY0348E SQL1008N  Invalid table space ID.  SQLSTATE=58036
```

Target table space must have corresponding tables

The target table space must also have all the table objects that you want to clone created in the target table space before you run the **db2ox** command. In the next example, the translation map has been updated to reflect the new TID values for the target table space. However, the table objects do not exist. So, the following message is returned:

```
C:\Program Files\IBM\SQLLIB\BIN>db2ox -d petest2 -b 20051020171957
-r retest2 -X "11:2-5:2 11:3-5:3"
ARY0250E Unable to access table TID=5, FID=2
```

Here, we have specified the correct new TID value for the newly created RELAY table space in the PETEST1 target database. However, the new TID/FID combination is meaningless to **db2ox** because the two tables in the target table space have not been created.

The target tables must exactly match the source tables

If the target table space has more tables than the source table space, corruption of the database occurs. Those extra tables become unavailable after running **db2ox**.

If you execute **db2ox** with extra tables at the target, you see the following error message:

```
C:\Program Files\IBM\SQLLIB\BIN>db2ox -d petest2 -b 20051020171957
-r retest2 -X "11:2-5:3 11:3-5:2"
ARY0348E SQL2549N  The database was not restored because either all of the
table spaces in the backup image are inaccessible, or one or more table
space names in list of table space names to restore are invalid.
```

Do not extract to a target table space that has more tables than you are writing to. If the target table space is a super-set of source tables, then use **db2ox** in table extraction mode instead of table space recovery mode.

If you do attempt to use **db2ox** to map to a target table space where the target has extra tables, you corrupt all the other tables in the target table space. When you attempt to access any of these other tables, you receive an SQL1034C error message informing you that the database has been damaged and that all transactions processing the database have been terminated (Figure 8-17).

You have to remedy this situation by recovering to a previous backup image of the database.

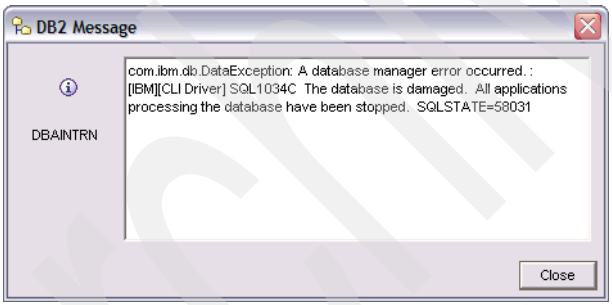


Figure 8-17 Corruption of table space being overwritten

If you only need a subset of the tables extracted from the source table space, use the table extraction form of the **db2ox** command, as discussed in 8.4.4, “Cloning a table” on page 239. You list the TID:FID pairs in the translation map and this creates an separate IXF for each table that is ready to be imported or loaded into the target tables.

Important: When extracting a table space and several tables, make sure your target table space does *not* contain any extra tables that do not map to the source database. Otherwise, you target database will be corrupted. In this case, extract the data for the source tables that you need individually and load them individually.

8.4.4 Cloning a table

The following steps are required to clone a table using the object translation tool:

1. Confirm that you have a full offline backup of the source database, and determine the time stamp to use in the **-b** parameter for **db2ox** execution.
2. Determine the TIDs and FIDs of the source tables.
3. Extract the DDL for the tables required from the source database.
4. Execute the DDL in the target database to create the new tables.
5. Run **db2ox** against the source database backup image.
6. Populate the target tables using the IXF data files created by **db2ox**.

The DEPARTMENT table in the SAMPLE database, for example, can be cloned into the same database or a different database. Clearly, if you clone the table into the same database, it has to have a different name and different table identifiers. If the table is truly cloned into a separate database, you can choose to create it with the same name and table space and table identifiers providing the TID/FID combination does not already exist.

The structure and data of a table can be extracted using the following basic command from a command line:

```
db2ox -d source_database -b source_backup_timestamp -e IXF -f
IXF_output_file_path -X “oldtid1:oldfid1 oldtid2:oldfid2”
```

Important: When extracting a table, always use the **-e** and **-f** parameters to create a data file as output. This file is then used to populate the target table.

8.4.5 Scenario to clone a table

In this scenario, we extract the data for the VAN1 table from a full offline database backup image of the PEDEMO source database, write the output file to the C:\RE directory, and import the data into the target database PETEST1:

1. Determine which source database backup file from which you want to extract the data.

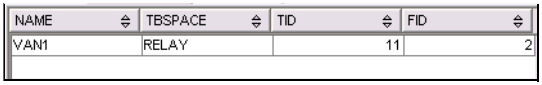
   Use the backup time stamp in the **-b** parameter in the **db2ox** command.

2. Next determine the TID and FID of the source table VAN1.

   Connect to the PEDEMO database and run the following SQL from the Control Center or the DB2 CLP:

   ```
   SELECT name, tbspace, TID, FID FROM SYSIBM.SYSTABLES where name = 'VAN1';
   ```

   Figure 8-18 on page 240 shows the TID and FID for the source object. These are used to code the translation map in the **-X** parameter for **db2ox**.



NAME	TBSPACE	TID	FID
VAN1	RELAY	11	2

Figure 8-18 Display of TID and FID in the DB2 Control Center

3. Extract the DDL for the VAN1 table from the source database PEDEMO.

   Extract the DDL by running the db2look command against the source objects or by running the Generate DDL option from the DB2 Control Center.

4. Connect to the target database PETEST1 and execute the DDL to create the VAN1 table.

5. Run the **db2ox** command from a command line

   Code the source database name, the source database backup timestamp, the **-e** parameter and the **-f** parameter for the file output path and finally the **-X** translation with the VAN1 TID and FID. See Example 8-12.

Example 8-12 Run the db2ox command to create the IXF file

```
C:\Documents and Settings\Administrator>db2ox -d PEDEMO -b 20051020171957 -e
ixf -f C:\RE\ -X 11:2
```

   If the **db2ox** command is successful, no message is returned in the command window. To verify the command has run successfully check the IXF file named [11].[2].IXF has been created in the folder specified, C:\RE in this case. This output file is named after the TID:FID combination of the source table, NOT the target table.

   This file can be used to load or import the data into the new table. The target table is identified in the load or import command.

6. Populate the target table using the IXF data files created by **db2ox**.

   Populate the target table by connecting to the PETEST1 database and running the import command. In this example the VAN1 table contains a GENERATED ALWAYS identity column, which is coded to be ignored on the import statement; the IDENTITYIGNORE clause, see Example 8-13.

Example 8-13 Import the data into the target table

```
C:\Program Files\IBM\SQLLIB\BIN>db2 IMPORT FROM "C:\RE\[11].[2].ixf" OF IXF
MODIFIED BY IDENTITYIGNORE MESSAGES "C:\RE\expmsg.txt" INSERT INTO VAN1
Number of rows read         = 25000
Number of rows skipped      = 0
Number of rows inserted     = 25000
Number of rows updated      = 0
Number of rows rejected     = 0
Number of rows committed    = 25000
```

Now the target table is populated successfully with data from the point-in-time from which the source backup copy was taken, and is available to users.

If you want the target table to be cloned to a different point-in-time, you can run the **db21a** command against the source table to generate undo or redo SQL. This SQL can then be run against the target table to effectively go or forward from the backup timestamp.

8.4.6 Extracting data for several tables at once using db2ox

You can extract several tables from several table spaces in the same **db2ox** command to create a separate IXF file for each table, as shown in Example 8-14.

Example 8-14 Extracting data from multiple tables from multiple table spaces

```
C:\Documents and Settings\Administrator>db2ox -d PEDEMO -b 20051020171957 -e
ixf -f C:\RE\ -X "11:2 11:3 11:5 13:3 13:4"
```

This example produces five IXF files, each of which can be used individually to populate the five target tables. The target tables must, of course, be created first.

8.4.7 Restriction using the IXF file generated from db2ox

It is also worth noting there is a restriction with the IXF file that **db2ox** creates. Using Import with the `create into` clause will not create the target table correctly, as shown in Example 8-15. The column names are created as C1, C2, and so on. This is because DB2 RE cannot determine the table column names at this point. Thus, you must *always* create the target table *first* before loading or importing the data.

Example 8-15 Incorrect use of the output IXF file

```
C:\Program Files\IBM\SQLLIB\BIN>db2 IMPORT FROM "C:\RE\[11].[2].ixf" OF IXF
MODIFIED BY IDENTITYIGNORE MESSAGES "C:\RE\expmsg.txt" CREATE INTO VAN1
```

Restriction: Although **db2ox** creates an IXF, do not use the `create into` clause to create the target table, because the table definition will not be accurate.

8.4.8 Using sample program EasyOX to automate cloning tables

EasyOX is a sample Perl script that is designed to perform the tasks that are necessary to create a duplicate table, at a selected point-in-time, on the local instance.

EasyOX is automating steps via a script, all of which could all be executed individually without EasyOX. This sample program is intended to demonstrate DB2 RE facilities and usage. It is just a sample, which you can modify for remote use or other needs in your environment.

EasyOx extracts the metadata that is necessary to drive the process automatically and passes that metadata to the down stream steps that require it. Remote cloning, other points in time, and other schema versions can be achieved by modifying the sample script that is provided, thus reducing the number of steps to clone your tables into a target database and reducing the likelihood of errors.

You can download the EasyOX zipped file from the IBM Redbook Web site and extract the files into a temporary directory. Copy the files into the DB2 RE bin directory, and tailor the files as described. For download details, refer to Appendix C, “Additional material” on page 375.

The only extra software you might need to run EasyOX on the machine where you have installed DB2 RE is ActivePerl. ActiveState’s ActivePerl distribution for Windows and AIX can be downloaded free from their Web site.

Attention: EasyOX is a sample set of programs and is provided as is for demonstration and example purposes only. It is not covered under the product warranty that comes with IBM DB2 Recovery Expert for Multiplatforms.

The three main steps are as follows:

1. Determine the time stamp of a source database backup.
2. Tailor, and then run the EasyOX table space create job easyox00.bat for the target database, in case the target objects do not already exist.
3. Tailor, then run the **easyox01** program which calls the Perl script.

You need to provide the following input parameters to the Perl script program by updating the easyox01.bat file:

-sdb	source_database
-u	user ID
-p	password
-o	source_table_owner
-n	source_table_name
-cn	target_table_name
-tts	target_table_space
-bts	source_database_backup_timestamp
-lts	recovery_point_in_time
-tdb	target_database
-bdir	source_backup_directory
-tsbdir	target_tablespace_container_directory

There are a total of four files supplied:

- Initialize the demo environment (easyox00.bat)
- Execute the demo EasyOX script (easyox01.bat)
- Print demo results (easyox02.bat)
- Perl script file (easyox.pl)

As well as reducing the number of steps in the cloning process, EasyOX has another major benefit. EasyOX includes a rollforward option using the DB2 RE Log Analysis tool, **db2la**. This tool allows you to rollforward the target tables to any point-in-time by creating redo SQL.

Note: We ran the example discussed here on Windows. If you want to run on AIX, be sure to update the Perl script file easyox.pl to run **db2la** successfully. Comment out the first line and uncomment the second as follows:

```
$db2la = "db2la";
#$db2la ="/opt/IBM/DB2RecoveryExpert/bin/db2recex db2la";
```

Copy the three files into DB2 bin installation directory so that you can execute them from the DB2 CLP window. Then, follow these steps:

1. Update the easyox00.bat file with the target database and table space and run the following:
   - Change the target database to SAMPRE
   - Change the table space to DEMOXTS
   - Execute the easyox00.bat file

   The creation of the target table space in database SAMPRE is seen in Example 8-16 on page 244.

Example 8-16 Run easyox00.bat to create target table space

```
C:\Program Files\IBM\SQLLIB\BIN>easyox00
reset connection because
SQL1024N  A database connection does not exist.  SQLSTATE=08003
setup and creation of demotablespace demoxts

   Database Connection Information

 Database server        = DB2/NT 8.2.2
 SQL authorization ID   = KEVINTAY...
 Local database alias   = SAMPRE

DB21034E  The command was processed as an SQL statement because it was not a
valid Command Line Processor command.  During SQL processing it returned:
SQL0204N  "DEMOXTS" is an undefined name.  SQLSTATE=42704
DB21034E  The command was processed as an SQL statement because it was not a
valid Command Line Processor command.  During SQL processing it returned:
SQL0601N  The name of the object to be created is identical to the existing
name "BP1" of type "BUFFERPOOL".  SQLSTATE=42710
DB20000I  The SQL command completed successfully.
DB20000I  The SQL command completed successfully.
DB20000I  The SQL command completed successfully.
DB20000I  The SQL command completed successfully.
```

2. Update easyox01 file to provide the parameters for the Perl script as follows:
   - -sdb RETEST1
   - -u *userid*
   - -p *password*
   - -o KEVINTAYLOR
   - -n CAR
   - -cn CARNEW
   - -tts DEMOXTS
   - -bts “2005/10/31 14:50:35”
   - -lts “2005/10/31 15:05:00”
   - -tdb SAMPRE
   - -bdir C:\DB2\RETEST1.0\DB2\NODE0000\CATN0000\20051031
   - -tsbdir C:\DB2\NODE0000
3. Execute the easyox01.bat file from the DB2 CLP using the following command line:

   ```
   C:\Program Files\IBM\SQLLIB\BIN>easyox01
   ```

The easyox01 batch file executes the following processes automatically:

- EasyOX setup steps:
  i. Cleanup, drop target table CARNEW
  ii. Setup source, get table space IDs and table IDs (TIDs/FIDs)
  iii. Setup source, get DDL
  iv. Setup target, create table
  v. Setup target, get table space IDs and table IDs (TIDs/FIDs)
- Table copy and translation steps:
  i. Use the DB2 RE db2ox to extract table and automatically load table into target table space.
  ii. Backup target table space

- Rollforward steps:
  i. Use the DB2 RE db2la to extract REDO SQL from logs.
  ii. Modify REDO SQL to point to the target table.
  iii. Run REDO SQL to rollforward the target table.

The third batch file, easyox02, is supplied so that you can easily see the before, or source, condition and the after, or target, condition.

Update the easyox02 file with the names of the source and target databases together with the source and target table. Then, from the DB2 CLP run the easyox02 file, and output shows the before and after, as shown in Example 8-17.

Example 8-17 Execute easyox02

```
C:\Program Files\IBM\SQLLIB\BIN>easyox02

   Database Connection Information

 Database server        = DB2/NT 8.2.2
 SQL authorization ID   = KEVINTAY...
 Local database alias   = RETEST1

C:\Program Files\IBM\SQLLIB\BIN>DB2 Select * from car

NAME       NUMBER      DESCRIPTION
---------- ----------- ---------------
FORD                99 MUSTANG
MERCEDES           101 E320 CDI
ESCALADE         54000 SUV
RANGEROVER         601 V8
HUMMER             701 FUH2

  5 record(s) selected.
```

```
C:\Program Files\IBM\SQLLIB\BIN>DB2 connect reset
DB20000I  The SQL command completed successfully.

C:\Program Files\IBM\SQLLIB\BIN>DB2 connect to sampre

   Database Connection Information

 Database server        = DB2/NT 8.2.2
 SQL authorization ID   = KEVINTAY...
 Local database alias   = SAMPRE

C:\Program Files\IBM\SQLLIB\BIN>echo on

C:\Program Files\IBM\SQLLIB\BIN>DB2 Select * from carnew

NAME       NUMBER      DESCRIPTION
---------- ----------- ---------------
MERCEDES           101 E320 CDI
ESCALADE         54000 SUV
HUMMER             701 FUH2

  3 record(s) selected.

C:\Program Files\IBM\SQLLIB\BIN>DB2 connect reset
DB20000I  The SQL command completed successfully.
```

We can see the source table CAR currently has five rows, while the target table CARNEW has only has three rows. The source table has had an extra two rows inserted, while the new target tables reflects the situation at a previous point-in-time.

9

Fast Backup

This chapter introduces the concept of DB2 Recovery Expert Fast Backup, discusses the planning consideration for using Fast Backup, and explains how to install Fast Backup. It also describes how to prepare and use IBM DB2 Recovery Expert Fast Backup features and functions. This chapter uses the following scenarios to demonstrate Fast Backup:

- Scenario A: Recovering a table space
- Scenario B: Recovering a database on the multiple network storage volume

It includes the following sections:

9.1 About DB2 Recovery Expert Fast Backup

DB2 Recovery Expert (DB2 RE) Fast Backup provides high performance backup and restore capability on the large scale volumes for the database with SAN or NAS storage systems. Fast Backup uses the hardware level backup and restore function such as SAN volume copy or NAS snapshot, as illustrated in Figure 9-1.

However, you do not need to know any network storage volumes to use this backup or restore, even if a database or table space is spread across multiple storage volumes. You only have to point to a database or table space which you want to back up or restore. Fast Backup detects all storage volumes to be backed up or restored automatically. It invokes the volume level copy function on each storage volume and backs up or restores the specified database or table space.

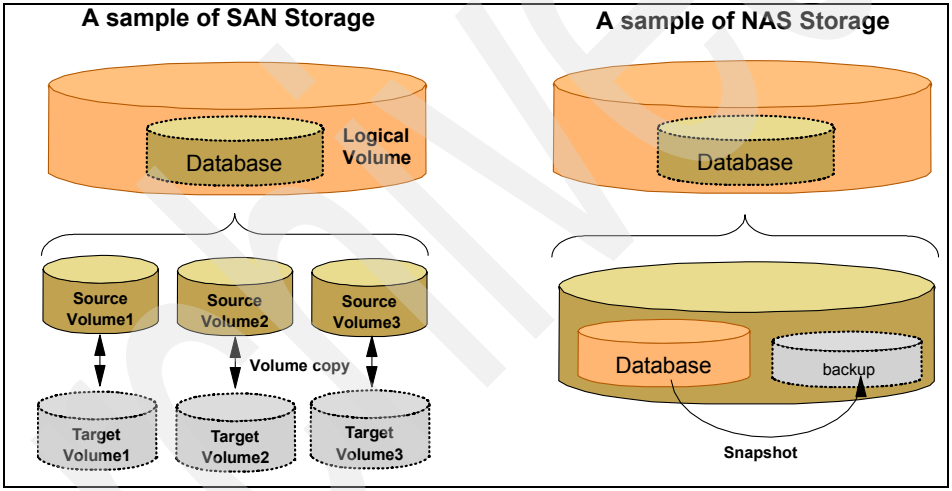


Figure 9-1 Hardware level backup function of the network storage

Fast Backup supports some of SAN or NAS hardware level backup and restore function. SAN provides high-speed back capability, and NAS has its cost merit and flexibility. Which you choose depends on your system structure. In this book, we explain Fast Backup with SAN network storage system.

The backup of Fast Backup is one of the recovery paths for DB2 RE. When a backup of Fast Backup is selected as a recovery path on the DB2 RE window and the recovery run, the restore function of Fast Backup is invoked and the hardware level restore operation is performed.

9.1.1 Fast Backup module structure

Fast Backup consists of the client module and the server module. To use Fast Backup, you need to install DB2 RE with the Fast Backup custom option and Fast Backup client on DB2 Windows client.

Fast Backup client module

Fast Backup client module provides the following user interfaces mainly for the backup process of Fast Backup:

- Fast Backup Administration window

  Configuring Fast Backup client setting and Network Storage setting and editing Fast Backup specifications.

- Fast Backup client window

  Launching Fast Backup backup process and creating backup specification.

You launch the Fast Backup backup process through the Fast Backup client window interface.

Fast Backup server module

You install the Fast Backup server module with the DB2 RE custom option. It is invoked by the Fast Backup client backup process and the DB2 RE recovery process.

9.1.2 Backup process

The backup process of Fast Backup is launched through Fast Backup client interface which is a DB2 Control Center plug-in, as shown in Figure 9-2 on page 250. This process is similar to launching a normal backup from the DB2 Control Center. Then Fast Backup invokes volume backup through the storage system interface and volume level copy. This copy process is much faster when compared with the usual copy operation on an operating system.

Attention: You can restore a database when you backup a database through Fast Backup. You can restore a table space when you do backup operation of Fast Backup to a table space. You *cannot* restore an individual table space if you backup a database through Fast Backup.

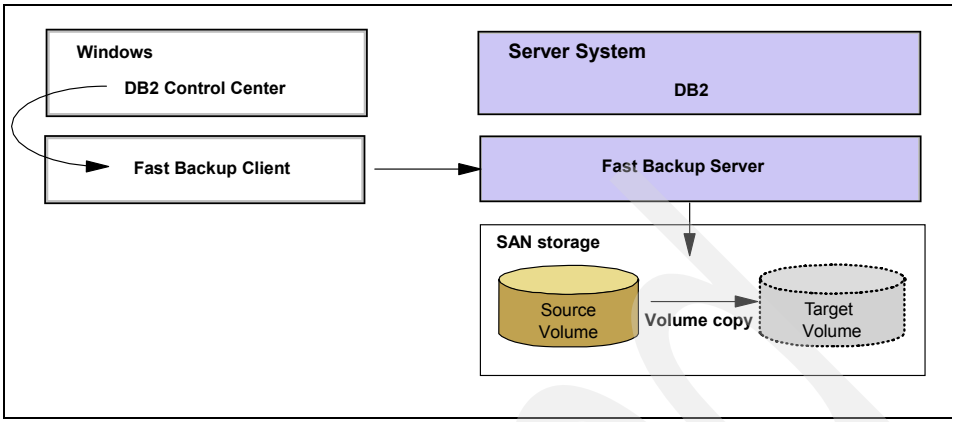


Figure 9-2 Backup process of Fast Backup concept

9.1.3 Restore process

The Fast Backup restore process works with DB2 RE. When you select a backup of Fast Backup as a recovery path and perform the recovery, DB2 RE invokes the Fast Backup restore process internally. Fast Backup provides two types of restore features: volume restore and object restore. This is specified in Fast Backup specification which is created on Fast Backup client window.

Note: You can specify volume restore or object restore at the point of backup. You cannot select volume restore or object restore at the point of restore.

Volume restore of Fast Backup

Volume restore is a typical use of Fast Backup. To reduce recovery time, you should select this restore method. Fast Backup volume restore process invokes the volume level copy operation in the network storage system, then whole volume is quickly restored, as shown in Figure 9-3 on page 251.

Object restore of Fast Backup

Fast Backup object restore is to restore a database or a table space object in the storage volume. Fast Backup provides this feature using a copy function of host operating system. This feature can be chosen if a database or table space is relatively small in the large storage volume and the remaining storage space is used as the archive directories. In this case, newer archived data will be replaced with older one in the target volume when you run Fast Backup volume restore.

This operation takes much longer than volume restore and its availability depends on the combination of OS and network storage system, as shown in Figure 9-3.

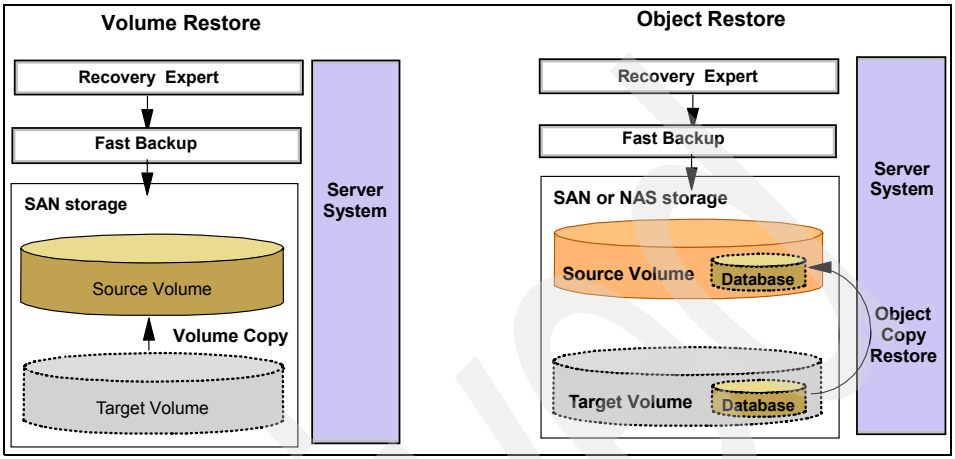


Figure 9-3 Volume restore and object restore of Fast Backup

9.2 Planning SAN network storage configuration

When you use Fast Backup, you need to discuss with your storage administrator and system administrator to configure and assign SAN network storage with the considerations for DB2 RE Fast Backup environment.

9.2.1 Planning SAN volume configuration

The volume level copy operates per unit of logical volume (LUN) in the network storage system. Fast Backup provides a backup and restore capability for a database or a table space in the network storage volumes. This means, in case of AIX and SAN environment, after you create a database on a file system on a LV (logical volume), you cannot create another database on another file system using different LV under the same VG (volume group), as shown in Figure 9-4 on page 252.

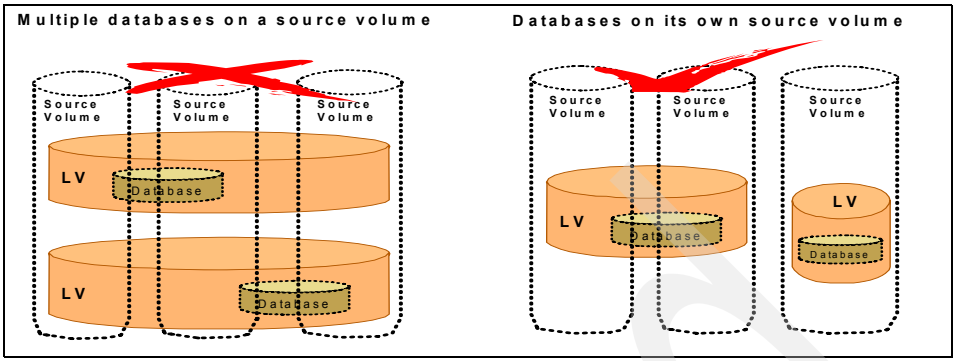


Figure 9-4 Logical Volume configuration

A target volume should be a dedicated volume for the specific backup task. If a target volume is used for a database backup first, then as a table space backup target volume, even though the table space is one of the table spaces in that database, the database backup cannot be used for database restore. To preserve the database backup copy, another volume should be use as the target for table space backup.

You should also consider the number of target volumes (backup target volumes). If you define three volumes for backup target volumes, you can have three historical backups. These backups will be useful as complementary backups between two offline full backups to reduce the recovery time. The target volumes size should be same as the source volumes, as shown in Figure 9-5.

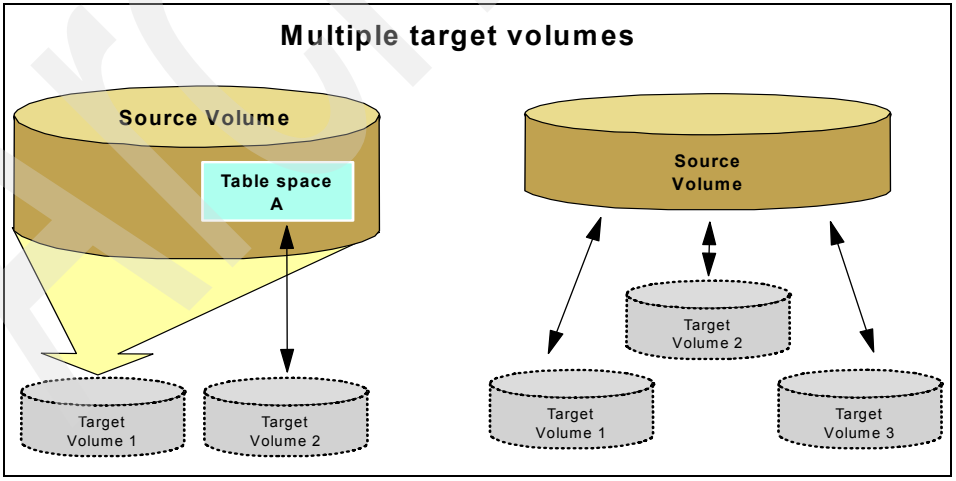


Figure 9-5 Logical Volume configuration on SAN and target volumes configuration

9.2.2 Assigning SAN volume to the system

How to assign the SAN network storage volumes on the host system depends on whether you plan to use volume restore or object restore. The source volumes must be assigned to the system and mounted on the system. However, the target volumes should not be assigned to the host system.

Volume assignment on volume restore

If you select to use *volume restore*, the source volumes must be assigned to and mounted on the host system, but the target volumes should not be assigned to and not be in active state (not be included in any volume group) on the host system, as shown in Figure 9-6 on page 254. If you assign the target volume in a VG (**lspv** shows the target physical volume state is active) on the host system and run Fast Backup, you might see the same physical volume ID on the system (PVID on AIX) and it can cause failure on the next Fast Backup operation.

Note: If you see the same PVID on AIX using the **lspv** command, you can reset it by using the following:

```
> reducevg -dv vgnamexx hdiskxx
> chdev -l hdiskxx -a pv=clear
```

The `hdiskxx` must be a backup target volume of Fast Backup operation. Use the ESSCLI script, rsList2105s.sh, to check the target IBM ESS volume.

Volume assignment on object restore

If you choose to use *object restore*, the source volumes must be assigned to and mounted on the system, and the target volumes must be assigned to the system but should not be included in any VG and not be mounted on the system, as shown Figure 9-6 on page 254. Fast Backup might show the following error if it is mounted:

```
an error on umount operation
```

If both methods are need, you can use this volume assignment.

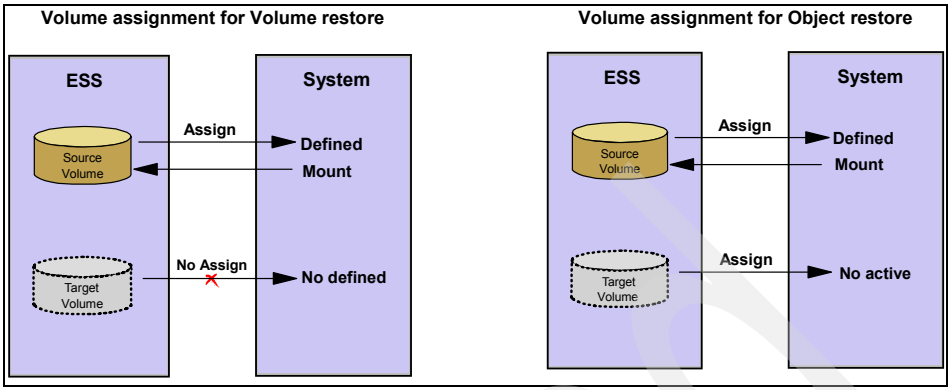


Figure 9-6 Volumes assignment consideration for Fast Backup

9.3 Fast Backup environment preparation

Fast Backup supports the following storage servers:

- IBM Enterprise Storage Server® Model F20, 800
- Hitachi Freedom Storage Lighting 9900V series
- EMC Symmetrix DMX800 Series
- NEC iStorage S Series S4100
- Netapp Filer F700 Series

For more details, refer to the hardware requirements listed in *IBM DB2 Recovery Expert for Multiplatforms User's Guide*, SC18-9564.

In this book, we demonstrate Fast Backup function with IBM ESS Model 800 storage system and AIX system.

To prepare for Fast Backup environment, the storage administrator needs to:

- Assign IBM ESS volumes
- Create IBM ESS FlashCopy® establishment tasks

The detailed steps are described in 9.3.2, “Configuring network storage, IBM ESS” on page 256.

Note: Fast Backup supports IBM ESS FlashCopy. In DB2 Recovery Expert for Multiplatform V2.1 FP1, Freeze FlashCopy Consistency Group option is also available for backup. To use this option, you also need to create the Consistency Group task on IBM ESS. Refer to the readme file for details. This option can help preventing consistency issues from multiple accessing to a source volume. On restoring, this option is not necessary because the source volume on restoring (a target volume on backing up) is dedicated to be used for the volume restore operation.

The system administrator needs to do assign and mount IBM ESS volumes on the system. The detailed steps are described in 9.3.3, “Configuring volumes on the system -AIX” on page 261.

The database administrator needs to be prepared to administer the Fast Backup installation as described in 9.3.5, “Preparing Fast Backup, Fast Backup Administration” on page 271.

9.3.1 Description of the lab environment

The operating systems that we used in the lab are:

Server system:	AIX v5.3 on pSeries LPAR
Client system:	MS Windows Server 2003
Network storage:	IBM ESS model 800, 5GB on each volume (8 LUNs)
Source volume ID:	201, 205
Target volume ID:	202, 203, 204, 206, 207, 208
Server system:	AIX v5.3 on pSeries LPAR
Client system:	MS Windows Server 2003

Figure 9-7 on page 256 and Table 9-1 on page 256 show the system and the network storage volumes that we configured in our lab environment for some scenarios in this chapter.

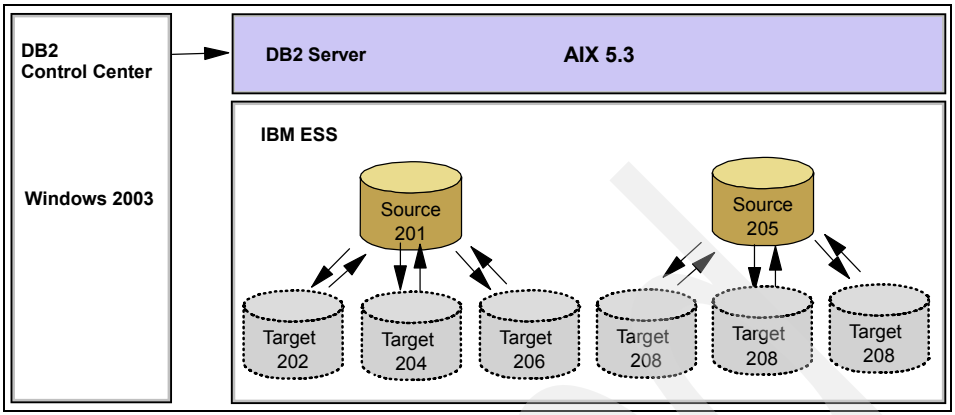


Figure 9-7 Lab environment for DB2 RE Fast Backup

Table 9-1 Volume configuration for Lab environment

src - tgt pair	src mount point	backup method	restore method
201 - 202	/ess1	FlashCopy, Freeze	FlashCopy
201 - 203	/ess1	FlashCopy, Freeze	FlashCopy
201 - 204	/ess1 (assign 204)	FlashCopy, Freeze	ObjectRestore
205 - 206	/ess2	FlashCopy, Freeze	FlashCopy
205 - 207	/ess2	FlashCopy, Freeze	FlashCopy
205 - 208	/ess2 (assign 208)	FlashCopy, Freeze	ObjectRestore

9.3.2 Configuring network storage, IBM ESS

You should set up the IBM ESS volumes configuration through the IBM ESS Web user interface as described in this section. To access the URL (host name), a user ID and password are required. The storage administrator usually carries out these steps.

Assigning volumes - IBM ESS Specialist menu

To assign volumes through the IBM ESS Specialist menu, follow these steps:

1. Select **ESS Specialist** → **Storage Allocation** → **Open System Storage™** → **Add Volumes** (Figure 9-8).

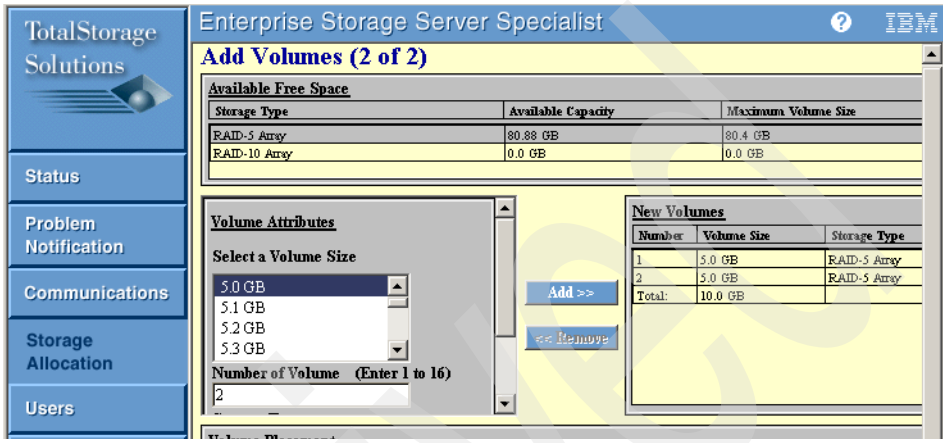


Figure 9-8 Adding Volumes to the system on ESS web

2. Modify IBM ESS volumes assignment (for example, 201-22513, 202-22514) through the Modify Volumes Assignments window (Figure 9-9 on page 258) by selecting **ESS Specialist** → **Storage Allocation** → **Open System Storage** → **Modify Volumes Assignments**.

The source volume 201 and 205 are assigned to the host system. The target volume 204 and 208 are also assigned to the host system for the object restore. The other volumes are not assigned to the host system.

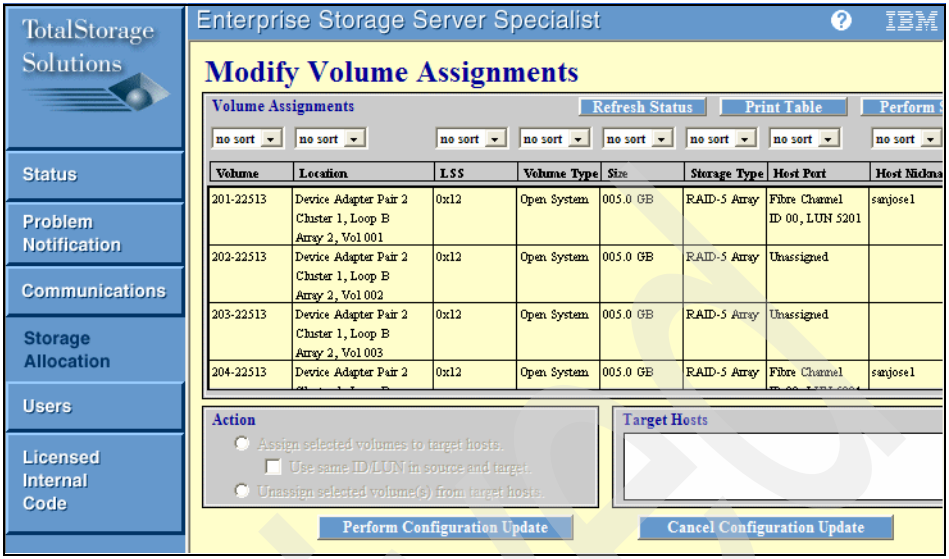


Figure 9-9 Modify Volume Assignment on ESS web

Creating copy task, IBM ESS Copy Services menu

You need to create at least two tasks against a copy pair for backup and restore using the following IBM ESS Web user interface.

Now, three target volumes (three copy pairs) for every source volume are planned. For each source-target pair, two FlashCopy tasks are required — one for backup and one for restore. For the backup method, add one Freeze FlashCopy Consistency Group optional task for each source-target pair. In addition, One Consistency Group task is needed for the ESS Logical Subsystem.

In our lab, we have six source-target pairs. Thus, a total of 19 tasks are created, including 12 FlashCopy tasks, six Freeze FlashCopy Consistency Group tasks, and 1 Consistency Group task.

To create these tasks, select **ESS Copy Services** → **Start with serverX (IP address)** → **Volumes** (as shown in Figure 9-10 on page 259).

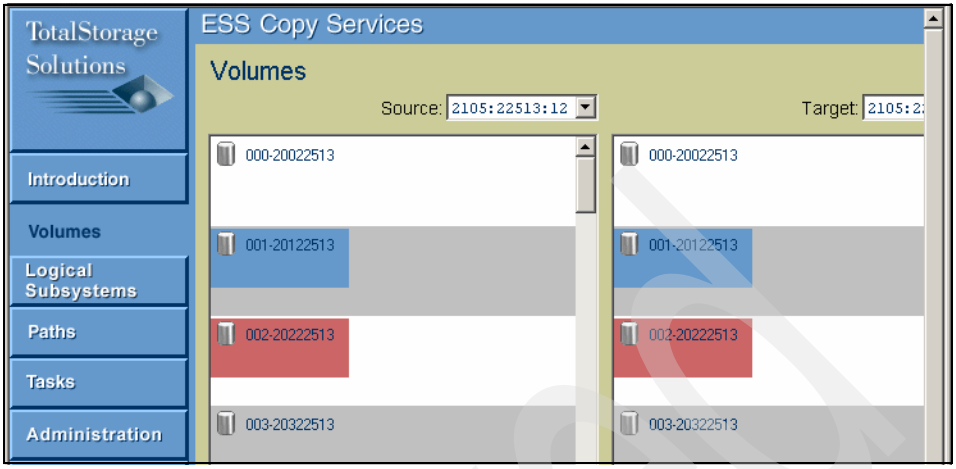


Figure 9-10 Volumes of ESS Copy Services

Creating FlashCopy task

Six FlashCopy tasks are created against three target volumes for backup and restore methods for each source volume. To create the Flashcopy task for copy pairs 201-202, 201-203, 201-204, 205-206, 205-207, and 205-208, follow this procedure:

1. Select Source LSS (Logical Subsystem) ID and Target LSS ID.
2. Left-click the source volume.
3. Right-click the target volume.
4. Right-click the target volume again, the Select task type window opens.
5. Select **Establish FlashCopy pair** and click **Next**.
6. Select nothing in the Select copy options window and click **Next**.
7. Enter zero (0) as the default FC Sequence Number and click **Next**.
8. Enter `FB_201_202_FC` as task name (`201` is the source volume and `202` is the target volume) and description, and click **Save**.
9. Repeat step 2 through 8 using task name *`FB_xxx_yyy_FC`* where *`xxx_yyy`* is the rest of the copy paris. The task name should be unique.

Repeat this procedure to create six tasks for the three target volumes with each source volume. Twelve tasks are created in total.

Creating FlashCopy task with Freeze FlashCopy Consistency Group

FlashCopy task with Freeze FlashCopy Consistency Group is created as follows.

1. Select Source LSS ID and Target LSS ID.
2. Left-click the source volume.
3. Right-click the target volume.
4. Right-click the target volume again, then Select task type window opens.
5. Select **Establish FlashCopy pair** and click **Next**.
6. Select **Freeze FlashCopy Consistency Group** on the Select copy options window and click **Next**.
7. Enter zero (0) as default FC Sequence Number and click **Next**.
8. Enter `FB_201_203_FZFC` as task name and description and click **Save**.

Repeat these steps to create three tasks for the three target volumes for each source volume. Six tasks are created in total.

Creating Consistency Group task

To create the Consistency Group task, select **ESS Copy Services** → **Start with serverX (IP address)** → **Logical Subsystem** (as shown in Figure 9-11).

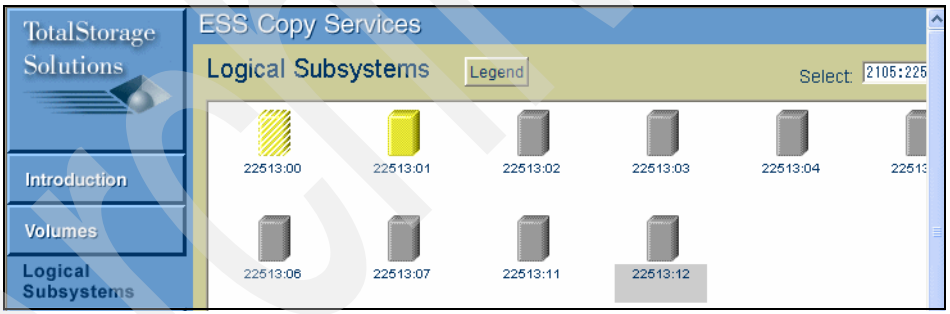


Figure 9-11 ESS Consistency Task creation on Logical Subsystems

Then, follow these steps:

1. Select an ESS in the Select pull-down menu.
2. Left-click the Logical Subsystem icon.
3. Right-click the Logical Subsystem icon.
4. Right-click the target volume again, then Select task type window opens.
5. Select **Consistency Created** on the Select task type window and click **Next**.
6. Select **FlashCopy Consistency Group** on the Select copy options window and click **Next**.

7. Enter `FB_22513_12_CG` as the task name and description, and click **Save**.

You can check if the tasks work well using **Run** on Tasks (Figure 9-12) by selecting **ESS Copy Services** → **Start with serverX (IP address)** → **Tasks**.

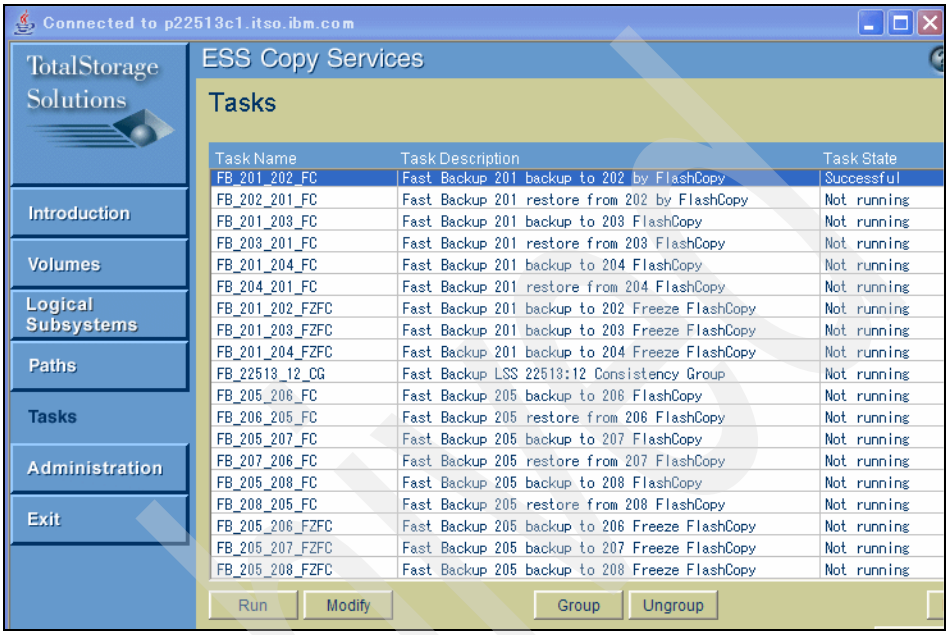


Figure 9-12 ESS Task list

9.3.3 Configuring volumes on the system -AIX

At this point, IBM ESS SDD and ESSCLI should have been installed. The system administrator can now assign and mount the volumes.

Considerations of IBM ESS multiple path I/O

Normally, multiple ports are configured on IBM ESS, and multiple HBAs (Host Bus Adapter) are configured on AIX. Each IBM ESS LUN (logical unit) is configured to be accessed via multi path to each path's HBA for high availability. You can use reList2105s.sh, one IBM ESSCLI tool, to verify the volume path configuration.

Example 9-1 shows multiple copies of IBM ESS volumes on AIX.

Example 9-1 Sample volumes on IBM ESS multiple path I/O

```
root:/> /opt/ibm/ESScli/rsList2105s.sh
disk name       2105 serial number
---------       ------------------
 hdisk1            20122513
 hdisk2            20422513
 hdisk3            20522513
 hdisk4            20822513
 hdisk5            20122513
 hdisk6            20422513
 hdisk7            20522513
 hdisk8            20822513
 hdisk9            20122513
 ...
 vpath0            20122513
 vpath1            20422513
 vpath2            20522513
 vpath3            20622513
root:/>
```

DB2 RE Fast Backup does not support multiple paths for volumes configured on AIX. ESSCLI tools rsList2105s.sh shows IBM ESS volumes attached with volume ID. If you find multiple copies of IBM ESS volumes in your AIX, your SAN switch zoning need to be adjusted based on either port-by-port basis or Worldwide Name connecting devices.

To use Fast Backup function correctly, request your storage administrator to make sure that SAN switch zoning is configured properly so that multiple copies of IBM ESS volumes are not listed by rsList2105s.sh on AIX, as shown in Example 9-2.

Example 9-2 Sample volumes configuration on switch zoning for Fast Backup

```
root:/> /opt/ibm/ESScli/rsList2105s.sh
disk name       2105 serial number
---------       ------------------
 hdisk1            20122513
 hdisk2            20422513
 hdisk3            20522513
 hdisk4            20822513
root:/>
```

Assigning and mounting IBM ESS volumes on the system

You need to define IBM ESS volumes (source volumes) on AIX, and mount it. You can use SMIT on AIX for these tasks as a root user. Follow these steps:

1. Create volume group (for example, `vgess1`).
2. Create logical volume (for example, `lvess1`).
3. Create file system (for example, `/ess1` mount point).
4. Mount file system.
5. Permit all user's access.
6. Confirm if the storage volume is mounted correctly:

   Type the following command on the server as the root or administrator:

   ```
   > <RE_install_dir>/fastbackup/bin/gdbchkst
   ```

   With the correct storage and storage software setting, this command returns the following results:

   ```
   Mount point: "/ess1/"
   Mount type: SAN
   Device information: "lv:lvess1,vg:vgess1"
   Storage server type: ibmess
   Storage server ID: 22513
   Storage server volume ID: 20122513
   ```

Note: The **gdbchkst** shows ESS devices status through ESSCLI command line interface. The location of ESSCLI command line interface is specified in <*RE_install_dir*>/fastbackup/etc/gdbconf. You should verify if the ESSCLI path is correct.

After completing these steps, you have configured IBM ESS volumes for DB2 RE Fast Backup on AIX. Now, you need to prepare Fast Backup administration on Windows and Fast Backup will be ready to go.

Attention: You must not place database logs in the network storage that Fast Backup uses. If database logs are overwritten during restoring a database by Fast Backup, DB2 RE loses the data integrity of backup assets and DB2 RE cannot recover objects correctly.

9.3.4 Installing Fast Backup Server and Client

Fast Backup consists of the client and the server module. The server module is included in the DB2 RE package. Select the custom installation option in the

DB2 RE installation program to install Fast Backup server module. You also need to install Fast Backup client on a Windows client.

To install DB2 RE and Fast Backup Server, you need to log in as a DB2 administrator of the local Windows system or a root user on the UNIX system.

Prerequisite for Fast Backup

Fast Backup server requires administration node definition. For example,

```
db2 catalog admin local node sanjose1 system sanjose1 ostype aix
(note. sanjose1 = hostname)
```

Installing Fast Backup Server

To install the Fast Backup Server, start the DB2 RE InstallShield program and then follow these steps:

1. Select **Custom** option and click **Next**.

   Fast Backup is not selected to install as default on the DB2 RE installation program. To install Fast Backup server, you must select the custom installation the option, as shown in Figure 9-13.

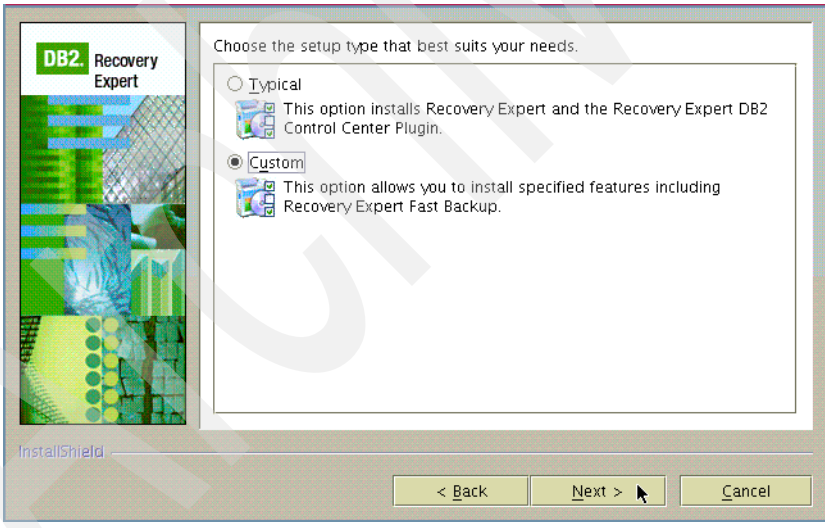


Figure 9-13 Custom for Fast Backup server installation

2. Select **Recovery Expert Fast Backup Server** and click **Next** (Figure 9-14).

   If you have already installed DB2 RE, Fast Backup Server is the only option here.

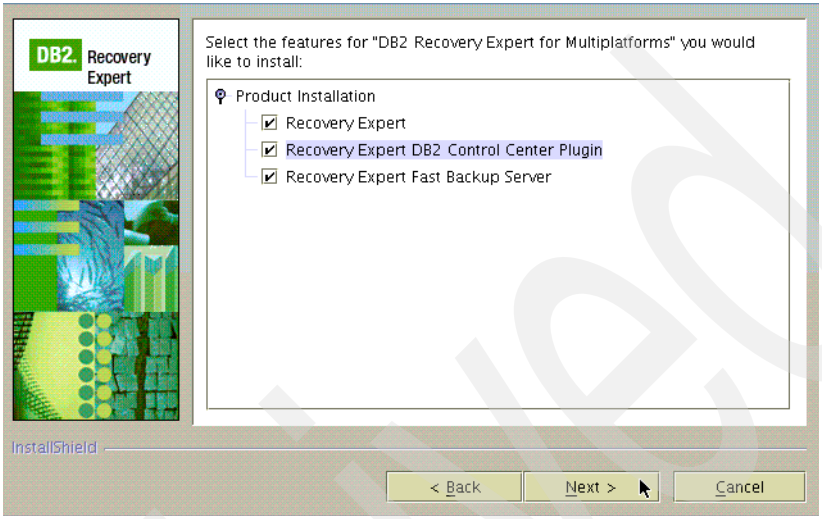


Figure 9-14 Select Fast Backup server

3. Select **Create the FBA Repository** Database and click **Next** (Figure 9-15 on page 266).

   Fast Backup Administration Repository Database holds information that is used by Fast Backup, such as network storage configuration, Fast Backup tasks, specifications, and so forth. The options in this windows are as follows:

   – *Create the database*: If you have no Fast Backup administration repository database.

   – *Recreate the database*: If you already have Fast Backup administration repository database but want to renew all settings of Fast Backup.

   – *Do not create the database*: If you want to create Fast Backup administration repository database using Fast Backup command line interface later.

   – *Use the current database*: If you want to use the existing settings of Fast Backup when you re-install or upgrade Fast Backup.

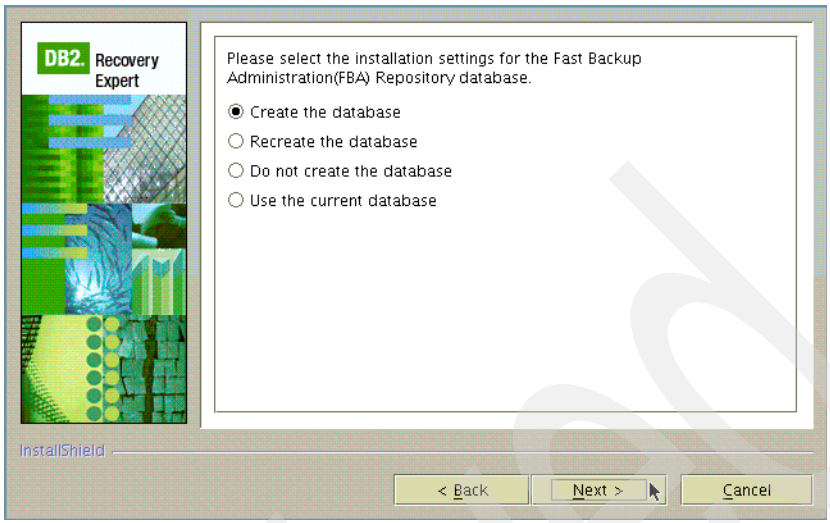


Figure 9-15 Mark Create Fast Backup Administration Repository

4. Enter the Fast Backup administration repository database name, DB2 Instance name, and password, and then click **Next** (Figure 9-16).

   The instance is the local instance in which you want the Fast Backup administration repository database to be created.

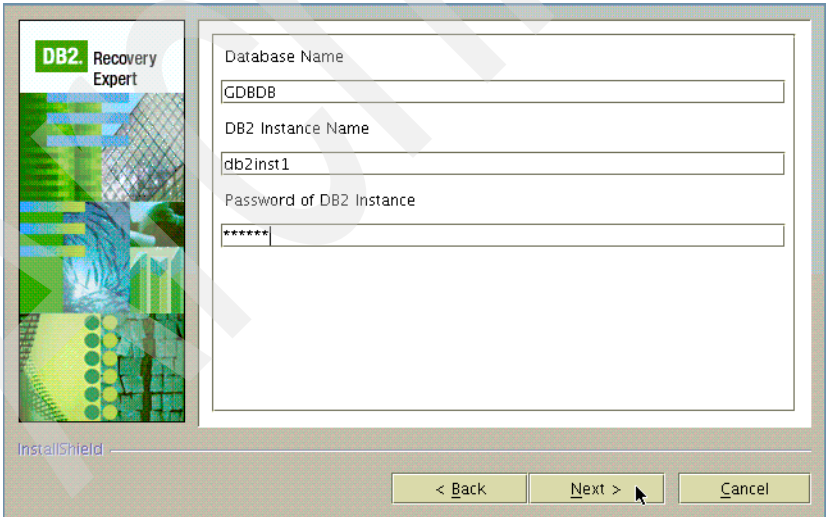


Figure 9-16 Input DB2 Instance name and password

5. Verify the installation contents on the summary window and click **Next** (Figure 9-17), and then the copy operation begins.

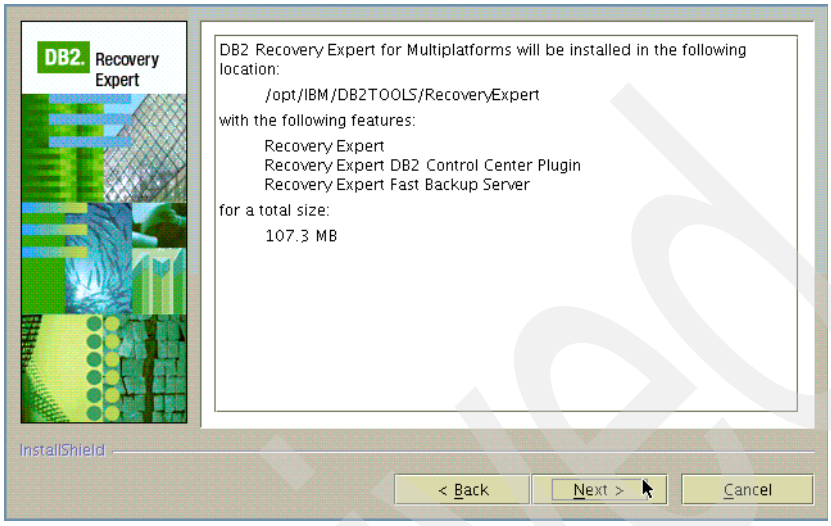


Figure 9-17 DB2 RE installation summary

6. After the copy ends, click **Next** (Figure 9-18), and then click **Finish**.

Figure 9-18 DB2 RE and Fast Backup server installation end

Installing Fast Backup Client

To install the Fast Backup Client, follow the steps:

1. Run Fast Backup Client Installation wizard setup.exe on windows client (Figure 9-19).

Figure 9-19 Fast Backup client installation

2. Enter installation path or use the default directory, and click **Next** (Figure 9-20).

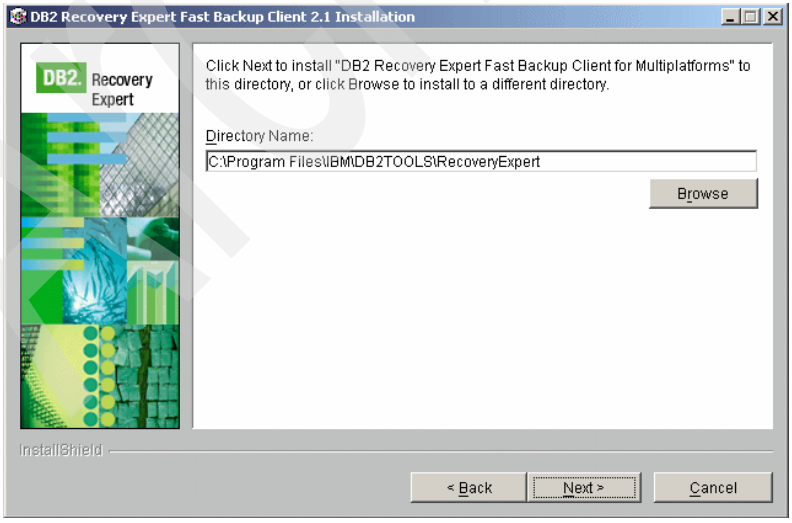


Figure 9-20 Installation Path

3. Verify the installation contents in the summary window, and click **Next** (Figure 9-21).

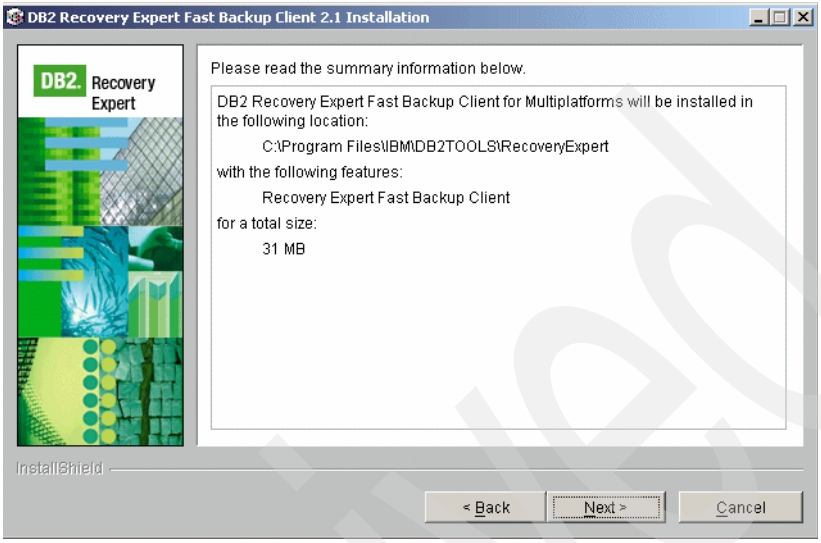


Figure 9-21 Fast Backup client Installation summary

4. Click **Finish** to complete the installation (Figure 9-22).

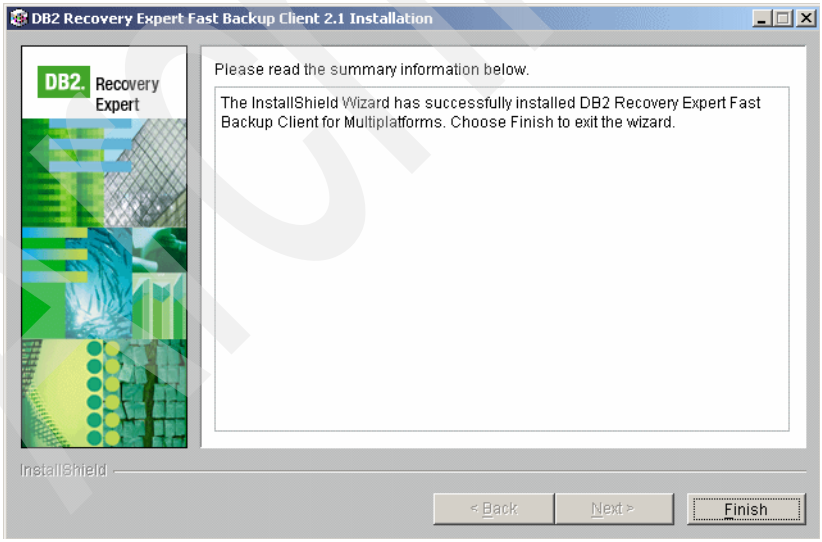


Figure 9-22 Fast Backup client installation end

After Fast Backup installation

You need to verify a couple of tasks after the Fast Backup installation:

- The Fast Backup network configuration file, /RE_install_dir/fastbackup/etc/gdbconf, includes the network storage command line interface (CLI) path definition. Make sure the correct path name is listed in this file. For example, in our lab, we change the configuration file location from this:

  ```
  IBMESSCLI=/usr/opt/ibm/ibm2105cli
  ```

  To this:

  ```
  IBMESSCLI=/opt/ibm/ESScli
  ```

  Note: Remove the comment character (#) in front of the definition line.

- Fast Backup includes many Java stored procedures. You need to change the DB2 database manager configuration parameter JAVA_HEAP_SZ value to 4096 as follows:

  ```
  db2 update dbm cfg using JAVA_HEAP_SZ 4096
  ```

Configuring multiple instance users for Fast Backup

If multiple DB2 instance users will run DB2 RE Fast Backup, you must complete the following configuration steps on each instance except the one which has Fast Backup repository database:

1. Catalog the Fast Backup repository database and the instance in each instance. For example:

   ```
   db2 catalog local node db2inst1 instance db2inst1 system db2server
   db2 catalog database GDBDB as GDBCATXX at node db2inst1
   ```

   Where GDBDB is the Fast Backup repository database under instance db2inst1.

2. Create the property file for Fast Backup for each DB2 instance uses Fast Backup to backup objects:
   - For UNIX and Linux systems, issue the following command:

     ```
     RE_install_dir/fastbackup/ddl/gdbcprop -db
     cataloged_FBA_repository_database
     ```

   - For Windows systems, issue the following command:

     ```
     RE_install_dir/fastbackup/ddl/gdbcprop.cmd -db
     cataloged_FBA_repository_database
     ```

9.3.5 Preparing Fast Backup, Fast Backup Administration

You launch the Fast Backup Administration window from the DB2 Control Center plug-in on Windows client. Fast Backup Administration provides the following services for backup operations:

- Showing and editing backup specification
- Showing backup history
- Creating and editing task settings
- Defining storage server settings
- Defining client setting

Administrating DB2 Recovery Expert Fast Backup

To set up Fast Backup Administration to back up a database or table space, the following tasks are required:

- Defining client setting
- Defining storage server settings
- Creating task settings

The steps to set up Fast Backup Administration are as follows:

1. Launch Fast Backup Administration.

   Run DB2 Control Center on Windows client to catalog system first. Right-click **All Systems** in left side, and then click **Add** to catalog administration TCP/IP node sanjose1, instance TCP/IP node, application database, and Fast Backup Administration Repository database GDBDBSJ1.

   After the instance and database are cataloged, restart the DB2 Control Center and then launch to the Fast Backup Administration window.

   Right-click **SANJOSE1** under All Systems and click **Recovery Expert - Fast Backup Administration...** (Figure 9-23 on page 272).

   Note: If you see the following message, just click **Apply**:

   ```
   Set the Recovery Expert - Fast Backup repository database
   ```

   You might also see the RE Fast Backup Logon window for the connection. If so, just input the instance owner ID and password.

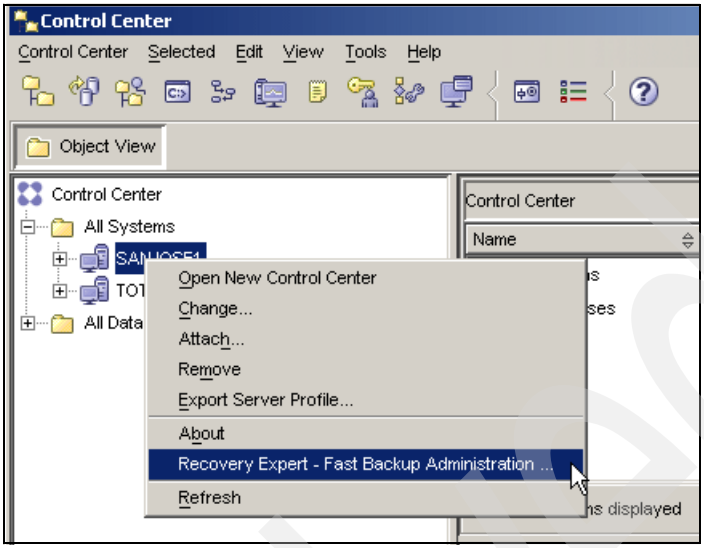


Figure 9-23 Launch Fast Backup Administration

2. Define the client setting.

   Click the Client Setting tab to enter the Fast Backup Administration Repository database alias name, as shown in Figure 9-24.

   Input the correct alias name, such as `GDBDBSJ1` and click **Apply**. You might be required to input an instance owner ID and password also.

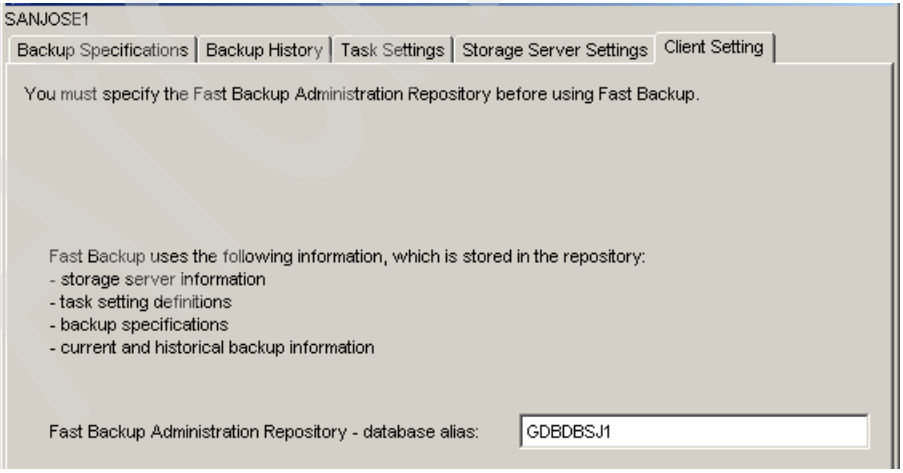


Figure 9-24 Client Setting on Fast Backup Administration

3. Define storage server settings.

   DB2 RE Fast Backup communicates with the network storage system through its command line interface to invoke hardware level volume copy. On this window, the parameters which are necessary for the CLI are defined. To define storage server settings:

   a. Click Storage Server Settings tab.

   b. Select the storage server to be used in left-side list, see Figure 9-25 on page 274.

      Fast Backup determines and lists the server ID, server types, and mount type.

   c. Enter the **h**ost name, user ID, and password.

      The Host name points to IBM ESS administration Web interface which can be accessed from the host system. User ID and Password are for this interface.

   d. Click **Apply**.

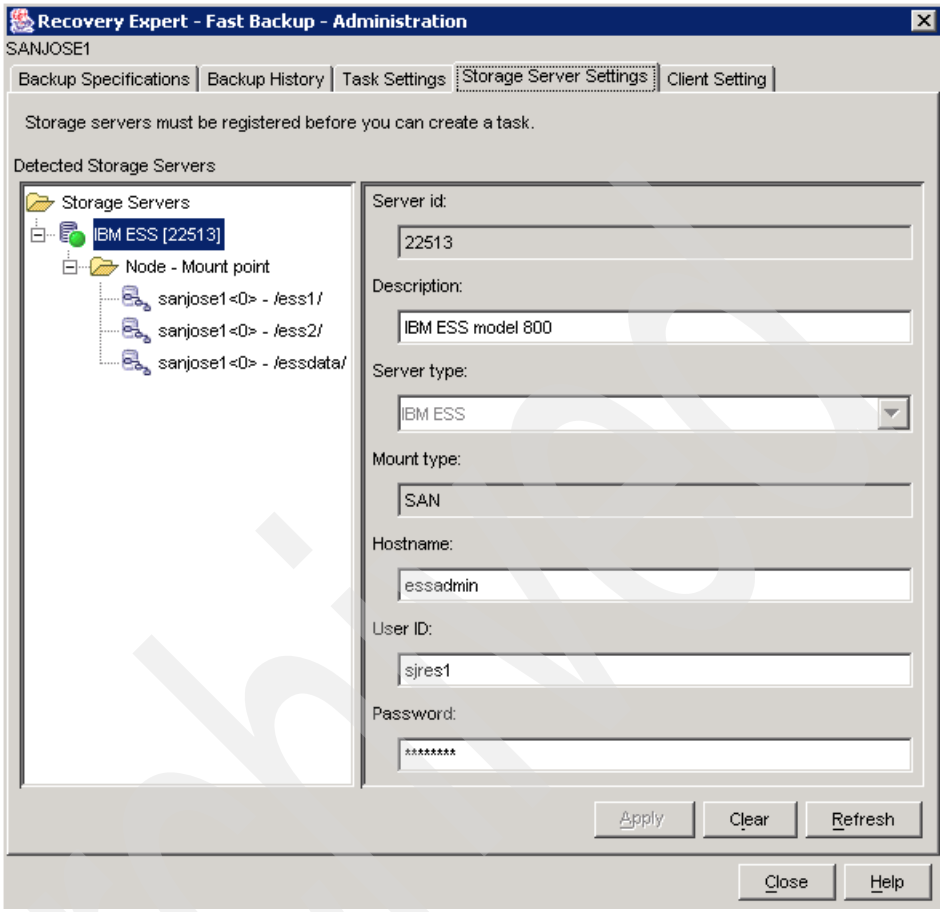


Figure 9-25 Storage Server settings on Fast Backup Administration

4. Create task settings.

   The ESS copy tasks have already been created through ESS administration Web interface. Fast Backup requests ESS to do FlashCopy by invoking those ESS copy tasks. In the Fast Backup Task Settings, the ESS copy tasks to be used by Fast Backup on backup and restore are defined.

   Table 9-2 on page 275 lists the Fast Backup tasks for scenarios in this chapter. In Scenario A, a table space on /ess2 is backed up and restored. In the Scenario B, a database on /ess1 and /ess2 is backed up and restored.

Table 9-2 Fast Backup tasks table for some scenarios

FB task name	src mnt, copy pair	backup method	restore method
BACKUP_TS_207	/ess2, 205 - 207,	flashcopy	flashcopy
BACKUP_DB_202	/ess1, 201 - 202	flashcopy-freeze	flashcopy
BACKUP_DB_206	/ess2, 205 - 206	flashcopy-freeze	flashcopy

To create task settings:

a. Click **Task Settings**.
b. Click **Create Task**.
c. Click **Create Volume Mapping**.
d. Select a volume of IBM ESS in left side.

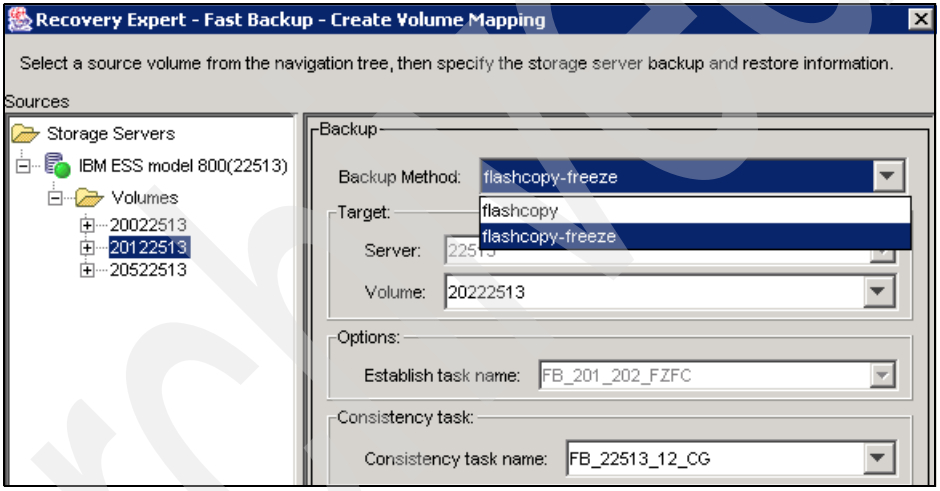


Figure 9-26 Backup settings of Task Settings on Fast Backup Administration

e. Select **flashcopy** or **flashcopy-freeze** as the Backup Method.

   The list of Backup Method, Target volume, and Establish task name fields lists what you have defined in ESS administration Web page, see Table 9-1 on page 256.

f. Select or make sure Target Volume and Establish task name are what you want.

g. Select or make sure that `FB_22513_12_CG` is the Consistency task name if the selected Backup Method is flashcopy-freeze.

h. Select flashcopy as the Restore Method.

   The volume restore does not work if other than flashcopy task is selected in the restore process. If the target volume is assigned to the host system on ESS administration web page, ObjectRestore is selectable as Restore Method. An object is restored by copy operation on restoring process. If *none* is selected as Restore Method, all restore process should be handled by User Exit without DB2 RE and Fast Backup support.

i. Select **Source Volume** and **Establish task name**.

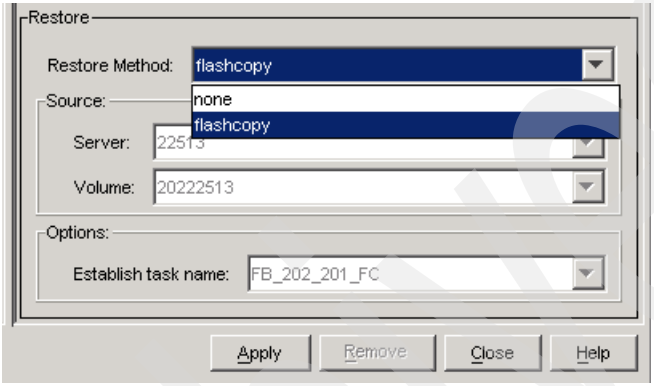


Figure 9-27 Restore settings of Task Settings on Fast Backup Administration

j. Click **Apply** and **Close**.

k. Task screen comes up again. Enter **Task ID** and **Description**.

l. Click **OK**.

m. Repeat steps b on page 275 through l and create three tasks in total, as shown in Example 9-28.

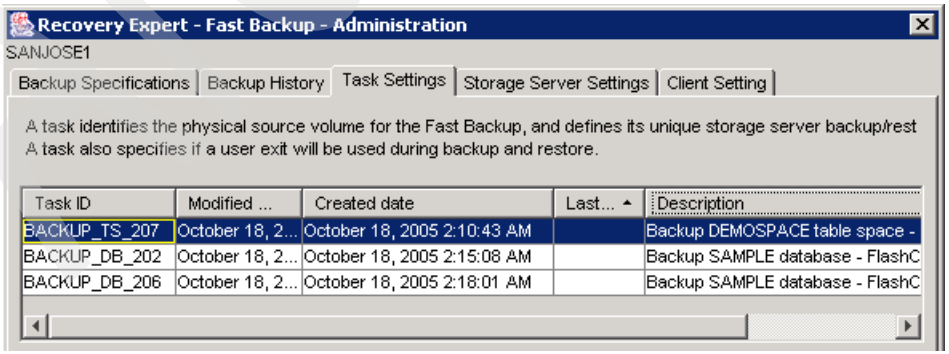


Figure 9-28 Task list of Task Settings on Fast Backup Administration

Now, Fast Backup is ready for backing up and restoring a database or a table space.

9.4 Scenario A: Recovering a table space

This scenario demonstrates how to recover deleted data by restoring the table space using Fast Backup table space backup copy and the DB2 RE GUI. The primary merit of Fast Backup is its high performance backup and restore capability on the large scale database or table space. This scenario might not be enough of a challenge in this aspect, but it easily shows the Fast Backup procedures.

9.4.1 Scenario description

ITSO Co. is a small company that is driving a CRM system on several AIX servers. They save DB2 backups to the tape media, but they need to reduce their recovery time to respond to customer requirements. They have decided to install an IBM ESS storage system to have a high availability system.

You are the DBA, and you make a trial plan to prepare some ESS volumes to move an active table space to ESS volume and to use Fast Backup. You talk to your storage administrator and system administrator about system configuration and ESS volume assignment for Fast Backup environment. The first trial is to move a database and a table space to two ESS volumes. Then, you will try a table space backup and restore first and a database backup and restore later through Fast Backup.

The details are of this plan are:

- IBM ESS volumes for a database: 2 (one source volume, one target volume)
- IBM ESS volumes for a table space: 2 (one source volume, one target volume)
- IBM ESS copy task: FlashCopy tasks
- System name: *SANJOSE1*
- Mount point: database on /ess1, table space on /ess2
- Database name: *SAMPLE*
- Table space name: *DEMOSPACE*

The typical usage of a database or table space recovery is a point-in-time recovery such as recovering an object caused by an application.

In this scenario, the assumption is that a user has deleted data from the DEPARTMENT table accidentally. The user requests that you to recover DEMOSPACE table space to the time before the wrong transaction was run. The mistaken delete statement is as follows:

```
db2 “delete from department where admrdept='A00'”
```

9.4.2 Prerequisites for the scenario

This scenario uses DB2 UDB SAMPLE database. It uses all of the defaults for database structure and configuration except the following:

- SAMPLE database is created on /ess1. NEWLOGPATH must be located in the volume other then the backup source volume. This is a DB2 RE Fast Backup requirement.
- LOGRETAIN must be set to RECOVERY. This is a DB2 RE requirement.
- DEMOSPACE table space is created on an IBM ESS volume /ess2 using makedms.sh. Tables are created using demospace.ddl. Data are imported using demospace.sh.
- The Versioning Repository must have been created and updated once the last DDL was run. This is a DB2 RE requirement.
- A backup must have been taken after the last Versioning Repository update. This is a DB2 RE requirement.
- The operation flow is as follows.

  a. Run the following:

  ```
  db2sampl /ess1
  ```

  b. Set NEWLOGPATH, For example

  ```
  db2 update db cfg for sample using newlogpath /home/db2inst1/logs/sample
  ```

  c. Run the following:

  ```
  makedms.sh
  demospace.sh
  ```

  d. Create the Versioning Repository on DB2 RE.

  e. Set LOGRETAIN recovery to SAMPLE database.

  f. Backup the SAMPLE database.

  g. Launch the Fast Backup client for a table space backup.

Note: If the URID is selected for the DB2 RE Fast Backup point-in-time recovery, you *must* set the Data Capture Changes (DCC) option to `on` for tables to be backed up using Fast Backup. This is also required in order to support SQL generation during point-in-time recovery or by the Log Analysis tool. For example:

```
db2 alter table DEMOTABLE data capture changes
```

You can find the scripts that we used to create table spaces and tables in Appendix A, “Sample applications” on page 357.

9.4.3 Backing up a table space through the Fast Backup client GUI

The backup process of Fast Backup is launched through Fast Backup client interface, which is a DB2 Control Center plug-in. This section describes the operations that are needed to take a backup of the table space using Fast Backup.

Launch Fast Backup client for a table space backup

On the DB2 Control Center, list the table space to be backed up and launch Fast Backup client by following these steps:

1. Click and expand the SAMPLE database tree in the left side.
2. Select **Table Spaces** and table spaces are listed in the right-side pane.
3. Right-click **DEMOSPACE** and select **Recovery Expert - Fast Backup**, as shown in Figure 9-29 on page 280.

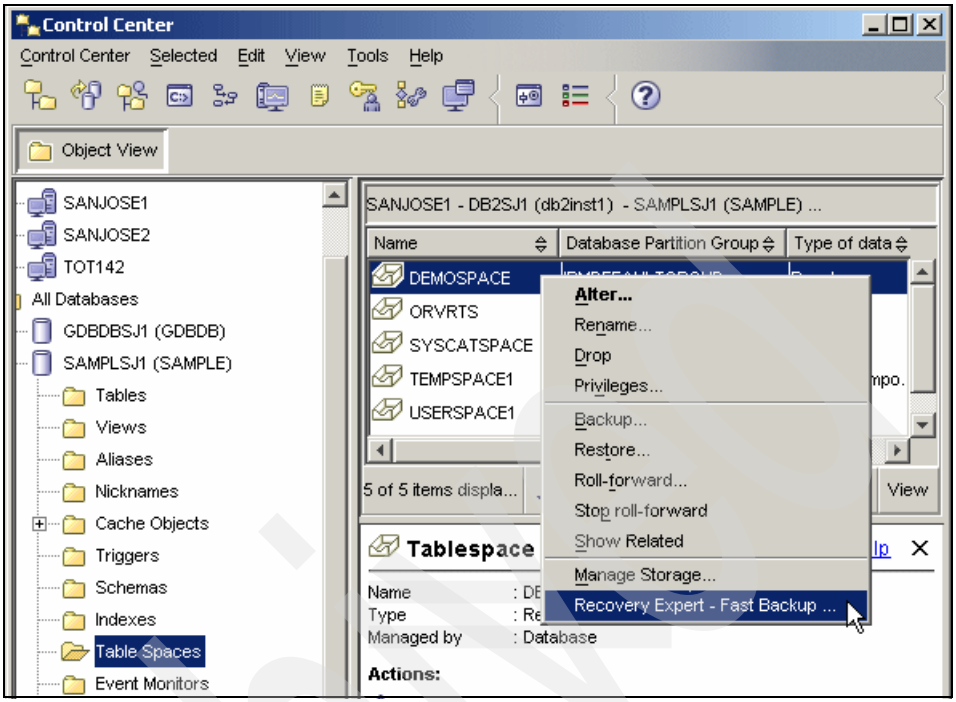


Figure 9-29 Launch Fast Backup client on DB2 Control Center

Click Backup now on Fast Backup client

Fast Backup client window shows the volumes to be backed up and tasks to be used for the table space backup. Look at the Category field of Table spaces List. If multiple tasks are available for this backup, an `Insufficient` message is indicated in Category, and none of the tasks are selected. In this case, mark the task for this backup and the *Insufficient* message changes to `Ready`.

Continue with these steps:

1. Select **BACKUP_TS_207** in Available Tasks.
2. Click **Backup Now** to start Fast Backup to back up the table space, as shown in Figure 9-30 on page 281.

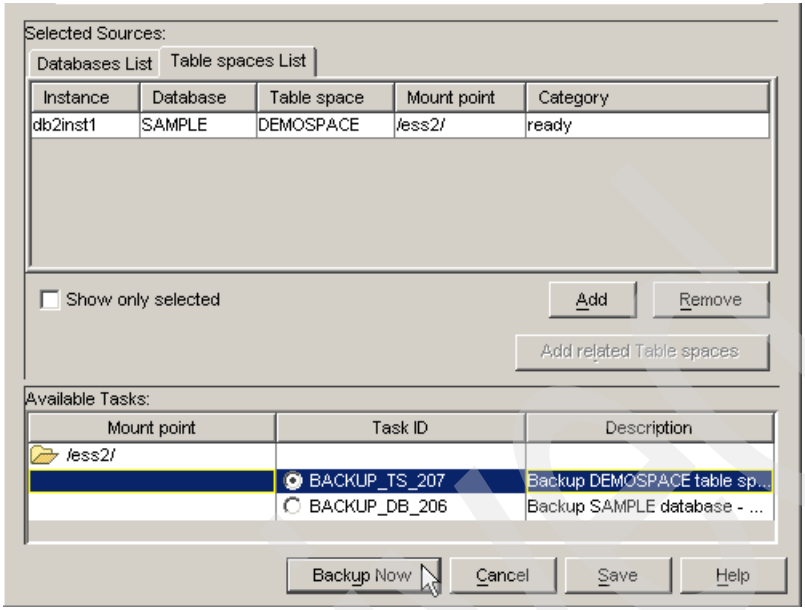


Figure 9-30 Do Backup Now on Fast Backup client

Now, Fast Backup table space `DEMOSPACE` backup is available in target volume ID 207, as shown in Figure 9-31.

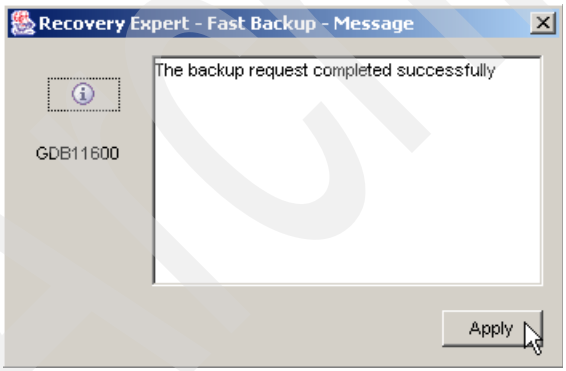


Figure 9-31 Fast Backup backup completion message

9.4.4 Recovering a table space to recover deleted data from a table

Now, you have a Fast Backup backup in a target volume for the table space restore. Run the following wrong transaction for this scenario.

```
db2 connect to sample
db2 “delete from department where admrdept='A00'”
db2 connect reset
```

Example 9-3 shows the current DEPARTMENT table contents. You can see that A00 in ADMRDEPT rows have been deleted.

Example 9-3 Deleted data in DEPARTMENT table on SAMPLETS table space

```
db2inst1:/home/db2inst1> db2 "select * from department"

DEPTNO DEPTNAME                    MGRNO  ADMRDEPT LOCATION
------ ---------------------------- ------ -------- ----------------
D11    MANUFACTURING SYSTEMS       000060 D01      -
D21    ADMINISTRATION SYSTEMS      000070 D01      -
E11    OPERATIONS                  000090 E01      -
E21    SOFTWARE SUPPORT            000100 E01      -
```

Launch DB2 Recovery Expert

After running the wrong transaction, go to the next step, which is point-in-time recovery for a table space. Run DB2 RE and recover the DEMOSPACE table space. Follow these steps:

1. Run DB2 RE.
2. Select the SAMPLE database and connect to it (Figure 9-32).

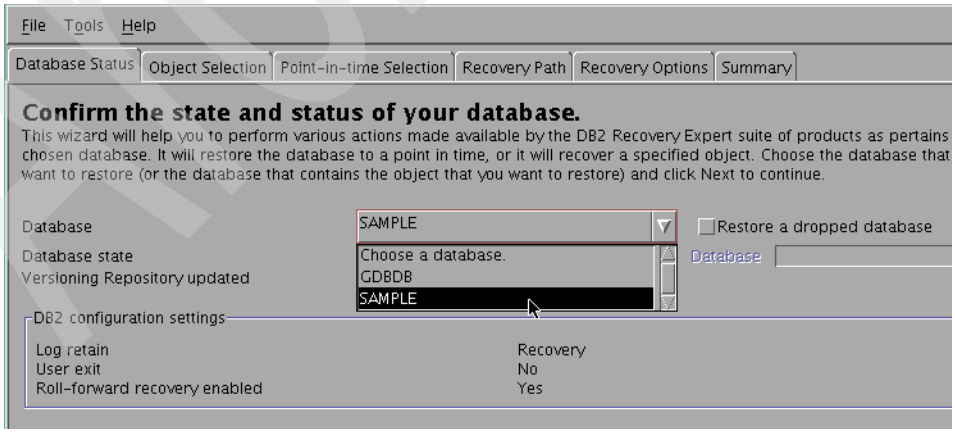


Figure 9-32 Select SAMPLESPACE table space

Select a table space

DB2 RE provides the object retrieving function. You can select the proper table space from the listed table spaces (Figure 9-34) as follows:

1. Go to Object Selection.
2. Select **Table Space** and click the more icon ..., Figure 9-33.

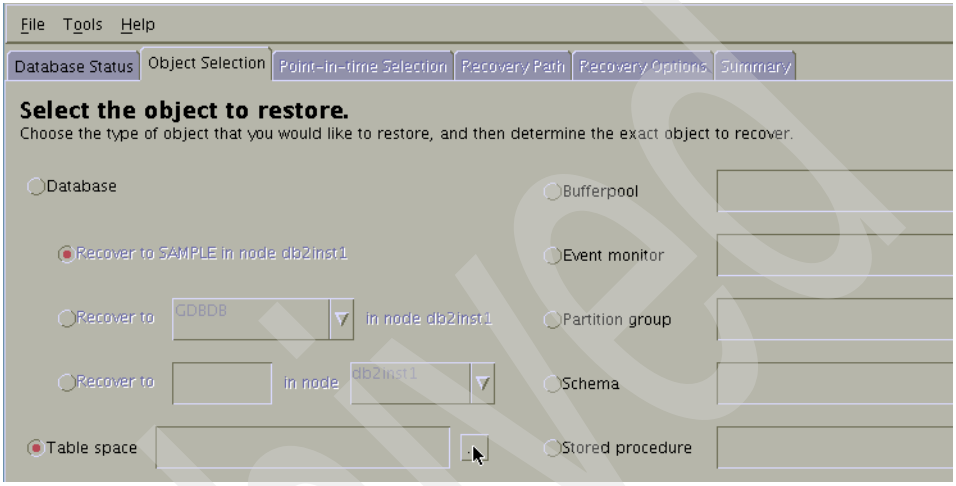


Figure 9-33 DB2 RE Object Selection - Table space

3. The Object selection window opens (Figure 9-34). Click **Retrieve** and select DEMOSPACE table space from the listed table spaces.

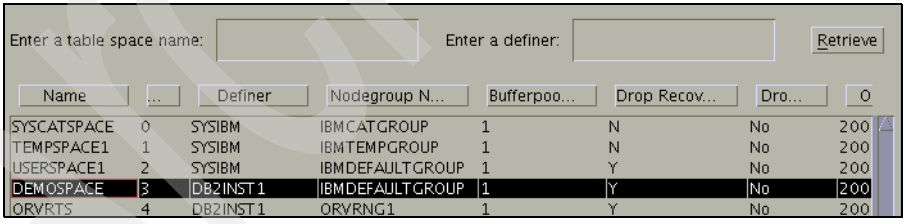


Figure 9-34 Select SAMPLESPACE table space

Select point-in-time for recovering a table space

To recover the DEMOSPACE table space, specify point-in-time before running the delete from department transaction. Follow these steps:

1. Go to Point-in-time Selection and click **Advance**
2. Select **Backup images** and select **Fast Backup Media Type (**Figure 9-35 on page 284).

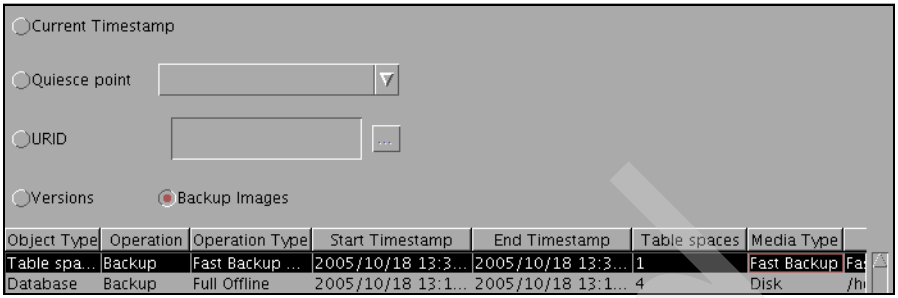


Figure 9-35 Select Fast Backup Media Type for Fast Backup Point-in-time

Select Fast Backup as a recovery path

To recovery a table object using the Fast Backup backup image, the Fast Backup backup image needs to be selected as the recovery path on the Db2 Re.

3. Go to *Recovery Path*. Click **OK.** If "`The Active Log Inventory Data may not be up-to-date`." message appears, just click OK.
4. Select **Restore from a Fast Backup image** on *Recovery Path* for Fast Backup restore scenario. See Figure 9-36.

Figure 9-36 Select Fast Backup as Recovery Path for this scenario

Do Run Now

Run the Fast Backup table space restore process to recover a table.

1. Go to Summary and click **Run Now**.

   If the following message displays, click **OK**.

   ```
   Are you are sure you want to run now?
   ```

   A processing complete message appears and the DEMOSPACE table space has been recovered (Figure 9-37).

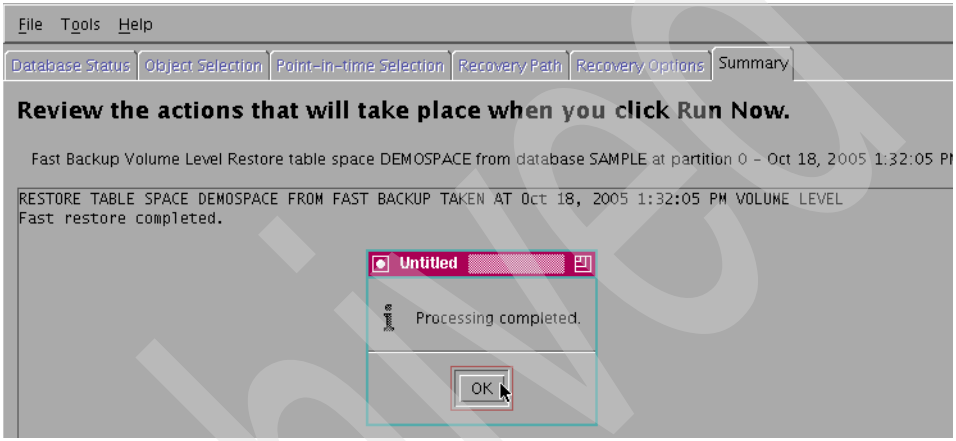


Figure 9-37 Go Run Now and its result

Example 9-4 shows the recovered DEPARTMENT table contents. You can see that A00 under ADMRDEPT rows are restored.

Example 9-4 Recovered data in DEPARTMENT table on SAMPLESPACE table space

```
db2inst1:/home/db2inst1> db2 "select * from department"

DEPTNO DEPTNAME                    MGRNO  ADMRDEPT LOCATION
------ ---------------------------- ------ -------- ----------------
A00    SPIFFY COMPUTER SERVICE DIV.  000010 A00      -
B01    PLANNING                     000020 A00      -
C01    INFORMATION CENTER           000030 A00      -
D01    DEVELOPMENT CENTER           -      A00      -
D11    MANUFACTURING SYSTEMS        000060 D01      -
D21    ADMINISTRATION SYSTEMS       000070 D01      -
E01    SUPPORT SERVICES             000050 A00      -
E11    OPERATIONS                   000090 E01      -
E21    SOFTWARE SUPPORT             000100 E01      -

  9 record(s) selected.
```

9.5 Scenario B: Recovering a database on the multiple network storage volumes

This scenario demonstrates how to recover a dropped table by restoring the database on the multiple network storage volumes using both the Fast Backup backup command line interface and the DB2 RE GUI. You can put the backup command on a scheduler such as **cron** on UNIX or **at** on Windows for the periodic backup.

A table space Fast Backup backup has been made on the previous scenario and now a database Fast Backup is made on different target volume. So, multiple backups are available against a source database.

9.5.1 Scenario description

In this scenario, a table was dropped by mistake, and we want to recover the SAMPLE database to point-in-time before the table PROJECT was dropped. To restore the SAMPLE database, a Fast Backup database copy before the table was dropped should be available.

9.5.2 Managing Fast Backup Administration

Fast Backup Administration provides several services for backup operation of Fast Backup:

- Showing and editing backup specifications
- Showing backup history
- Creating and editing task settings
- Defining storage server settings
- Defining client setting

Backup history

To view the backup history, follow these steps:

1. Right-click **SANJOSE1** under All Systems on Fast Backup Windows client, and click **Recovery Expert - Fast Backup Administration...** (Figure 9-23 on page 272).
2. Go to the Backup History tab**.** Backup History shows the list of Fast Backup backups (Figure 9-38 on page 287).

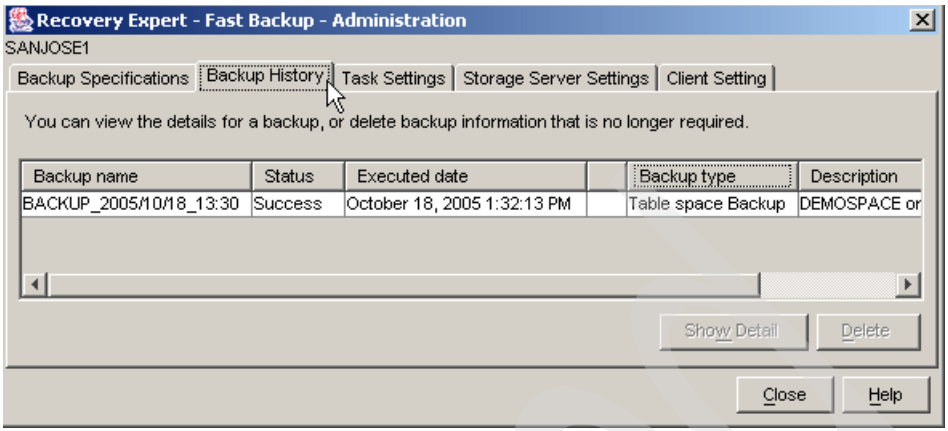


Figure 9-38 Fast Backup Specification creation

3. Select a backup and click **Show Detail** for detail information, including database task and mount point specific information.

Task settings

Through the Task Settings window, you can modify the created tasks (Figure 9-39). If you select **Volume Map**, the volume level modifications are available. In the Task Details dialog, you can add User Exits to the created tasks. There are two types of User Exits, one is invoked on backing up and the other is invoked on restoring a database or a table space.

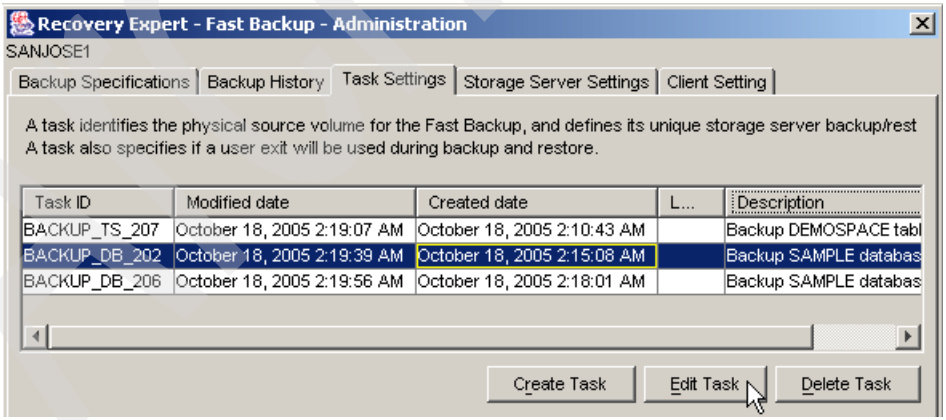


Figure 9-39 Task selection on Task settings

Fast Backup User Exits

Fast Backup provides a User Exits interface (Figure 9-40) that you can use to invoke a storage manager (such as Tivoli Storage Manager) to archive a volume backup to tape and to retrieve a volume backup back to disk so that it can be used as a recovery asset during a DB2 RE database or table space recovery.

All parameters are defined in Fast Backup Specifications in Fast Backup Administration Repository Database. In the backup process, these parameters are passed to Backup User Exits at the end of the Fast Backup backup process. In the restore process, they are passed to Restore User Exits at the beginning of the Fast Backup restore process.

Information such as source and target volume ID and database file names are passed through stdin by Fast Backup. You can check the information from User Exits through stdout in the Fast Backup Administration History window.

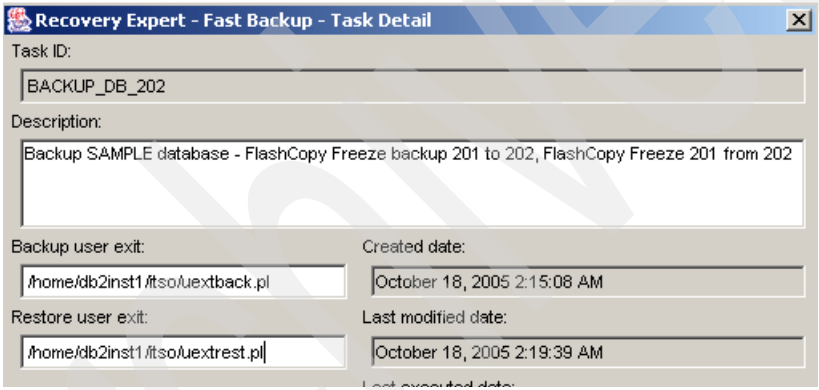


Figure 9-40 Edit Tasks for User Exit

Note: Although you modify a task that is defined in an existing Fast Backup specification, Fast Backup does not associate the modified task with that specification. You need to create a new Fast Backup specification for the modified task.

Fast Backup supports the following types of User Exits:

- Backup User Exit

  Backup User Exit is invoked when Fast Backup completes its back up of a database or a table space. Example 9-5 shows a sample User Exit. The information such as backup ID, target volume ID, and backed up files name (see Example 9-7 on page 290) that are passed by Fast Backup are used to make the archive server (such Tivoli Storage Manager server) request. The archive server that receives the request attaches the target volume and saves its image to archive destination.

Example 9-5 Sample of Backup User Exit

```
#!/usr/bin/perl

$fname = '/home/db2inst1/itso/userexit.out';
open (OUT, ">>$fname") or die "$!";

$i=<STDIN>;
print OUT "-backup----------------------------\n";
print OUT $i."\n";
print "backup userexit run successfully!!!"."\n";
```

- Restore User Exit

  Restore User Exit is invoked prior to the restore operation. Information such as the backup ID, target volume ID, and backed up file names that are passed by Fast Backup (see Example 9-7 on page 290) are passed to the archive server. The archive server brings the archived image back to the target volume for restoring.

Example 9-6 Sample of Restore User Exit

```
#!/usr/bin/perl

$fname = '/home/db2inst1/itso/userexit.out';
open (OUT, ">>$fname") or die "$!";

$i=<STDIN>;
print OUT "-restore----------------------------\n";
print OUT $i."\n";
print "restore userexit run successfully!!!"."\n";
```

Example 9-7 shows the result of the Backup and Restore User Exits running and receiving from Fast Backup on backing up and restoring a table space and a database.

Example 9-7 Sample of User Exits parameters from Fast Backup

```
-backup-----------------------------
 -id BACKUP_SAMPLE_DB -task "BACKUP_DB_202" -type ibmess -method
flashcopy-freeze -srcssid "22513" -srcvol "20122513" -destssid "22513" -destvol
"20222513" -o
"establishtaskname:FB_201_202_FZFC,consistencytaskname:FB_22513_12_CG" -rnode 0
-path "/ess1/db2inst1/NODE0000/SQL00001/" -path
"/ess1/db2inst1/NODE0000/SQL00001/ORVRTS" -path
"/ess1/db2inst1/NODE0000/SQL00001/SQLT0000.0" -path
"/ess1/db2inst1/NODE0000/SQL00001/SQLT0001.0" -path
"/ess1/db2inst1/NODE0000/SQL00001/SQLT0002.0" -path
"/ess1/db2inst1/NODE0000/SQL00001/SYSTOOLSPACE"

-restore----------------------------
 -id BACKUP_SAMPLE_DB -task "BACKUP_DB_202" -type ibmess -method flashcopy
-srcssid "22513" -srcvol "20222513" -destssid "22513" -destvol "20122513" -o
"establishtaskname:FB_202_201_FC" -rnode 0 -path
"/ess1/db2inst1/NODE0000/SQL00001/" -path
"/ess1/db2inst1/NODE0000/SQL00001/ORVRTS" -path
"/ess1/db2inst1/NODE0000/SQL00001/SQLT0000.0" -path
"/ess1/db2inst1/NODE0000/SQL00001/SQLT0001.0" -path
"/ess1/db2inst1/NODE0000/SQL00001/SQLT0002.0" -path
"/ess1/db2inst1/NODE0000/SQL00001/SYSTOOLSPACE" -archiveinfo "backup userexit
run successfully!!!\n"
```

Launching Fast Backup client for a database backup

Before the SAMPLE database is backed up through the Fast Backup command line interface, you need to create Fast Backup specification from the Fast Backup client. Follow these steps:

1. From the Control Center, click and expand the instance (for example, SANJOSE1).
2. Right-click the SAMPLE database, see Figure 9-41 on page 291.
3. Select **Recovery Expert - Fast Backup**.

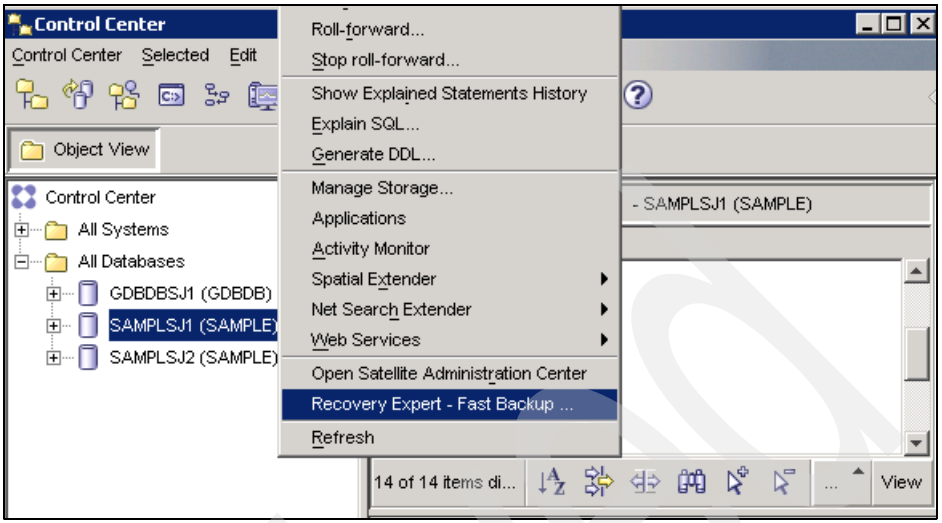


Figure 9-41 Launch Fast Backup client for a database backup

Saving the Fast Backup Specification

Fast Backup checks the location of the database, the table spaces, and the tasks for the source and target volume pairs to be used for the backup. Fast Backup shows all selectable tasks for this backup. To save the Fast Backup Specification, follow these steps:

1. Select **BACKUP_DB_206** in Available Tasks and make sure that Category is in the *ready* state on the Database List tab.
2. Click **Save** on the Fast Backup client window (Figure 9-42 on page 292) to open the Save Backup Specification window (see Figure 9-43 on page 292). This specification is necessary for Fast Backup command line interface.

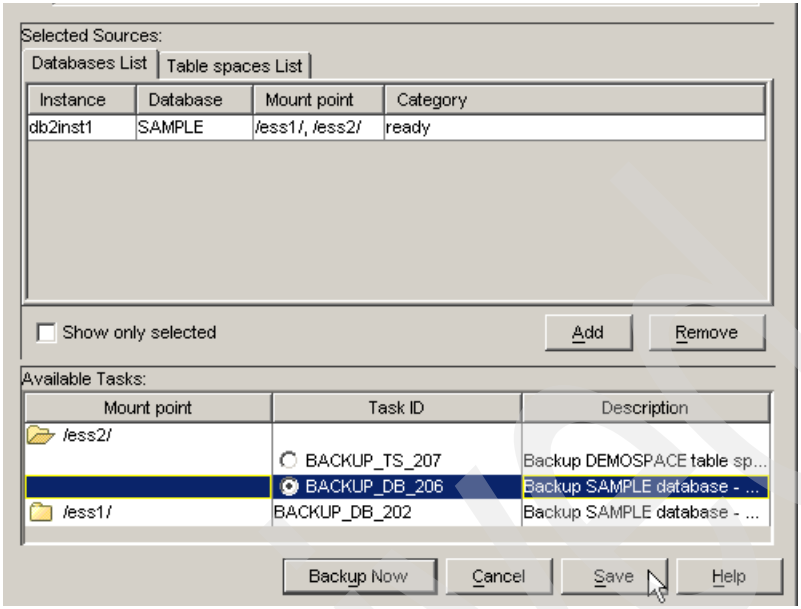


Figure 9-42 Fast Backup Specification creation

3. Click **Save** to save the specification to Fast Backup Administration repository (Figure 9-43).

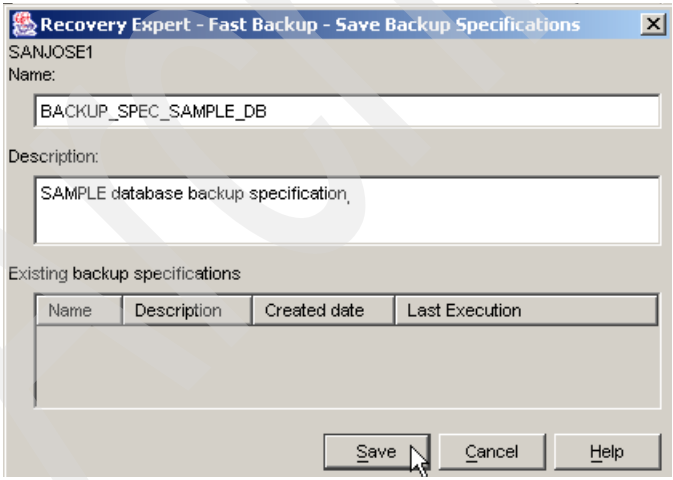


Figure 9-43 Save Fast Backup Specification

9.5.3 Using the Fast Backup command line interface

You can also launch the backup process of Fast Backup through Fast Backup command line interface, which is available on the Windows client and the host system. The Fast Backup command line interface requires the Java JRE setting and DB2 RE Fast Backup stored procedure environment. Confirm the Java JRE path and JAVA_HOME environment variable are set properly.

Fast Backup command line interface

Fast Backup provides the command line interfaces that is discussed in this section. These command line interfaces provide several parameters. For details about these parameters, refer to *IBM DB2 Recovery Expert for Multiplatforms User's Guide*, SC18-9564. Fast Backup provides the following command line interfaces:

- **gdbtask backup**

  The **gdbtask backup** command invokes Fast Backup backup process. This command is useful for scheduled backup operation. It requires Fast Backup specification. The specification might need to be re-created if table spaces composition in a database is changed.

- **gdbtask list**

  The **gdbtask list** command shows historical lists of backups, restores, and specifications, as shown in Example 9-8.

Example 9-8 A Fast Backup command line interface - gdbtask

```
C:\>set PATH=C:\Program Files\IBM\DB2TOOLS\RecoveryExpert\fastbackup\bin;%PATH%

C:\>path
PATH=C:\Program
Files\IBM\DB2TOOLS\RecoveryExpert\fastbackup\bin;C:\WINDOWS\system32;C:\WINDOWS
;C:\WINDOWS\System32\Wbem;C:\PROGRA~1\IBM\SQLLIB\BIN;C:\PROGRA~1\IBM\SQLLIB\FUN
CTION;C:\PROGRA~1\IBM\SQLLIB\SAMPLES\REPL

C:\>gdbtask.cmd list backup -db2system sanjose1 -user db2inst1 using aix123
BACKUP_2005/10/18_13:30 2005/10/18 13:32:13     Success DEMOSPACE on /ess2
table space backup

C:\>gdbtask.cmd list restore -db2system sanjose1 -user db2inst1 using aix123
0       BACKUP_2005/10/18_13:30 2005/10/18 13:44:42     Success

C:\>gdbtask.cmd list setting -db2system sanjose1 -user db2inst1 using aix123
BACKUP_SPEC_SAMPLE_DB   2005/10/18 14:30:34     SAMPLE database backup
specification
```

- **gdbchange**

  The **gdbchange** command is used to change the user ID and password for Fast Backup Administration repository database.

- **gdbchkst**

  The **gdbchkst** commands shows the network storage status. This is helpful to check their configuration and definition. Refer to “Assigning and mounting IBM ESS volumes on the system” on page 263.

Note: If the DB2SYSTEM environment variable is not set, the -db2system xxxx and -user xxxx using xxxx parameters might be required for each command line interface when they are invoked from the remote client node or on the multiple partition database node. See Example 9-8.

Scheduling Fast Backup

Using the Fast Backup command line interface, you can schedule a backup on the system using scheduling tools such as at on windows or **cron** on UNIX.

Note: The **-o replace** (replace BACKUP name) option might be useful when scheduling Fast Backup on a Windows client. On UNIX, you might use ‘date +%y%m%d’ for the BACKUP name instead. See Example 9-9.

Example 9-9 How to use the Windows scheduling tool at

```
C:\> type c:\backup.bat
C:\Program Files\IBM\DB2TOOLS\RecoveryExpert\fastbackup\bin\gdbtask.cmd backup
-name BACKUP_SAMPLE_SU_0300 -o replace -setting BACKUP_SAMPLE -db2system
sanjose1 -user db2inst1 using aix123

C:\> at 3:00 /every:su c:\backup.bat

C:\> at
Status ID   Day                       Time          Command Line
-------------------------------------------------------------------------------
       1    Each Su                   3:00 AM       C:\backup.bat
C:\> at 1 /delete

C:\> at
There are no entries in the list.
```

Running the Fast Backup command line interface

You invoke Fast Backup database backup by the **gdbtask backup** command line interface, as shown in Example 9-10.

Example 9-10 A Fast Backup backup command line interface - gdbtask

```
C:\>set PATH=C:\Program Files\IBM\DB2TOOLS\RecoveryExpert\fastbackup\bin;%PATH%

C:\>gdbtask.cmd list backup -db2system sanjose1 -user db2inst1 using aix123
BACKUP_2005/10/18_13:30 2005/10/18 13:32:13     Success DEMOSPACE on /ess2
table space backup

C:\>gdbtask.cmd list setting -db2system sanjose1 -user db2inst1 using aix123
BACKUP_SPEC_SAMPLE_DB   2005/10/18 14:30:34     SAMPLE database backup
specification

C:\>gdbtask.cmd backup -name BACKUP_SAMPLE_DB -setting BACKUP_SPEC_SAMPLE_DB
-description "SAMPLE on /ess1 and /ess2 database backup" -db2system sanjose1
-user db2inst1 using aix123
Successfully done backup of history name "BACKUP_SAMPLE_DB"

C:\>gdbtask.cmd list backup -db2system sanjose1 -user db2inst1 using aix123
BACKUP_2005/10/18_13:30 2005/10/18 13:32:13     Success DEMOSPACE on /ess2
table space backup
BACKUP_SAMPLE_DB        2005/10/18 15:02:45     Success SAMPLE on /ess1 and
/ess2 database backup
```

9.5.4 Recovering a database

Before starting the database recovery, run the following transaction for this scenario. Example 9-11 shows the PROJECT table content before running transaction.

```
db2 connect to sample
db2 "drop table project"
db2 connect reset
```

Example 9-11 PROJECT table on SAMPLE database

```
db2inst1:/home/db2inst1> db2 "select projno,deptno,respemp from project"

PROJNO DEPTNO RESPEMP
------ ------ -------
AD3100 D01    000010
AD3110 D21    000070
AD3111 D21    000230
AD3112 D21    000250
AD3113 D21    000270
```

```
IF1000 C01    000030
IF2000 C01    000030
MA2100 D01    000010
MA2110 D11    000060
MA2111 D11    000220
MA2112 D11    000150
MA2113 D11    000160
OP1000 E01    000050
OP1010 E11    000090
OP2000 E01    000050
OP2010 E21    000100
OP2011 E21    000320
OP2012 E21    000330
OP2013 E21    000340
PL2100 B01    000020
20 record(s) selected.
```

Launching DB2 Recovery Expert and selecting a database

In this section, we use point-in-time recovery to recover the SAMPLE database using DB2 RE:

1. Start DB2 RE.
2. Select the SAMPLE database (Figure 9-44).

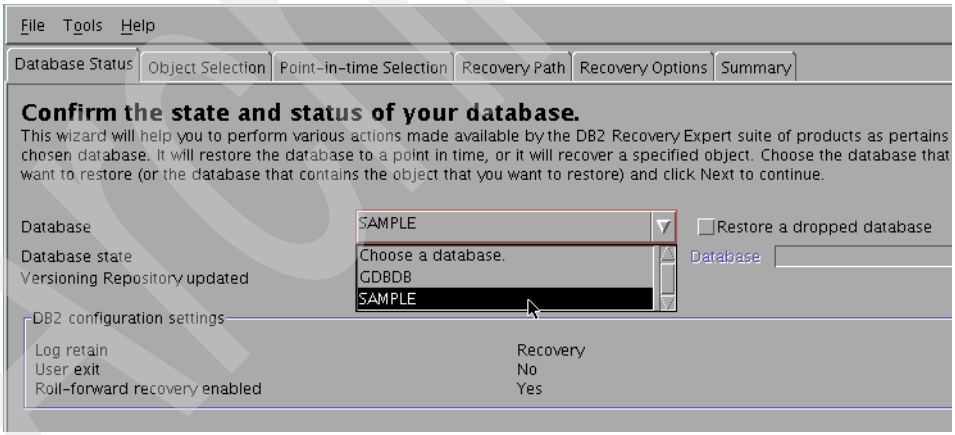


Figure 9-44 Launch DB2 RE to recover SAMPLE database

3. Go to Point-in-time Selection and click **Advance**.
4. Select **Backup images** and select **Fast Backup Media Type** (Figure 9-45 on page 297).

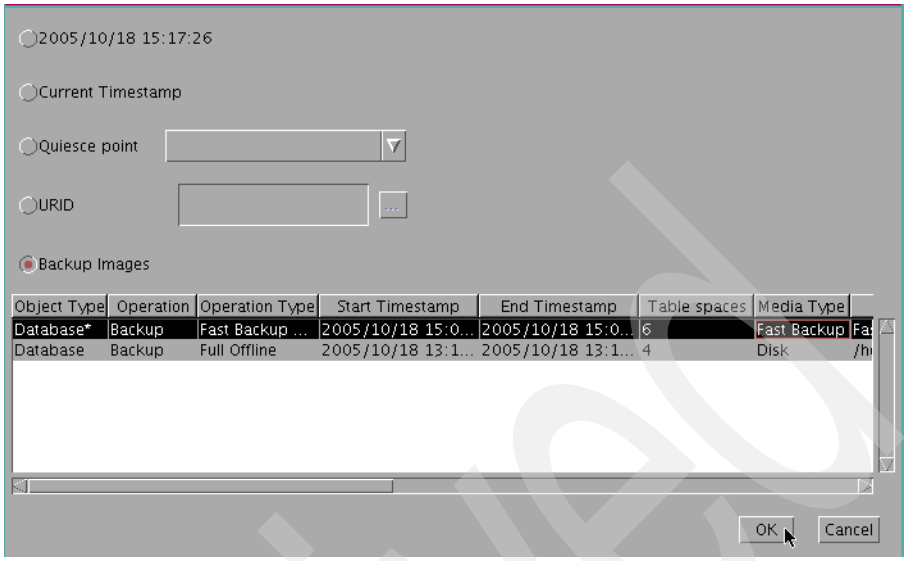


Figure 9-45 Select Fast Backup Media Type for Fast Backup point-in-time recovery

5. Go to **Recovery Path.** Click **OK** to go Recovery Path, If "`The Active Log Inventory Data may not be up-to-date`." message appears, click **OK**.
6. The *UNDO SQL using the DB2 logs* shows the cost is 0. For this scenario, select **Restore volume level from a Fast Backup image and rollforward** on Recovery Path (Figure 9-46).

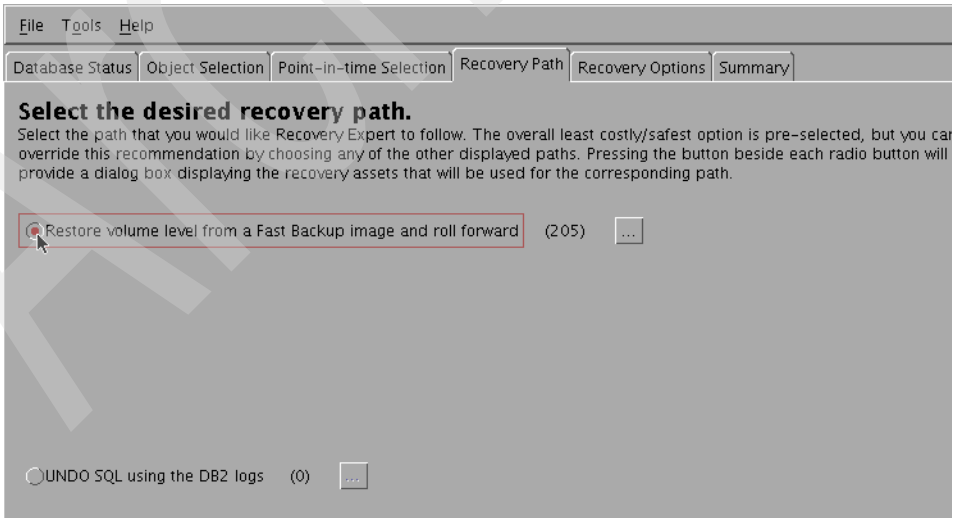


Figure 9-46 Select Fast Backup as Recovery Path for this scenario

7. Go to Summary and click **Run Now**.

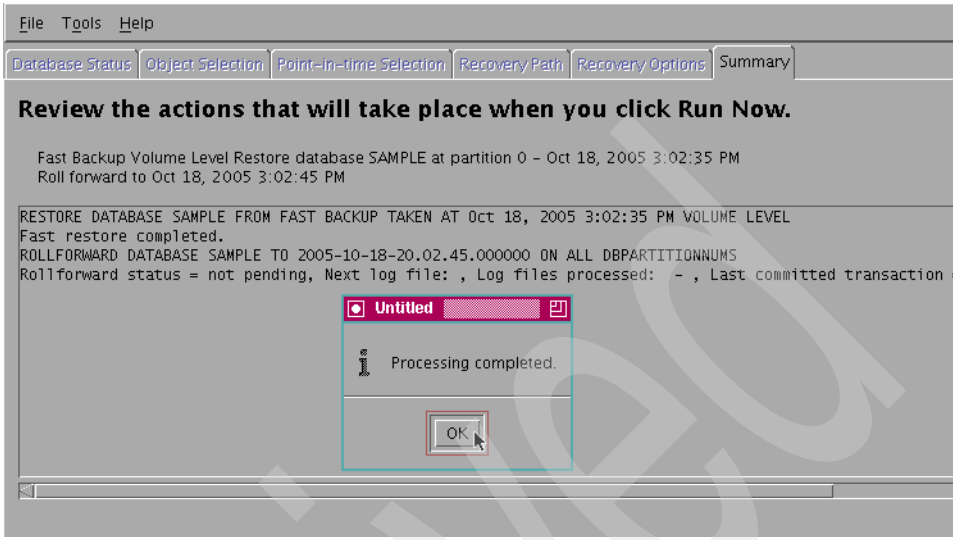


Figure 9-47 Go Run Now and its result

Example 9-12 shows the recovered PROJECT table content. You will find the dropped table PROJECT has been restored now.

Example 9-12 Recovered table PROJECT in SAMPLE database

```
db2inst1:/home/db2inst1/> db2 list tables

Table/View          Schema          Type  Creation time
------------------- --------------- ----- --------------------------
CL_SCHED            DB2INST1        T     2005-10-18-13.13.46.640389
DEPARTMENT          DB2INST1        T     2005-10-18-13.13.45.989211
EMP_ACT             DB2INST1        T     2005-10-18-13.13.46.163279
EMP_PHOTO           DB2INST1        T     2005-10-18-13.13.46.290691
EMP_RESUME          DB2INST1        T     2005-10-18-13.13.46.445285
EMPLOYEE            DB2INST1        T     2005-10-18-13.13.46.076095
IN_TRAY             DB2INST1        T     2005-10-18-13.13.46.680007
ORG                 DB2INST1        T     2005-10-18-13.13.45.812831
PROJECT             DB2INST1        T     2005-10-18-13.13.46.203032
SALES               DB2INST1        T     2005-10-18-13.13.46.600372
STAFF               DB2INST1        T     2005-10-18-13.13.45.903628

  11 record(s) selected.

db2inst1:/home/db2inst1> db2 "select projno,deptno,respemp from project"

PROJNO DEPTNO RESPEMP
```

```
------ ------ -------
AD3100 D01    000010
AD3110 D21    000070
AD3111 D21    000230
AD3112 D21    000250
AD3113 D21    000270
IF1000 C01    000030
IF2000 C01    000030
MA2100 D01    000010
MA2110 D11    000060
MA2111 D11    000220
MA2112 D11    000150
MA2113 D11    000160
OP1000 E01    000050
OP1010 E11    000090
OP2000 E01    000050
OP2010 E21    000100
OP2011 E21    000320
OP2012 E21    000330
OP2013 E21    000340
PL2100 B01    000020
20 record(s) selected.
```

9.6 Considerations about Fast Backup use

This section described some considerations about Fast Backup use.

9.6.1 Point-in-time of Fast Backup

A database is placed on the suspended I/O state during the Fast Backup backing up the database. The time of this suspended I/O set to the database is the point-in-time of this database backup.

A table space is placed on the quiesce share state during the Fast Backup backing up the table space. The time of this quiesce set to the table space is the point-in-time of this table space backup.

The database and table space are not available when Fast Backup is performing the backup. Applications are required to retry operations because of `SQL0290N` error return code for the accessing the objects during Fast Backup backing up the table space.

From the system availability point of view, it might be more efficient to backup table spaces rather backup the database.

9.6.2 Recovery paths of DB2 Recovery Expert

DB2 RE Fast Backup provides the capability to backup and restore the large scale database or table space efficiently. It is very useful in reducing the recovery time. But keep in mind that a backup of the object must be available as a recovery path in order for DB2 RE to recover it. Fast Backup image, in general, is a lower cost recovery selection when the database or table space to be recovered is large.

DB2 RE uses several recovery assets such as backups, logs, versioning repository, and Fast Backup backup image in recovery process. The Fast Backup backup image one of the DB2 RE database or table space recovery path but is not available for DB2 RE dropped object recovery. To recovery dropped object, DB2 RE requires versioning repository and DB2 offline backup. Depends on the recovery requirements, in addition to the Fast Backup, DB2 offline backup should also be planned.

To recovery database or table space to a point-in-time after restoring the Fast Backup backup image, DB2 RE rolls forward the transactions using DB2 logs. The more closer the time the backup image is taken to the required recovery point-in-time, the less rollforward time is required. Therefore, consider to backup objects more often and have multi backup copies in multiple volumes.

Fast Backup recovery asset

Both Fast Backup backup image and the backup image taken by DB2 **BACKUP** command are recovery assets. In some cases, the backups made by Fast Backup might not be available for the recovering objects by DB2 RE, even if Fast Backup backup is scheduled frequently and carefully.

For example, the database is getting larger and some tables are moved to new created table spaces on the other network storages. The new created table spaces are backed up frequently to reduce the impact on availability. However, if a table is dropped and a point-in-time table space recovery is required to recovery the dropped table, the Fast Backup table space backups can't be used as the recovery assets of DB2 RE because the database catalog information cannot be recovered by the table space backup image. New table space backup is required for the further table space recovery.

9.6.3 DB2 Recovery Expert Fast Backup troubleshooting

Fast Backup provides several logs for problem determination. It is especially helpful when encounter problems or I/O errors in the network storages.

DB2 Recovery Expert Fast Backup server log

Fast Backup server logs are in the <RE_install_dir>/fastbackup/log directory. You can check the latest gdbsp-node#-yyyymmddssssss.log file (yyyy:year, mm:month, dd:day, ssssss:seconds) for error message if Fast Backup operation failed.

DB2 Recovery Expert Fast Backup client log

Fast Backup client logs are in the <instance_home>/IBM/gdb/log directory. In Windows systems, client logs are in the %HOMEPATH%\Application Data\gdb\log directory.

The same PVID problem on the object restore method

To use Fast Backup object restore method, the target volume needs to be assigned to the system. In this case, the subjected problem cab happen by the host system re-configuration function on reboot. Let us think through the following scenario.

The SAMPLE database has been backed up through Fast Backup task which specifies the ObjectRestore as restore method. The target volume was assigned to the host system and is copied from the source volume with its PVID (physical volume ID). The target volume was not included in any volume group and the status was not active. The recovery process of DB2 RE Fast Backup works well.

But, after a while, if the host system is rebooted for some reason, it is highly possible that the configuration of the target volume might be changed. If DB2 RE is called to recover a database using Fast Backup database backup as recovery path and the target volume configuration has been changed after system reboot, you will receive “`mismatch volume group`” error message. See Figure 9-48 on page 302.

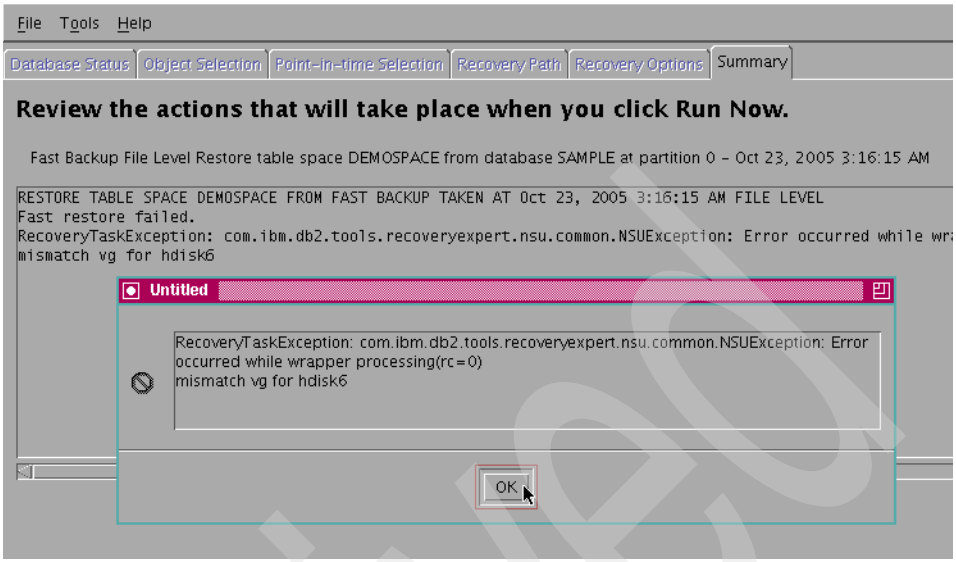


Figure 9-48 DB2 RE DEMOSPACE table space recovery failure

Error messages were found in Fast Backup server log gdbsp-xxxx.log (see Example 9-13). the log shows that hdisk2 and hdisk4 are correctly configured for /ess1 and /ess2 mount point but there was an error on the volume group mapping for hdisk6.

Example 9-13 Fast Backup log <RE_install_directory>/fastbackup/log/gdbsp-xxxx.log

```
2005/10/23 03:35:59(Thread-1): .sp.WrapperProcess.checkExitValue() stderr:
mismatch vg for hdisk6

2005/10/23 03:35:59(Thread-1): .sp.WrapperProcess.checkExitValue() try to
getStdOutString ()
2005/10/23 03:35:59(Thread-1): .sp.WrapperProcess.checkExitValue() stdout:
#./gdbgmp enter
#/usr/sbin/lspv
E]mismatch vg for hdisk6
#/usr/sbin/lsvg -l vgessdata
-mp "/essdata/" -mt SAN -mo "lv:lvessdata,vg:vgessdata" -t ibmess -id 22513
-vol 20022513 -pv hdisk1
#/usr/sbin/lsvg -l vgess1
-mp "/ess1/" -mt SAN -mo "lv:lvess1,vg:vgess1" -t ibmess -id 22513 -vol
20122513 -pv hdisk2
#/usr/sbin/lsvg -l vgess2
-mp "/ess2/" -mt SAN -mo "lv:lvess2,vg:vgess2" -t ibmess -id 22513 -vol
20522513 -pv hdisk4
#/usr/sbin/mount
#./gdbgmp exit
```

```
2005/10/23 03:35:59(Thread-1): .sp.WrapperProcess.checkExitValue() try to raise
()
2005/10/23 03:35:59(Thread-1): .common.NSUException.newInstance() enter
2005/10/23 03:35:59(Thread-1): .common.NSUException.newInstance()
exit(com.ibm.db2.tools.recoveryexpert.nsu.common.NSUException: Error occurred
while wrapper processing(rc=0)
mismatch vg for hdisk6
)
2005/10/23 03:35:59(Thread-1): .sp.WrapperProcess.checkExitValue() throw
code=19200,sqlstate=38501,msg=Error occurred while wrapper processing(rc=0)
mismatch vg for hdisk6

2005/10/23 03:35:59(Thread-1): .sp.Wrapper.executeParalell() thrown(logged)
```

Consequently, the hdisk6 physical volume ID is checked using lspv command. Its PVID is the same as PVID of hdisk4. Using rsList2105s.sh, we also found that hdisk6 is the target volume ESS 20822513.

During host system reboot, the host system has configured the formatted volume to active state and produce duplicated PVID even the volume is not included in any volume group. The Fast Backup recovery failure is caused by the duplicated PVID on the target volume.

Example 9-14 The same PVID problem happens on hdisk6

```
root:/> lspv
hdisk0          00257400809bc5df                    rootvg          active
hdisk1          002574008461db66                    vgessdata       active
hdisk2          00257400fc2ae17a                    vgess1          active
hdisk4          0025740099e9f211                    vgess2          active
hdisk3          none                                None
hdisk6          0025740099e9f211                    vgess2          active

root:/> /opt/ibm/ESScli/rsList2105s.sh
disk name       2105 serial number
---------       ------------------
 hdisk1           20022513
 hdisk2           20122513
 hdisk3           20422513
 hdisk4           20522513
 hdisk6           20822513
```

This status of `the same PVID on the target volume` is able to be reset by the command **`chdev -l hdisk6 -a pv=clea`**". See Example 9-15 on page 304.

Example 9-15 Recovery from the same PVID status

```
root:/> umount /ess2
root:/> varyoffvg vgess2

root:/> chdev -l hdisk6 -a pv=clear
hdisk6 changed

root:/> varyonvg vgess2
root:/> mount /ess2

root:/> lspv
hdisk0          00257400809bc5df                    rootvg          active
hdisk1          002574008461db66                    vgessdata       active
hdisk2          00257400fc2ae17a                    vgess1          active
hdisk4          0025740099e9f211                    vgess2          active
hdisk3          none                                None
hdisk6          none                                None
```

Now DB2 RE Fast Backup ObjectRestore works fine without any data lost (Figure 9-49).

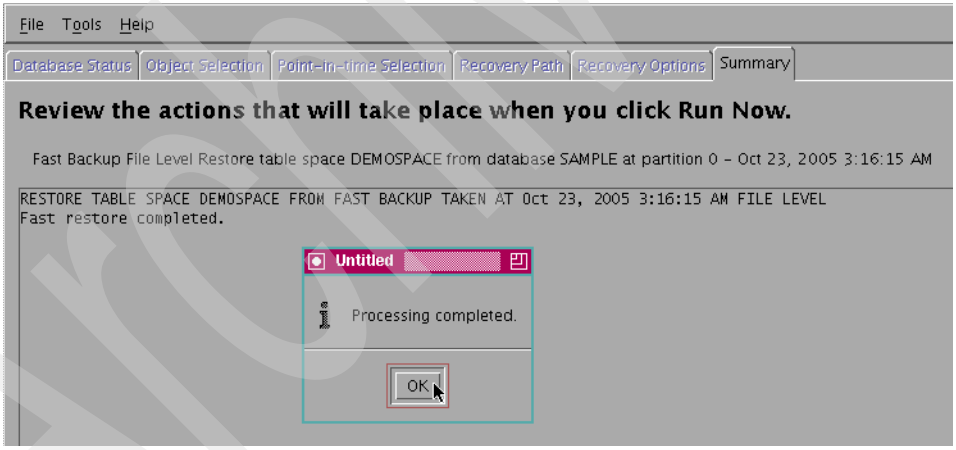


Figure 9-49 DB2 RE DEMOSPACE table space recovery complete

10

Installing and using the DB2 Grouper component

This chapter includes information about how to install the DB2 Grouper component in a multiplatform environment. It also explains how to use the component to create sets that DB2 Recovery Expert uses. It includes the following sections:

10.1 Overview of DB2 Grouper

DB2 Grouper is a client-server utility that you can use to identify table relationships for a single database. DB2 Grouper discovers, records, allows administration for, and provides group relationship information to other IBM DB2 tools as a basis for database administration activities. These activities can be performed by tools that work with DB2 Grouper or manually by the user. Currently in the multiplatform environment, you can use DB2 Grouper information with DB2 Test Database Generator, DB2 Data Archive Expert, and DB2 Recovery Expert (DB2 RE).

10.1.1 How DB2 Grouper works with DB2 Recovery Expert

If you have DB2 Grouper installed, you can recover related DB2 objects using DB2 RE. In a recovery scenario, when the first recovery completes for your selected object (table or table space), DB2 RE asks if you want to recover related objects. If you answer yes, you can select the related objects that you want to recover. You then follow the normal DB2 RE recovery steps to recover the related object. If there are more related objects, the recovery process prompts you for each object until all of the related objects are recovered.

To see an example of how to recover related tables, see 6.6, “Point-in-time recovery with referential integrity and DB2 Grouper” on page 166.

10.1.2 Making DB2 Recovery Expert aware of DB2 Grouper sets

Before DB2 RE can know about related objects, you must first use the DB2 Grouper client to identify the tables in a database that are related. You can use the DB2 Grouper client to specify both DB2-enforced and non DB2-enforced referential constraints to DB2 Grouper. These relationships are global for DB2 Grouper and all products that work with DB2 Grouper and are used by all subsequent grouping operations.

With DB2 RE, you use the DB2 Grouper client to create a set and a version to hold the groups of interest. You must specify one or more starting point tables for use in searching the catalog. You can specify additional options to tailor the output of the group. Next, you run a Group Discovery, which calls the server DB2 stored procedure and assembles the options and the information in the catalog to determine one or more groups of related tables. You must mark the discovered group as available for the companion products to use. DB2 RE uses the results of this grouping operation. The remaining sections in this chapter discuss these steps in more detail.

10.2 Installing the DB2 Grouper component

The DB2 Grouper component is a client-server program. Thus, you need to install files on both a server and a client.

10.2.1 Installing the DB2 Grouper server

The DB2 Grouper *server* is actually a set of stored procedures that are installed into each database that you want to investigate. There is no process or service that runs in the background, as you might expect when something is called a *server*. When you initiate a Group Discovery from the DB2 Grouper client, the stored procedure executes and populates the DB2 Grouper tables in the database.

Important: The DB2 Grouper documentation, which includes the user's guide and the help files, uses the term *DB2 Grouper server*. This term really means the individual database into which you have installed the DB2 Grouper stored procedures. It does not refer to the instance or machine onto which you installed DB2 Grouper. You must install the DB2 Grouper stored procedures into *each* database with which you want to work. Thus, you need to think of each database as the *DB2 Grouper server* in the context of the DB2 Grouper documentation.

Installing the DB2 Grouper server in a UNIX environment

Attention: In this section, we describe the steps that we performed when installing DB2 Grouper server on a Linux environment in the ITSO lab that we used for this book. This is not a replacement for the official installation instructions, which you can find in *IBM DB2 Grouper User's Guide*, SC18-7409.

To install the DB2 Grouper server in a UNIX environment, follow these steps:

1. Login as root.
2. You must run the **`db2profile`** of the instance you want to use. We suggest that you also verify that the DB2 environment is set properly by issuing a simple command such as **`db2level`**. Example 10-1 on page 308 shows an example of this command from Linux.
3. Verify that you have an appropriate Java in the `PATH` for root (see also Example 10-1 on page 308).
4. Change to the directory where the setup program exists.

5. Run the installation program, in either graphical mode or console mode. We show the graphical mode install on a Linux server, which is launched by issuing the **./setupxlnx** command, as shown in Example 10-1.

Example 10-1 Installation pre-work for root

```
[db2inst1@lead db2inst1]$ su - root
Password:
[root@lead root]# . /home/db2inst1/sqllib/db2profile
[root@lead root]# db2level
DB21085I  Instance "db2inst1" uses "32" bits and DB2 code release "SQL08022"
with level identifier "03030106".
Informational tokens are "DB2 v8.1.0.89", "OD_14086", "MI00105_14086", and
FixPak "9".
Product is installed at "/opt/IBM/db2/V8.1".
[root@lead root]# java -version
java version "1.4.1"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.1)
Classic VM (build 1.4.1, J2RE 1.4.1 IBM build cxia321411-20040301 (JIT enabled:
jitc))
[root@lead Server]# cd
[root@lead root]# cd /software/Grouper/Server/
[root@lead Server]# ll
total 7108
-rwxrwxrwt    1 bin      daemon         13 May  9 13:55 media.inf
-rwxrwxrwt    1 bin      daemon      30053 May 31 11:57 README.txt
-rwxrwxrwt    1 bin      daemon    6563781 May  9 13:55 setup.jar
-rwxrwxrwt    1 bin      daemon     658570 Jun 10 09:25 setupxlnx
[root@lead Server]# ./setupxlnx
InstallShield Wizard

Initializing InstallShield Wizard...

Searching for Java(tm) Virtual Machine...
........
```

At this point, the GUI window appears for the installation to proceed, as shown in Figure 10-1 on page 309. The User's Guide refers to this series of dialogs as the *installation wizard*. The rest of the install steps are described below with screen images.

6. Click **Next** on the Welcome window.

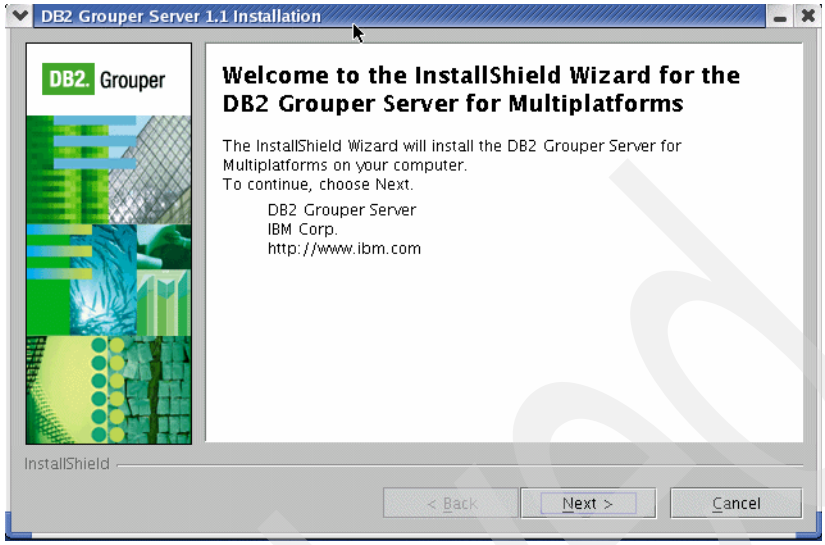


Figure 10-1 DB2 Grouper server installation welcome page

7. You can select an installation path for the DB2 Grouper server. We chose the default (Figure 10-2). Click **Next** to continue.

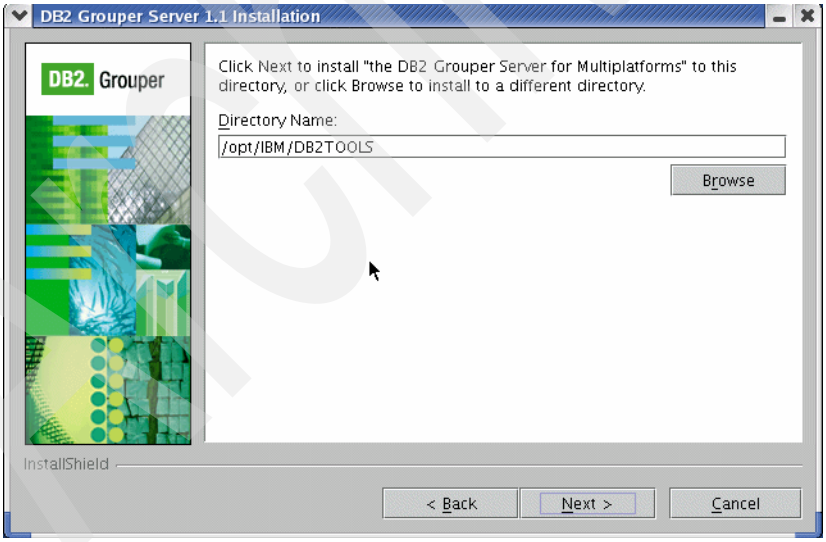


Figure 10-2 Select a DB2 Grouper server install path

The installed software part of DB2 Grouper is not large, because the *server* is not a server in the traditional sense — it is not a running process or a service. Instead, it is a set of stored procedures that exist in each database that you want to use with DB2 Grouper. The installation wizard, therefore, displays a series of questions to allow you to install (sometimes called *configure*) the DB2 Grouper objects in one database (Figure 10-3 on page 311).

In this example, we want to configure DB2 Grouper for the SAMPLE database. Select the correct DB2 version, and scroll down for one more question. DB2 Grouper comes with a set of sample tables, which are based on the DB2 SAMPLE database tables but with additional referential integrity added. (These tables are described in detail in *IBM DB2 Grouper User's Guide*, SC18-7409, but we do not describe them here.) They are tables and other objects that are created in the DB2 Grouper EGFSPACE table space.

We recommend that you install these sample tables into a small, simple database when you are getting started with DB2 Grouper. The examples in the user's guide refer to these tables, and it is quite helpful to use them as a learning tool.

Important: Do not forget to scroll down on this window (see Figure 10-3 on page 311)! There is another setup question to answer.

8. Click **Next** to continue with the wizard.

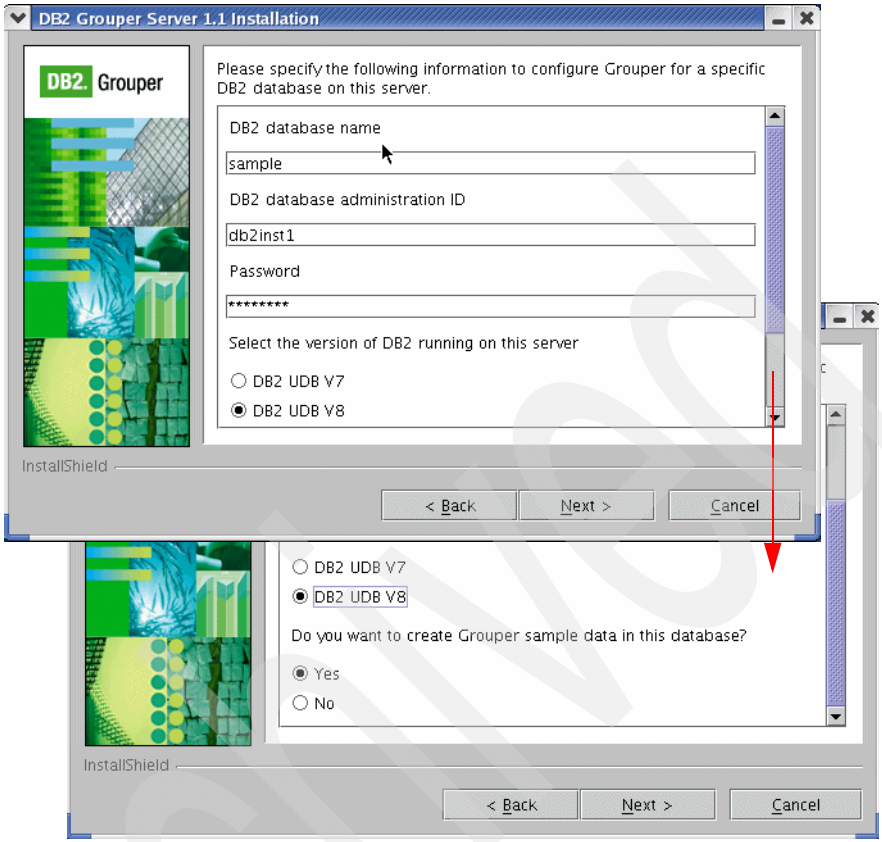


Figure 10-3 DB2 Grouper server configuration options, scroll down

In our testing, we made an error when we first installed DB2 Grouper. We show the error here, so that you will not make the same mistake. We logged in as root, but we forgot to run the **db2profile** for the instance, as instructed in “Installing the DB2 Grouper server in a UNIX environment” on page 307. Thus, when the installer tried to run a DB2 command, it could not find the command. Figure 10-4 shows the resulting error.

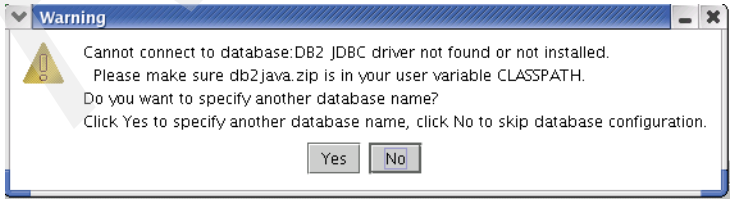


Figure 10-4 DB2 Grouper server, forgot to run db2profile

You can skip the database configuration and install the DB2 Grouper server files by choosing No. However, in this case, we decided to start over. So, we choose Yes, exited the install, ran the correct **db2profile**, and then proceeded with the installation process.

The next step is to create the DB2 Grouper objects in the target database (SAMPLE in our case.) DB2 Grouper puts all its objects in table space named EGFSPACE (Figure 10-5).

9. Click **Yes** to create the EGFSPACE table space.

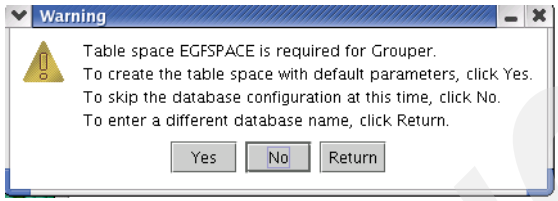


Figure 10-5 DB2 Grouper table space confirmation

Before continuing with the installation process, you have one last chance to confirm the installation path and see an space estimate for the DB2 Grouper server files, as shown in Figure 10-6. This space estimate does not include the space for the EGFSPACE table space, but the table space is not large and should not tax your system.

10. Click **Next** on the confirmation screen to perform the installation.

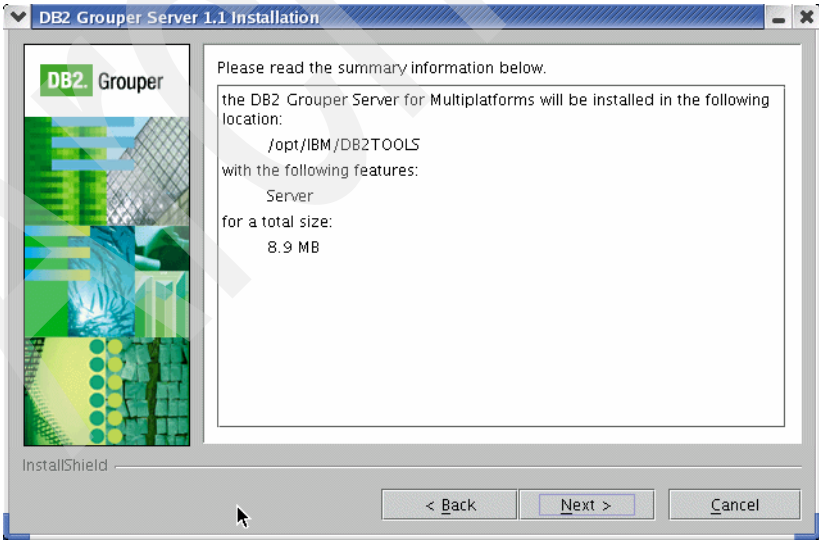


Figure 10-6 DB2 Grouper server confirmation

When the installation starts, there are several of the usual progress messages on the window, we do not show them here. Assuming that you have no errors, when the installation completes, the wizard launches the configuration using the database information that you supplied earlier.

The window shown in Figure 10-7 warns you that certain prerequisites are checked. These requirements are well documented in *IBM DB2 Grouper User's Guide*, SC18-7409. You should refer to them prior to installing DB2 Grouper. However, even if the requirements are not all met, you can continue the installation.

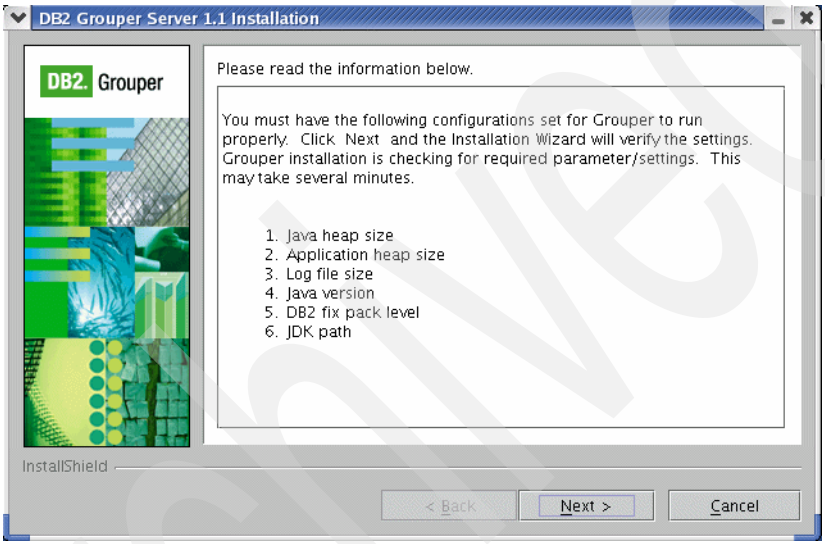


Figure 10-7 DB2 Grouper server prerequisite checking

11. Click **Next** to allow the checks to proceed.

    There are several progress screens during the remainder of the configuration. We do not show them here. If the requirement check fails, a window opens that identifies which parameters were found to be inadequate. However, you can continue the installation process.

    Figure 10-8 on page 314 shows that the installation has completed successfully. If there are errors, a message displays and refers you to a log file with more information.

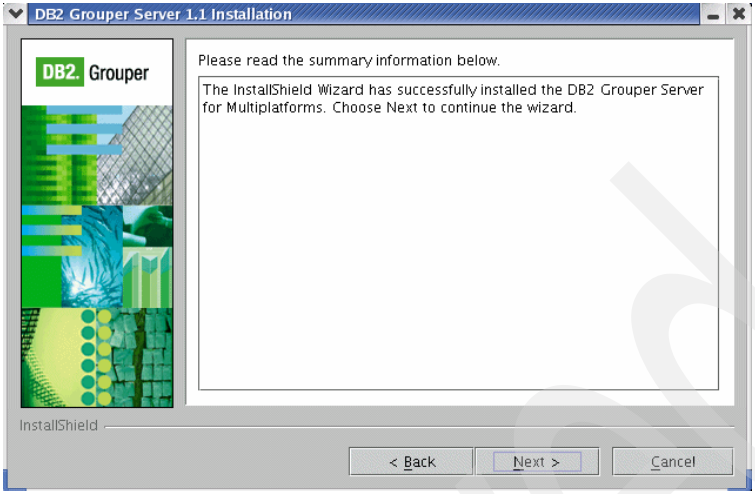


Figure 10-8 DB2 Grouper install and configuration complete

12. Click **Next** to continue.

    A window opens (Figure 10-9) which advises you to log out and back in again so that the server installation sets an environment variable in the /etc/profile directory. Thus, you need to run the profile again. (This window does not open in a Windows environment.) The change to /etc/profile affects all users. So, you should verify that the environment variables are set properly for the instance owner as well.

13. Click **Finish** to exit.

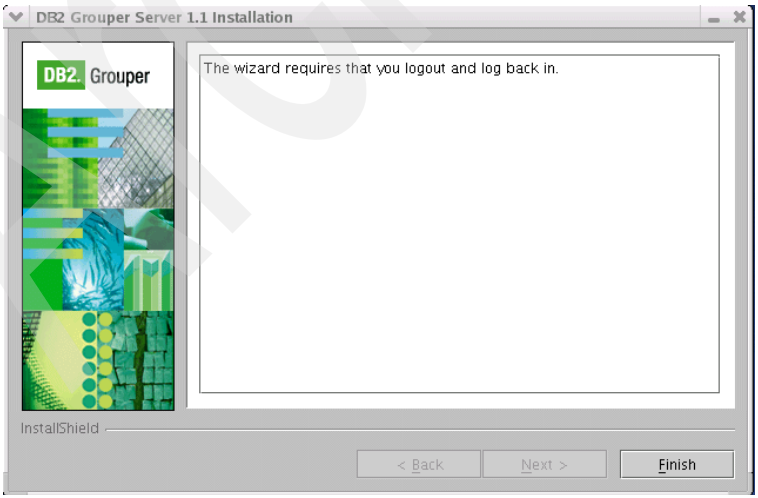


Figure 10-9 DB2 Grouper server installation post-install step

At this point, we have installed the DB2 Grouper server files and configured DB2 Grouper in one database (SAMPLE) on one instance (db2inst1).

Important: Installing DB2 Grouper server creates a new table space in your database. If you make table space backups, you should add the EGFSPACE to your backup scripts.

Installing the DB2 Grouper server in a Windows environment

When you install DB2 Grouper server in a Windows environment, the installation looks the same as described in “Installing the DB2 Grouper server in a UNIX environment” on page 307. There are only a few things to consider for Windows:

- You must install as the Administrator user, not a user with administration rights in an administration group. This is documented in *IBM DB2 Grouper User's Guide*, SC18-7409.
- You must have a valid Java existing in your `PATH`, or the installation will fail, although the InstallShield installs its own Java.

We do not show a Windows installation in this book.

Installing DB2 Grouper in additional instances and databases

When you install DB2 Grouper for the first time, you must install as root or administrator user. The first step is to run the **`db2profile`** of the instance owner, so that the installer has some context in which to run. The installation steps combine the software installation with the initial setup for a database. The installation program asks for a single database name in which to create the EGFSPACE table space and contents. This is fine if you only have one instance and one database.

However, if you have another database in that instance, the DB2 Grouper EGFSPACE needs to exist in *each* database with which you want to work. Alternatively, if you have another instance, you do not re-install the DB2 Grouper code itself, just to get to the configuration portion of the installation process.

In these cases, you need to run a utility that is provided in the DB2 Grouper path, which creates and populates the EGFSPACE in additional databases. You need to run this utility for each database individually. Also, this setup utility is GUI-based, and there is no command line option. In a UNIX environment, you must run this configuration utility as root.

Configuring DB2 Grouper to work in another database on UNIX

This example shows how to install DB2 Grouper objects into a database on a Linux server named LEAD, the DB2 UDB V8.2 instance is db2insl1 on port 50001, and the database is named TRADSIMP. (We use this particular database in another example in 6.6, “Point-in-time recovery with referential integrity and DB2 Grouper” on page 166.)

Before you run the script, you must establish the correct environment. The environment for this example is as follows:

1. Ensure that you have X-windows capability.
2. Login or su to root.
3. Run the **db2profile** for the instance with which you are working.
4. Change to the DB2 Grouper server directory under the installation path.

Example 10-2 shows these steps.

Example 10-2 Establish environment for DB2 Grouper maintenance

```
[db2insl1@lead server]$ su - root
Password:
[root@lead root]# . /home/db2insl1/sqllib/db2profile
[root@lead root]# cd /opt/IBM/DB2TOOLS/grouper/server
```

When the environment is set up correctly, then run the **cfgNewDb.sh** script with the appropriate parameters, as shown in Figure 10-10.

```
Usage: cfgNewDb database host port
[root@lead server]# ./cfgNewDb.sh  tradsimp localhost 50001
Client db2 version: 8.1.0.89
os Linux***
```

Figure 10-10 Run cfgNewDb.sh for TRADSIMP database

When the script starts, it launches a window to guide you through the remainder of the configuration. We show the configuration for TRADSIMP. These are the steps:

1. Authenticate to the database to which you want to add DB2 Grouper, as shown in Figure 10-11 on page 317. The database name is filled in with what you supplied on the command line. Verify that this name is correct, and supply the correct user ID and password. Then click **OK**.

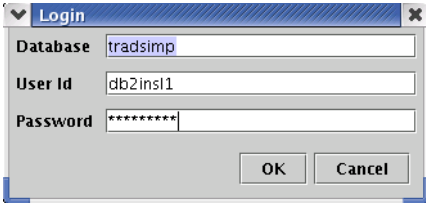


Figure 10-11 DB2 Grouper setup, login prompt

The steps from here are similar to the steps that occur during the installation. However, the windows look a bit different, so we show them again here.

2. The first question is about the prerequisite check (Figure 10-12). Click **Yes**.

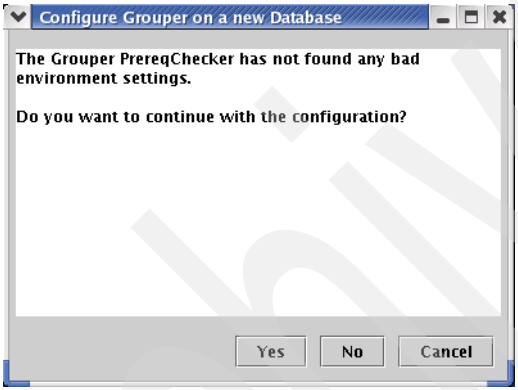


Figure 10-12 DB2 Grouper setup, prerequisite checker

3. DB2 Grouper creates a new table space named EGFSPACE in the database that you are configuring (Figure 10-13). Click **Yes**.

Figure 10-13 DB2 Grouper setup, create EGFSPACE

Figure 10-14 shows that the table space was created.

You are always offered the option of creating the sample tables in the database that you are configuring. We are satisfied with using the DB2 Grouper sample tables only in the SAMPLE database that we configured earlier during the installation process. So, we do not want to create them in our TRADSIMP database. You might not want to have extra tables in a production database, even if they are small and are in their own table space.

4. Click **No** to skip the step that creates the sample tables and to proceed with the installation.

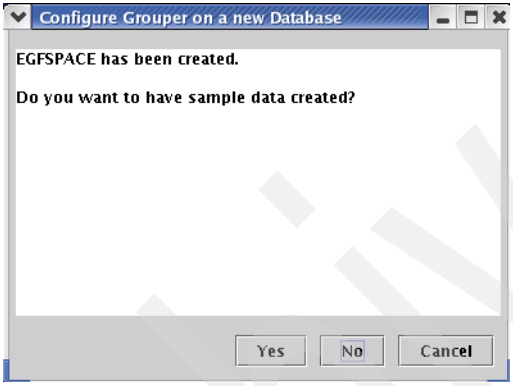


Figure 10-14 DB2 Grouper setup, create sample data

The configuration now occurs, and it writes progress messages to the window, as shown in Figure 10-15 on page 319. The messages are overwritten with each step, so it is a good idea to watch the window in case something goes wrong.

The configuration does capture its log messages to a file. You should examine that log file after the configuration or installation completes just to be sure that nothing strange happened.

The log file is named grpserver.log and is located in the path in the `$DB2TOOLS` environment variable. In our Linux example, the file is in /opt/IBM/DB2TOOLS/grpserver.log. The `$DB2TOOLS` environment variable is set when you install DB2 RE or DB2 Grouper. Windows also has the same environment variable.

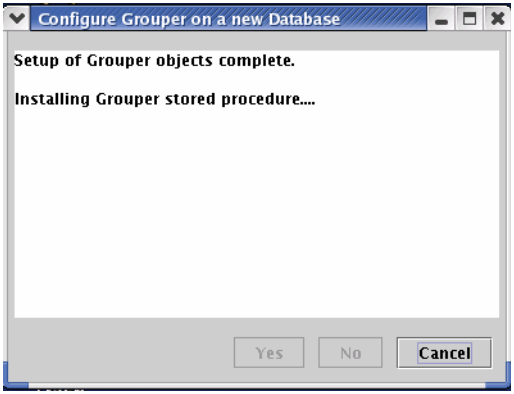


Figure 10-15 DB2 Grouper setup, progress box

Figure 10-16 shows that the configuration of DB2 Grouper into the TRADSIMP database completed successfully.

5. Click **OK** to exit.

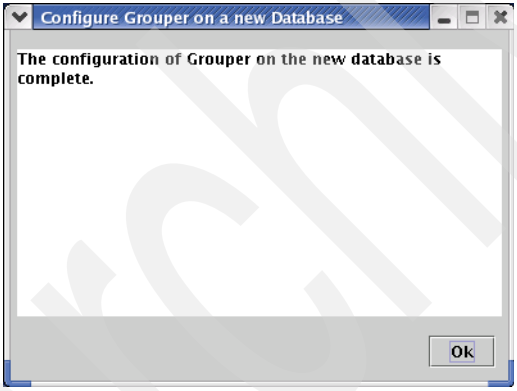


Figure 10-16 DB2 Grouper setup, complete

If you want to verify the installation, you can check the existence of EGFSPACE in the database and the DB2 Grouper tables named `SYSTOOLS.EGF_xxxxx`. Although we did not create the DB2 Grouper sample tables, DB2 Grouper has other tables that it uses to store the information that you create about related tables.

Configuring DB2 Grouper to work in another database on Windows

In a Windows environment, the steps are similar to the steps for UNIX. We do not show a full example here. The differences from the UNIX example are:

- To run the configuration command, you do *not* need to be in the DBADM group.
- Open a DB2 CLP window to run the command.
- If you have multiple instances on your server, you should establish the `DB2INSTANCE` environment variable in the session.
- The script name is **`cfgNewDb.bat`**.

For example, to configure DB2 Grouper in a new database named RECART on the local instance, you would type:

```
cfgNewDb RECART localhost 50000
```

The remainder of the GUI configuration steps are the same as those described in “Installing the DB2 Grouper server in a UNIX environment” on page 307.

10.2.2 Installing the DB2 Grouper client

The DB2 Grouper client is available only for Windows platforms. The DB2 Grouper functions are run via a remote connection to each database with which you want to work. You can run the installation program as any user that has rights to install software on your machine. To install the DB2 Grouper client, follow these steps:

1. To start the installation process, run the `setup.exe` program.

   This launches the InstallShield program to complete the installation, as shown in Figure 10-17 on page 321.

2. Click **Next** to proceed with the installation process.

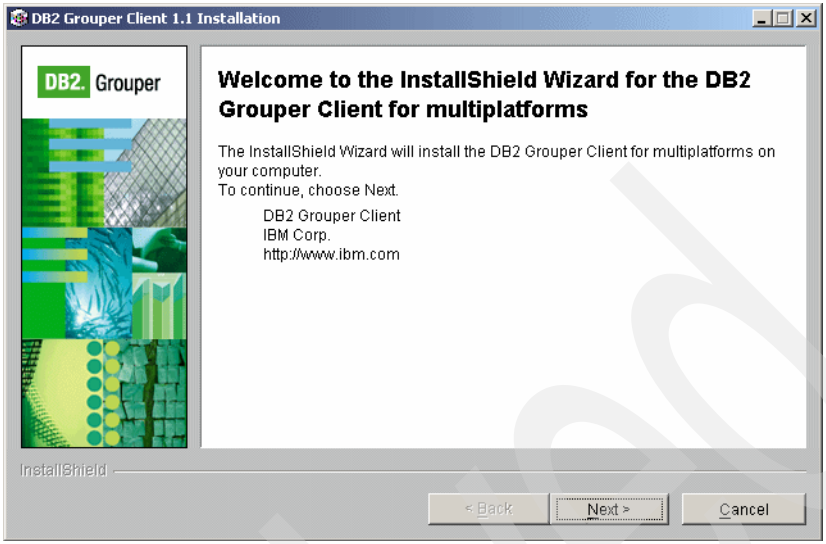


Figure 10-17 DB2 Grouper client installation welcome

3. You next specify the location where you want to install the files DB2 Grouper client files (Figure 10-18 on page 322).

   The default location is C:\Program Files\IBM\DB2TOOLS. In our testing, we prefer the tool in a higher path, so we change the installation directory it to C:\DB2TOOLS. (This is not a recommendation, rather just how we did it for the book.)

   Note: This installation directory is only the top-level directory, DB2 Grouper creates another path named `grouper` underneath the name that you specify here. If you choose a name such as C:\DB2TOOLS\grouper, you end up with C:\DB2TOOLS\grouper\grouper, which is probably not what you want.

   Other IBM DB2 tools products can use DB2 Grouper, so it is best to install them all under the same path.

4. Click **Next** to continue the installation process.

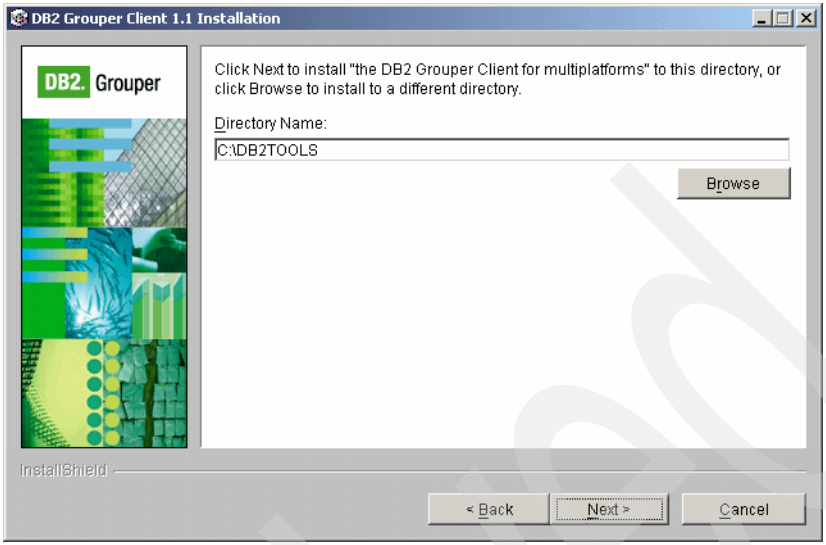


Figure 10-18 Choose DB2 Grouper client install directory

Figure 10-19 on page 323 shows the final confirmation before you initiate the installation. DB2 Grouper client needs to use Java, so DB2 Grouper might decide to install its own JVM under the DB2 Grouper client installation directory. So, the space amount can be more or less than what is shown here. You have no control over this but be aware that you might see a different size listed. The DB2 Grouper software does not consume a large amount of space, so there should be no issues for space planning.

5. Click **Next** to complete the installation process.

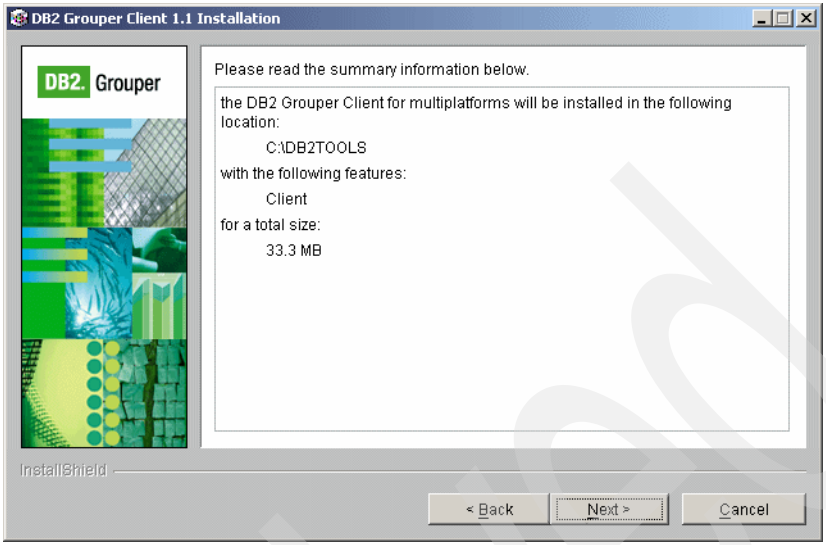


Figure 10-19 DB2 Grouper install confirmation

When the installation begins, you see the typical progress messages, and then the final window, as shown as in Figure 10-20.

6. Click **Finish** to exit the installation program.

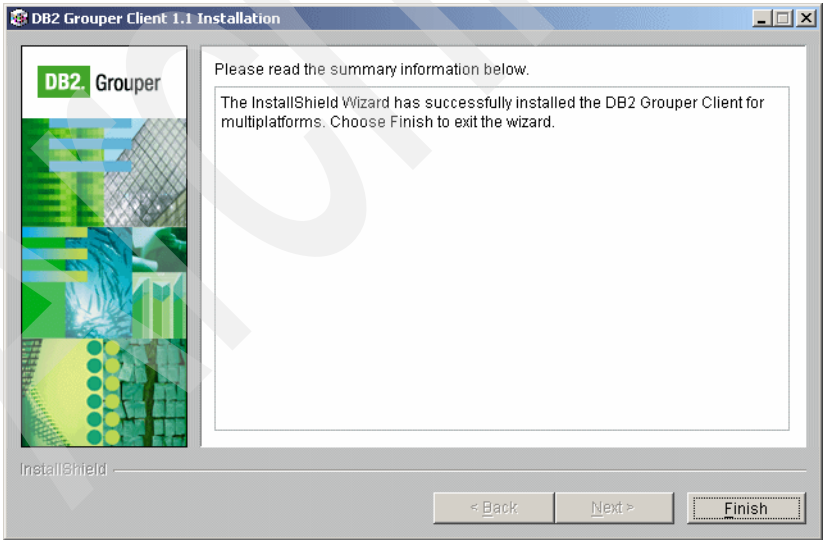


Figure 10-20 DB2 Grouper client install complete

If there are any errors during the installation process, you can look in the installation log file, named grpclient.log, which is placed in the top-level installation directory, C:\DB2TOOLS in our case.

10.3 Using the DB2 Grouper component

Now that you have the DB2 Grouper component installed, this section explains how to bind DB2 Grouper packages, how to use the DB2 Grouper launchpad, and how to create DB2 Grouper sets for DB2-enforced referential integrity.

Continuing to use our lab scenario, we now have DB2 Grouper installed on a Linux server, configured for one database named SAMPLE in the instance db2inst1, and one database named TRADSIMP in another instance db2insl1. Our Windows PC has the DB2 Grouper client installed. Now, we can start the client and connect to the database to begin the initial group discovery.

Note: The *IBM DB2 Grouper User's Guide*, SC18-7409 provides a detailed installation verification process that you can follow.

DB2 RE does not currently support the DB2 Grouper sets that are identified by unit-of-work discovery, so we do not show that here. In this section, we have two examples of the non-enforced and DB2-enforced relationships with which DB2 RE can work.

10.3.1 Binding the DB2 Grouper packages

To bind the DB2 Grouper packages, follow these steps:

1. Start the DB2 Grouper client. The first thing that you see is a connection prompt with the database, user ID, and password, as shown in Figure 10-21 on page 325.

   In our example, the database that we are working with is remote. So, we need to have this database already cataloged on the PC before we start DB2 Grouper. We do not show those steps here. In our case we called it LEADSIMP.

   This group set is used by the recovery scenario that is described in 6.6, "Point-in-time recovery with referential integrity and DB2 Grouper" on page 166.

2. Click **OK** to connect.

> **Tip:** The login box does not accept Enter as a default for OK. You can click **OK**, or you can tab to OK to give it focus and then press Enter or the space bar to select it.

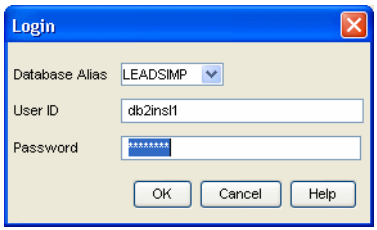


Figure 10-21 DB2 Grouper client startup

Figure 10-22 shows a message that packages are not found on the server. This happens the first time that you connect to the database after configuring DB2 Grouper for a new database.

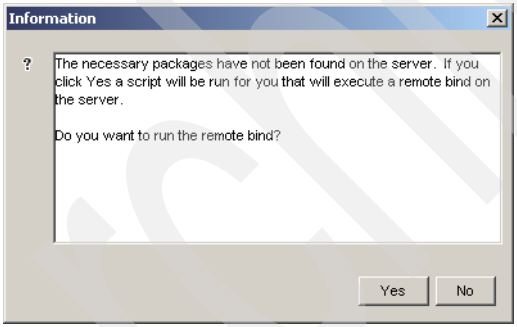


Figure 10-22 Packages not found

3. Click **Yes** to bind the packages for the database, TRADSIMP in this scenario.

> **Tip:** Remember our discussion earlier about the context of the term *server* in DB2 Grouper. You need to perform these binds in each database on which you configured DB2 Grouper.

The bind launches a DB2 CLP window that shows the binds as they occur. Example 10-3 shows a partial clip.

Example 10-3 Remote binds in progress

```
[ibm][db2][jcc][sqlj] Begin Bind
[ibm][db2][jcc][sqlj] Loading profile: EGFGroup_SJProfile0
[ibm][db2][jcc][sqlj] Driver defaults(user may override): BLOCKING ALL VALIDATE
BIND STATICREADONLY YES
[ibm][db2][jcc][sqlj] Fixed driver options: DATETIME ISO DYNAMICRULES BIND
[ibm][db2][jcc][sqlj] Binding package EGFA1021 at isolation level UR
[ibm][db2][jcc][sqlj] Binding package EGFA1022 at isolation level CS
[ibm][db2][jcc][sqlj] Binding package EGFA1023 at isolation level RS
[ibm][db2][jcc][sqlj] Binding package EGFA1024 at isolation level RR
[ibm][db2][jcc][sqlj] Bind complete for EGFGroup_SJProfile0
[ibm][db2][jcc][sqlj] Begin Bind
[ibm][db2][jcc][sqlj] Loading profile: EGFGrper_SJProfile0
[ibm][db2][jcc][sqlj] Driver defaults(user may override): BLOCKING ALL VALIDATE
BIND STATICREADONLY YES
[ibm][db2][jcc][sqlj] Fixed driver options: DATETIME ISO DYNAMICRULES BIND
[ibm][db2][jcc][sqlj] Binding package EGFB1021 at isolation level UR
[ibm][db2][jcc][sqlj] Binding package EGFB1022 at isolation level CS
[ibm][db2][jcc][sqlj] Binding package EGFB1023 at isolation level RS
[ibm][db2][jcc][sqlj] Binding package EGFB1024 at isolation level RR
[ibm][db2][jcc][sqlj] Bind complete for EGFGrper_SJProfile0
[ibm][db2][jcc][sqlj] Begin Bind
[ibm][db2][jcc][sqlj] Loading profile: EGFJobs_SJProfile0
...
```

When the binds are complete, the CLP window closes, the DB2 Grouper client opens, and the DB2 Grouper launchpad is displayed.

10.3.2 Using the DB2 Grouper launchpad

The DB2 Grouper launchpad is a handy way to learn about the tasks and, more importantly, the sequence of tasks that you must perform for doing DB2 Grouper discovery. Figure 10-23 shows the opening launchpad window.

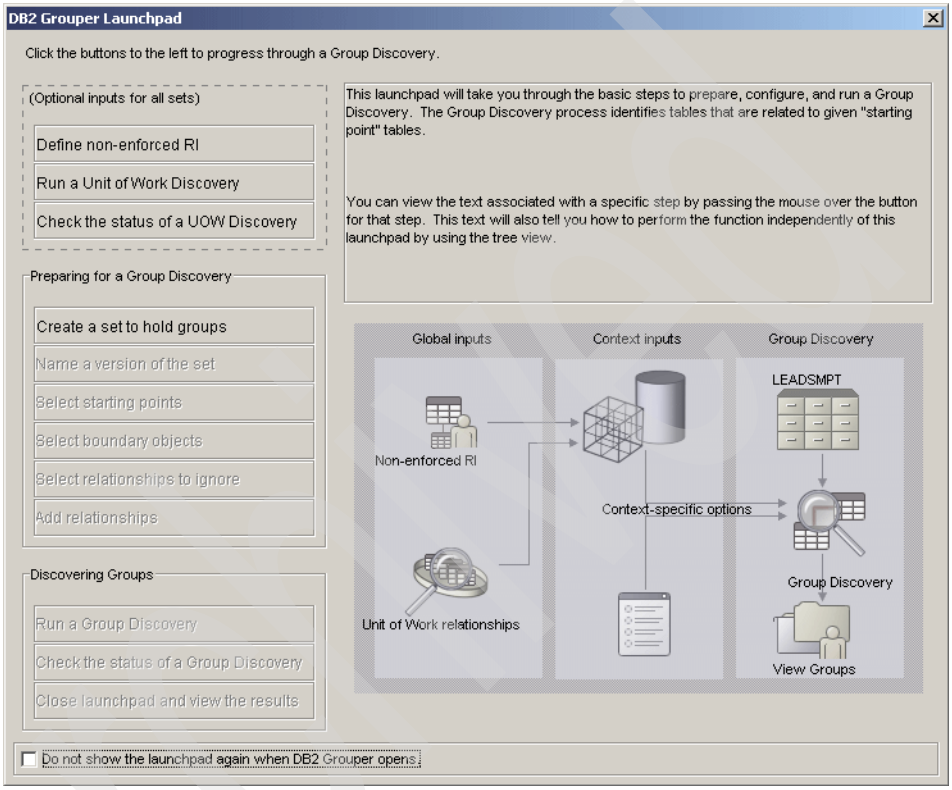


Figure 10-23 DB2 Grouper launchpad opening window

We found that using the launchpad as a guide was helpful in the beginning to become familiar with how to use DB2 Grouper. By default, the launchpad always opens when you start the DB2 Grouper client. You can disable this default by selecting the box at the bottom of the launchpad window. You can always open the launchpad from the DB2 Grouper menu in the client, by selecting **Grouper** → **Launchpad**. You can also roll your mouse over (but not click) each launchpad button on the left, and the explanation for it displays on the right.

One thing the launchpad does is help you to understand the sequence of steps that you must perform to create the group. Because DB2 RE does not yet

support unit-of-work discovered objects, we ignore those steps. The other steps are:

1. Define non-enforced referential integrity relationships, if they exist.
2. Create set to hold groups.
3. Name a version of the set.
4. Select starting points.
5. Select boundary options.
6. Select relationships to ignore.
7. Add other relationships.
8. Run Group Discovery.

Defining non-enforced referential integrity

In this scenario, we need to define the non-enforced referential integrity or our TRADSIMP database tables. These are the tables used by the WebSphere Trade6 sample application. (See 6.6, “Point-in-time recovery with referential integrity and DB2 Grouper” on page 166 for information about the Trade6 application.)

Figure 10-24 on page 329 shows a simplified look at the relationship of the TRADSIMP tables. There is a logical relationship between these tables but the foreign keys are not defined in the table DDL. The WebSphere Trade6 application handles the relationships internally, so they do not use the DB2 referential integrity. This is a common scenario for many customers, so we show how you can still use DB2 Grouper to work with application-enforced referential integrity.

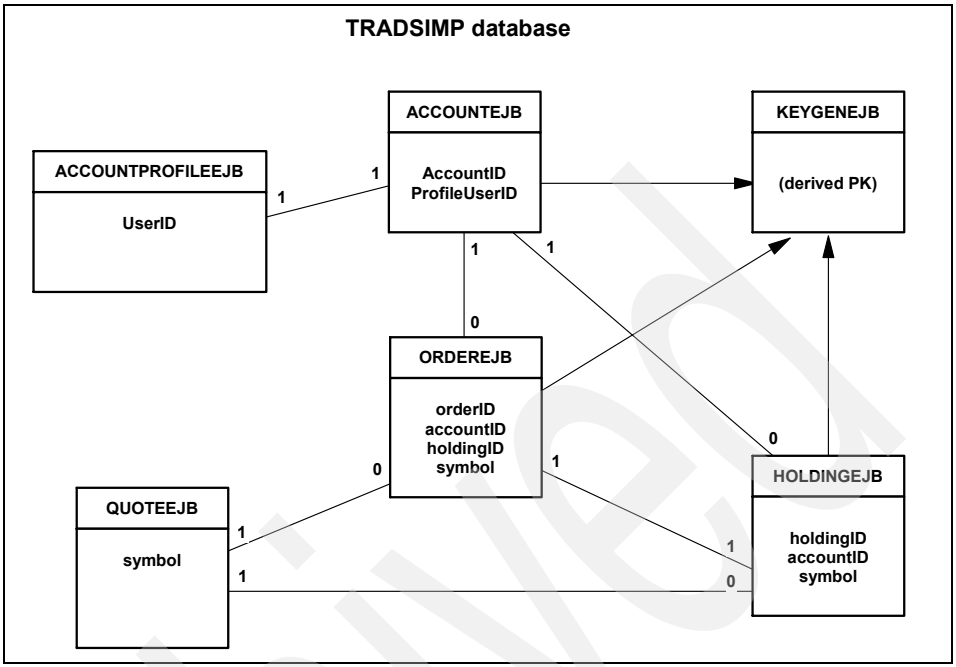


Figure 10-24 TRADSIMP tables

The launchpad description for non-enforced referential integrity (RI) says:

> Optionally, define non-enforced referential constraint relationships (also sometimes called Application-Managed Referential Integrity, or RI). These non-enforced referential constraint relationships are used globally as input for all subsequent Group Discoveries. if you define these relationships, you must define them before running a Group Discovery for them to be considered in finding related tables.

We use the launchpad to drive our relationship definitions. To define relationships, follow these steps:

1. Click **Define non-enforced RI** to launch the wizard.

   Figure 10-25 on page 330 shows the initial window for defining the non-enforced referential integrity. As you can see, it is empty. We identify each table and its relationship to other tables, and this window populates as we proceed.

   You should refer to Figure 10-24 during these steps to see how we define the relationships. We know that each table does have a column which relates, so we need to use DB2 Grouper to identify the parent table and column as well

as its dependent table and column. This identification must be done for each table relationship.

In our case, we have the following relationships to identify:

- Account → Accountprofile
- Account → Order
- Account → Holding
- Quote → Holding
- Quote → Order
- Holding → Order

In all but the Account → Accountprofile relationship, the dependent or child table has the parent's key column named as a column in the table (but not as foreign keys).

2. Click **Add Relationship** to begin adding the first table relationship.

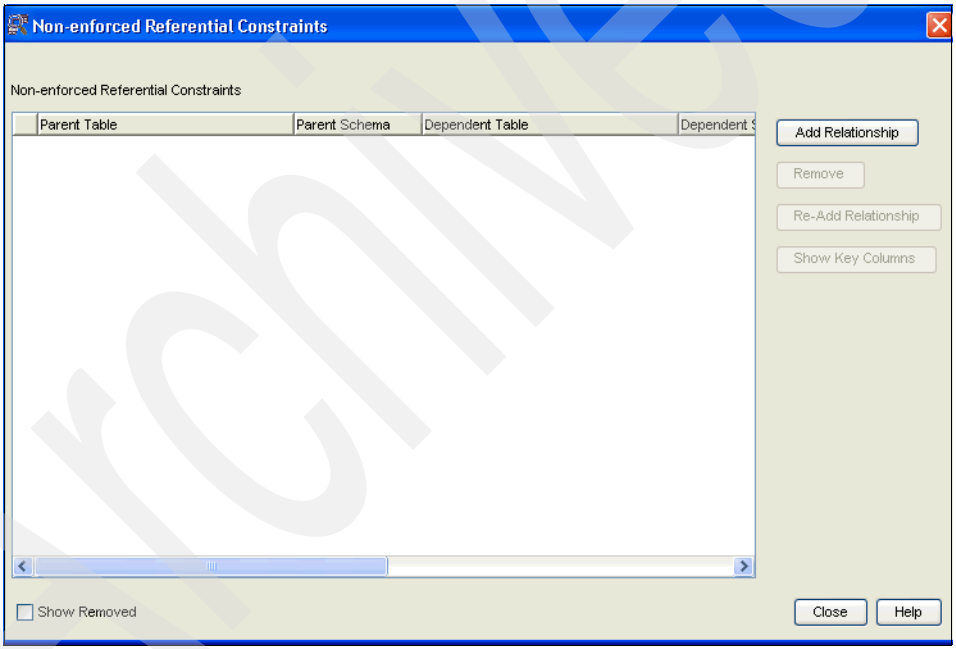


Figure 10-25 Starting window for non-enforced referential integrity

Figure 10-26 shows the first relationship between the ACCOUNTEJB table and the ACCOUNTPROFILEEJB table. You can either type in the table information or use the search dialog. We do not the show search dialog here. It is simple and self-explanatory.

Tip: The schema and table names in this dialog are case sensitive. Usually tables are in uppercase. So, be sure to type in uppercase. If you use the search dialog with a point-and-click selection scheme, the case is correct.

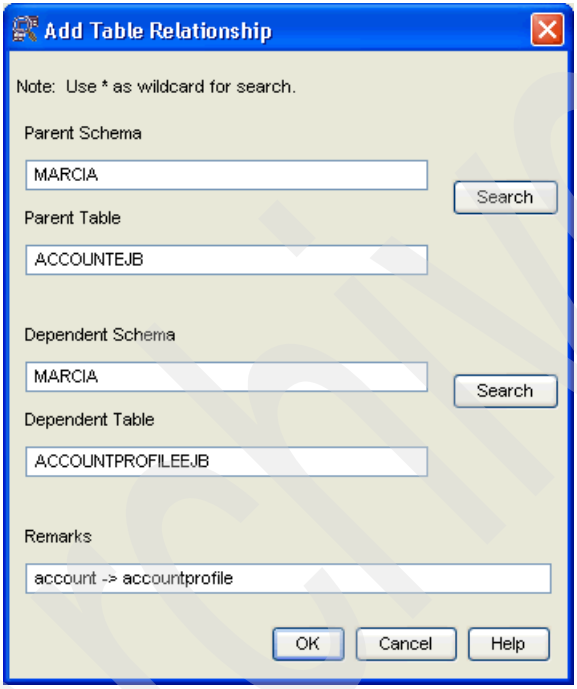


Figure 10-26 Add Table Relationship

3. Click **OK** to proceed with the definition.
4. Identify the columns that participate in the relationship.

   The parent table information is shown in Figure 10-27 on page 332. If there are any unique keys that are defined in the parent table, they are shown on this window. In our example, the ACCOUNTEJB table has a primary key that is defined on the ACCOUNTID column. However, the ACCOUNTID is not a column in the child ACCOUNTPROFILEEJB table. The relationship for these two tables is on the USERID column. We could have defined the relationship in the other direction, with ACCOUNTPROFILEEJB as the parent table, but in this case, we wanted to show both approaches.

You might not want to connect the key value to the child table because there is some other relationship. That is the purpose of the Link any Columns to Dependent button. If you want to work only with the parent's key column, highlight the column on the left side of the window to enable the button titled Link Unique Key to Dependent.

The end result is the same, but you have more choices with the any columns option.

Help! If you are confused about how this works, click **Help** to launch the DB2 Grouper online help. It is very useful for learning DB2 Grouper.

5. In this case, we know the ACCOUNTID column in the ACCOUNTEJB table is not referenced in the ACCOUTPROFILEEJB table. Click **Link Any Columns to Dependent**.

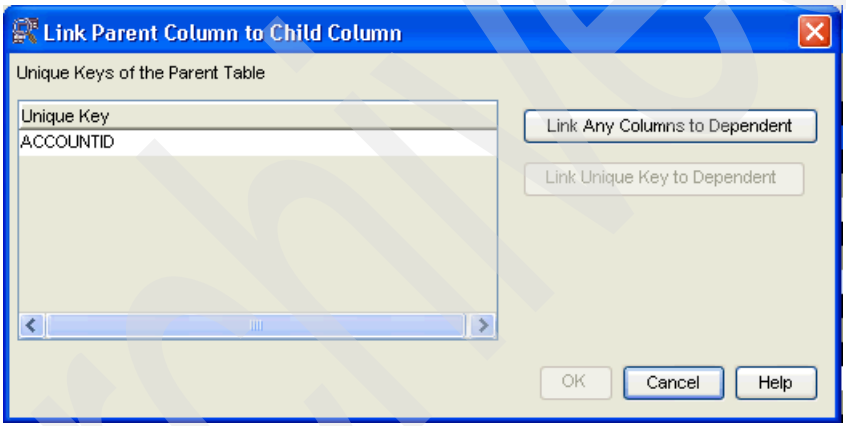


Figure 10-27 Link Parent Column to Child Column

6. Figure 10-28 on page 333 now shows the list of columns that we chose to define the relationship. Choose the parent table column on the left. Choose the related child table column from the list on the right. To enable the list, click the column name.
7. Select **PROFILE_USERID** from the parent list and **USERID** from the child list.
8. Click **OK** to continue.

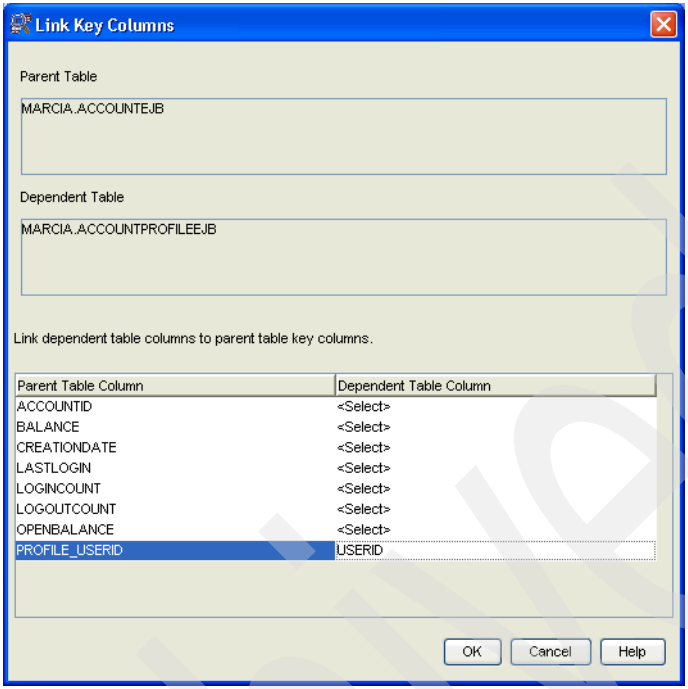


Figure 10-28 Link Key Columns

The first relationship is now defined, as shown in Figure 10-29 on page 334. We need to define the next five relationships in the same way. We do not show all those windows here, but the steps are all performed from the Non-enforced Referential Constraints window. You need to do the following:

1. Open Add Relationship dialog.
2. Identify parent and child tables.
3. Identify related column.

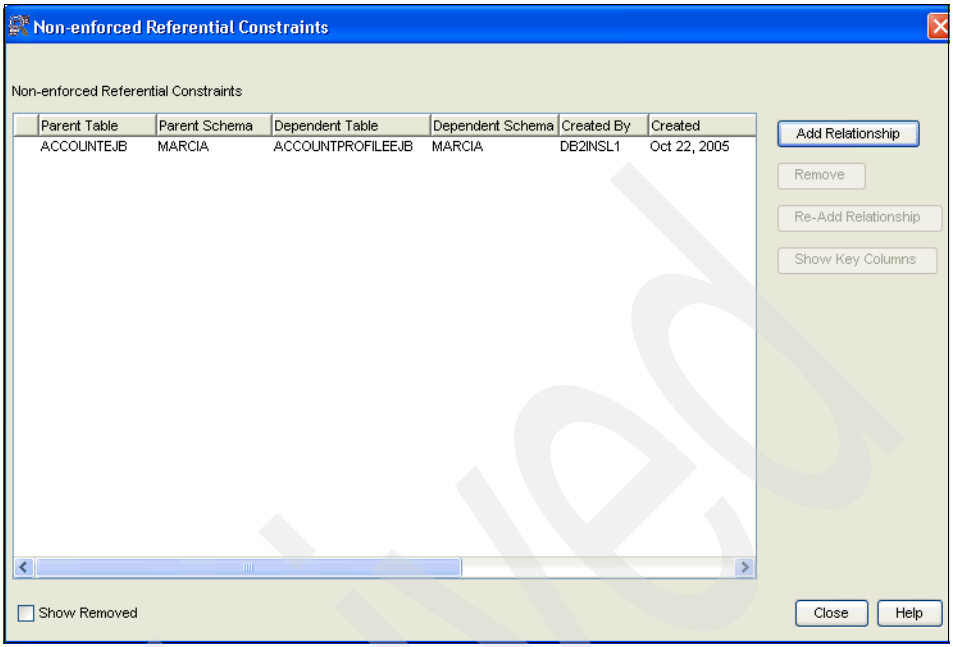


Figure 10-29 First relationship is defined

Figure 10-30 on page 335 shows the final list of related tables. (We will deal with table KEYGENEJB later in this example.)

Now, are we finished? Not yet. We have identified the non-enforced relationships manually. We need to use these relationships as input during the Group Discovery task.

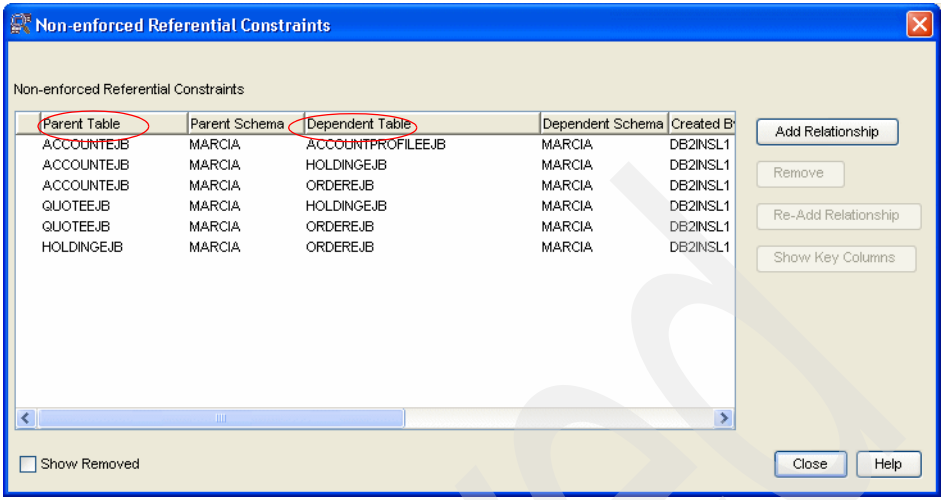


Figure 10-30 Final list of TRADSIMP table relationships

Creating a set to hold groups

Now, that we have identified the non-enforced referential integrity, we can do the remainder of the setup for DB2 Grouper. Next, we create a set to hold the groups.

Note: Just a reminder of the following DB2 Grouper terminology:

- *Group*: A collection of related tables.
- *Version*: A collection of groups.
- *Set*: A collection of one or more versions.

We strongly recommend that you read and understand the glossary terms as described in *IBM DB2 Grouper User's Guide*, SC18-7409.

We are still operating from the launchpad, as shown in Figure 10-23 on page 327. To create a set to hold groups, follow these steps:

1. Click **Create a set to hold groups**. This opens a window as shown in Figure 10-31 on page 336.
2. Enter a name and (optional) description. We determined later when using DB2 RE to recover these tables that the only names that we see in DB2 RE are the Set Name and the Version Name. Thus, you should choose a more meaningful name than the name that we chose in this example.
3. Click **OK** to create the set.

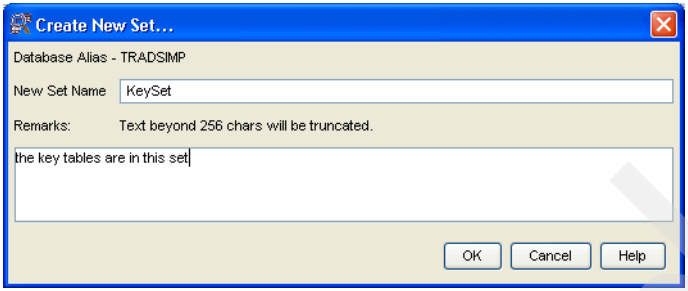


Figure 10-31 Create new set

Naming a version of the set

We are still in the launchpad. When the new set is created, the next launchpad button is enabled for use. In this case, it is the button for naming a version of the set. To name a version of the set, follow these steps:

1. Click **Name a version of the set**.

   You can enter any text name you like for the version name. In our case, again, we chose an unfortunate name of `VERSION1`, which does not really tell us much about the version. You might want to consider naming the version with the date or with a reason you are making the version.

   If you are in a production environment with controlled changes and if you are modifying the DB2 Grouper version due to some structural change in your database, you could use the change control number or ID as part of the version name. These are just some suggestions, but using `VERSION1` is not a very good idea. Choose a name that will make sense to you in the future, not just today.

2. Click **OK** to name the version and to return to the launchpad.

Selecting starting points

After creating the set and starting a version, you can select the starting point for the Group Discovery. DB2 Grouper defines starting point tables as:

> Tables that the user specifies in the group discovery options to limit the scope of the group discovery. For example, specifying tables A and B as starting points will direct the group discovery to find tables that are only related to tables A and B. The group discovery will return a maximum of two groups: One group that contains all of the tables related to table A, and another group that contains all of the tables related to table B. However, if a group discovery finds that any of the tables in the table A group are related to any of the tables in the table B group, then the group discovery will draw those groups together and return only one group. Not specifying any starting points will cause the

group discovery to find all of the tables related to every user table in the DB2 catalog.

We are still in the launchpad. To select starting points, do the following:

1. Click **Select starting points**. This opens the window shown in Figure 10-32.

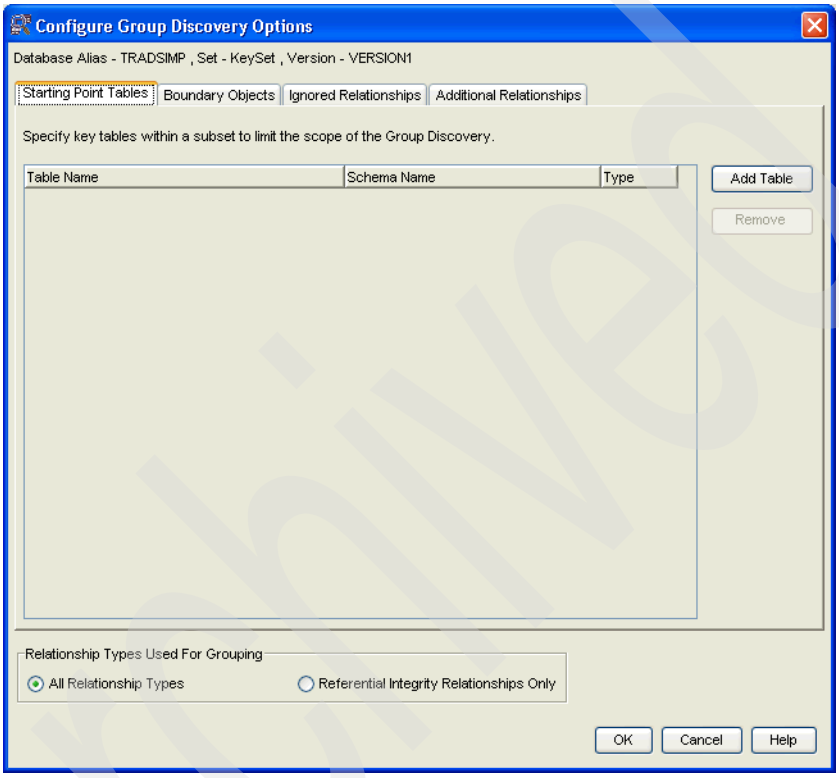


Figure 10-32 Starting Point Tables - initial

2. Click **Add Table** to define the starting point. This opens the window shown in Figure 10-33 on page 338. You can type in the table name (which is case sensitive again) or use a search.
3. Click **OK**.

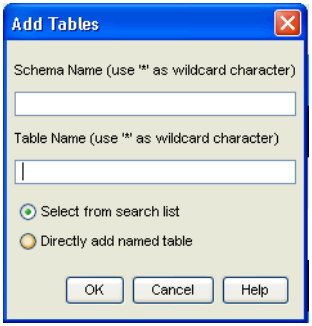


Figure 10-33 Add tables

4. Figure 10-34 show the tables that we can choose as the starting point. We opted to choose all of them except the KEYGENEJB table, as shown in Figure 10-35 on page 339.

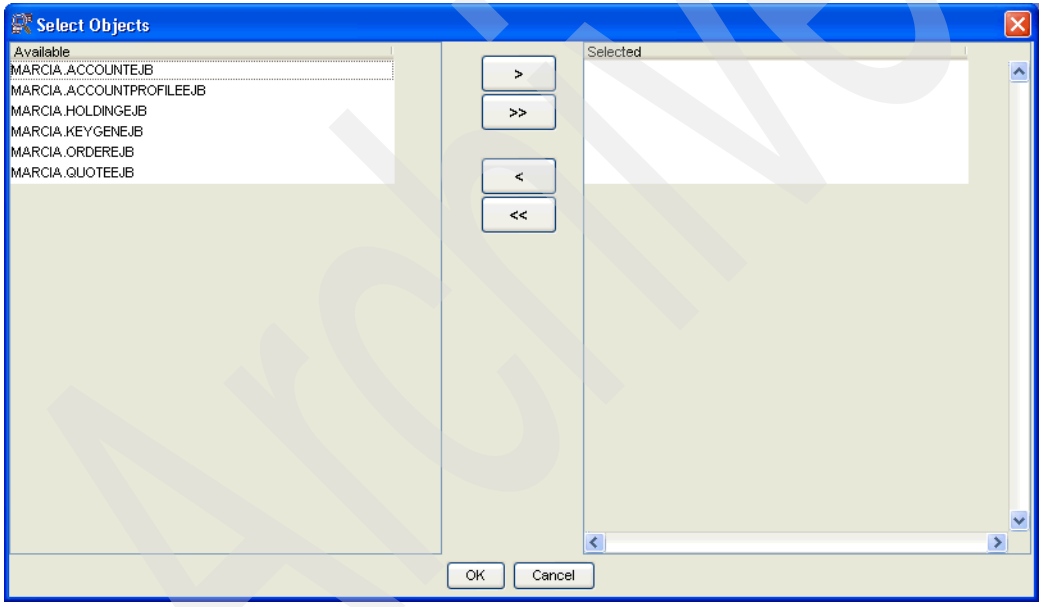


Figure 10-34 Starting Point table selection

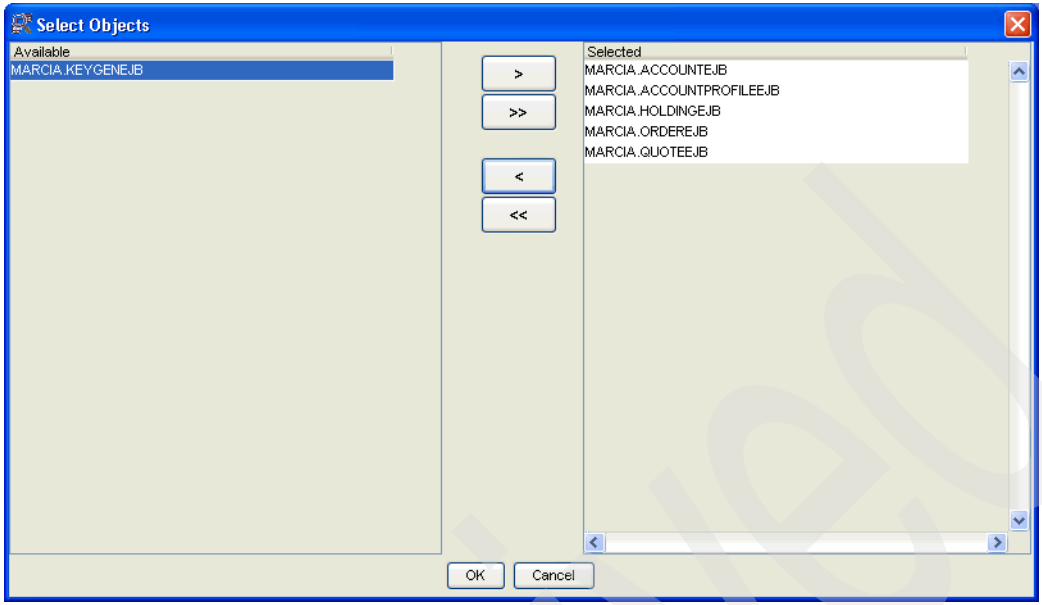


Figure 10-35 Multiple starting point tables chosen

5. Click **OK**.

   When you have defined the starting point table or tables, you are returned to the Group Discovery Options window, with your tables now populated in the window, as shown in Figure 10-36.

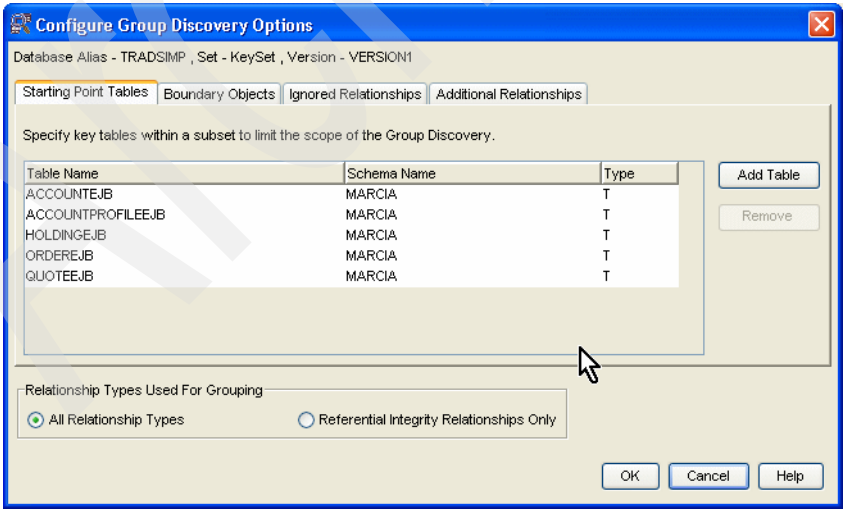


Figure 10-36 Group Discovery Options, Starting Point Tables defined

Selecting boundary objects

At this point you can either return to the launchpad or simply click the next tab on the Group Discovery Options window. Either one takes you to the next step which is to select boundary objects.

You can set certain tables as boundaries for the group discovery so that the traversal will not go too far. In our case, we leave the Boundary Objects tab blank, as shown in Figure 10-37.

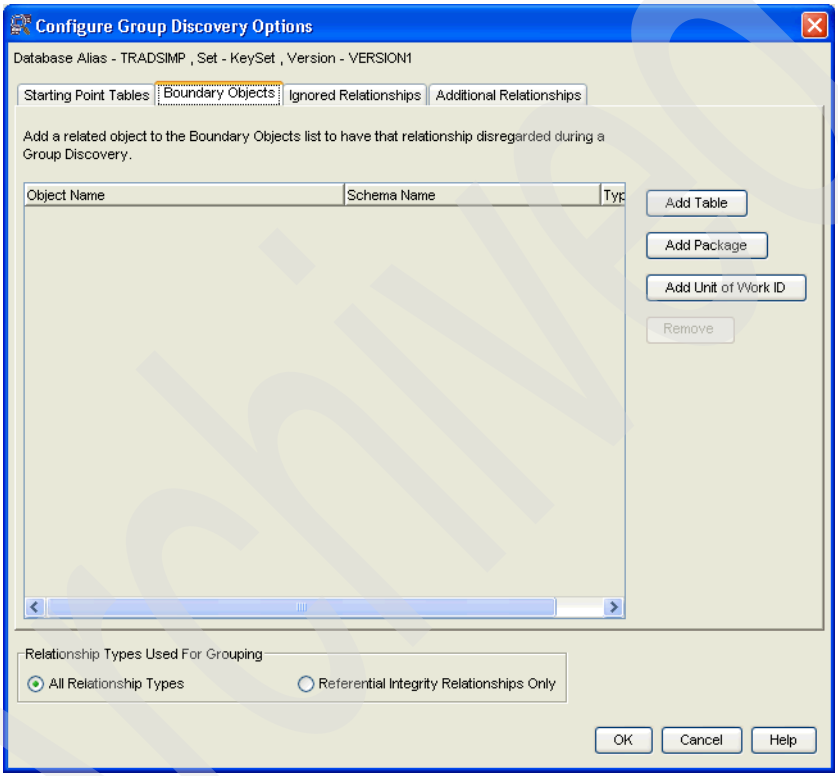


Figure 10-37 Group Discovery Options, Boundary Objects

Selecting relationships to ignore

In the Ignored Relationships tab (Figure 10-38), we do not select anything because we do not want to ignore anything.

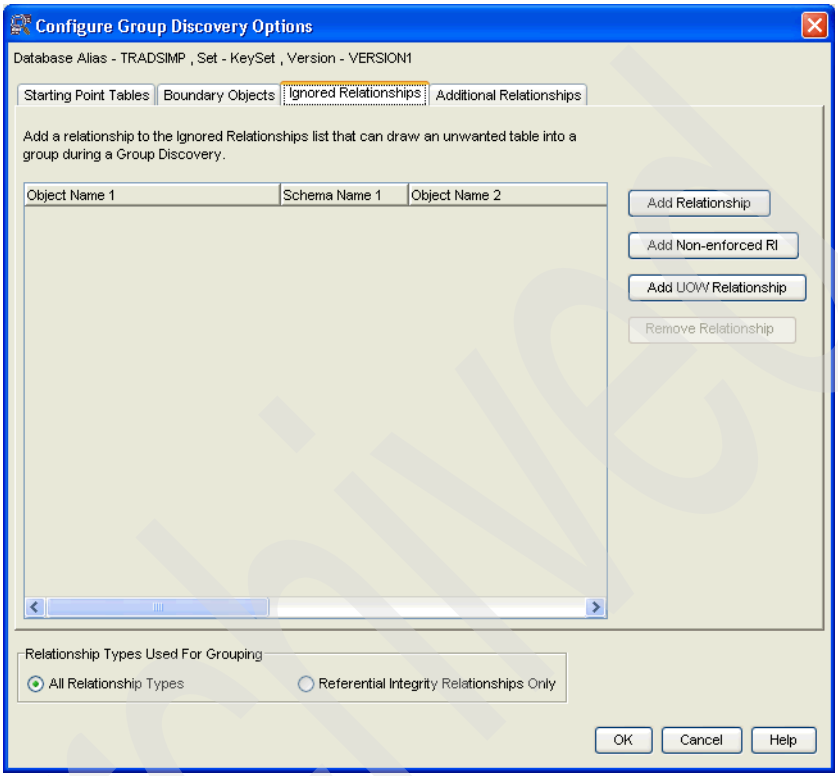


Figure 10-38 Group Discovery Options - Ignore Relationships

Adding other relationships

Our TRADE database has six main tables. We already identified the non-enforce referential integrity relationships among five of them, but there is one more table, KEYGENEJB, that we might want to consider (see Figure 10-24 on page 329). This table is not related to the others by any column name. It is used in the application.

In hindsight, we might have been safe to ignore this table completely because its recovery is not relevant to the other tables. However, we did not know this at the beginning. So, we added it to the list anyway, and we show that in this example.

We can add tables such as this in the Additional Relationships tab. This tab does not require a column relationship to exist. It is more or less a brute-force method of telling DB2 Grouper to include the table in the group.

Figure 10-39 shows the initial page for the Additional Relationships tab. You can either add tables or packages as the related objects. In our case, we want to add the table KEYGENEJB and relate it to the ACCOUNTEJB, HOLDINGEJB, and ORDEREJB tables. To add the table, follow these steps:

1. Click **Add Table Relationship**.

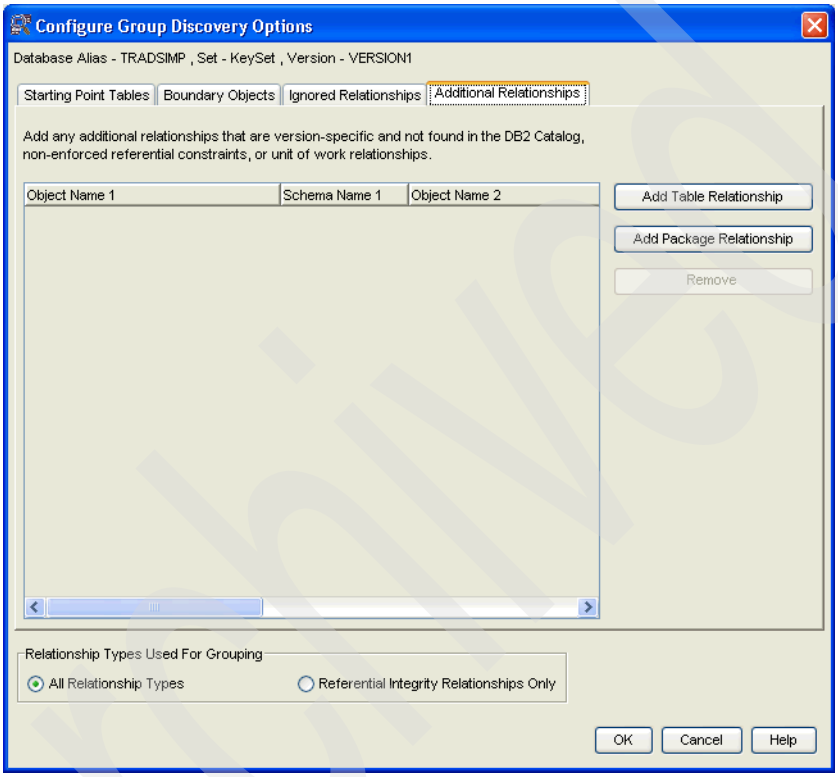


Figure 10-39 Additional Relationships

Figure 10-40 on page 343 shows the Add Table relationship window. Notice that there is no parent and child concept as we saw earlier, because we are not dealing with key columns. We still must define some sort of hierarchy, though, by choosing the appropriate Relationship Direction indicator.

We define three tables connected to KEYGENEJB (not shown here).

2. Press **OK** to return to the Group Discovery Options window.

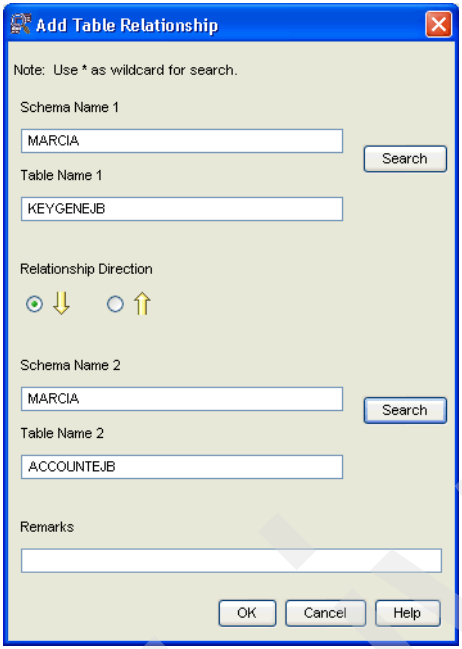


Figure 10-40 Add additional table relationship

Figure 10-41 on page 344 shows the final set of additional related tables.

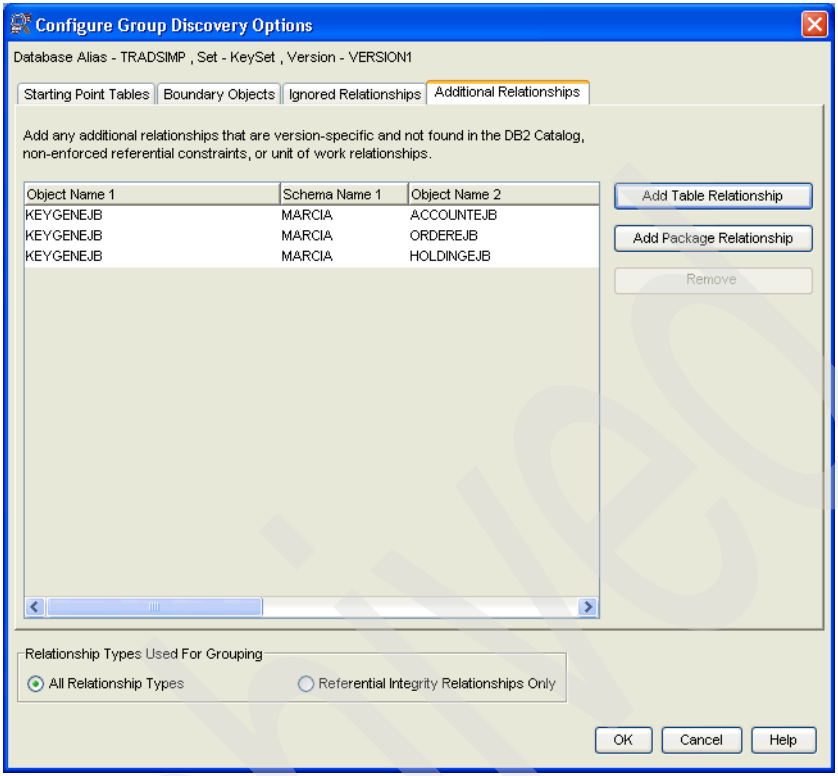


Figure 10-41 Additional Relationships complete

3. Click **OK** to exit the Configure Group Discovery Options dialog.

Now we have done all the necessary configuration steps to prepare for the Group Discovery. In the launchpad, we are now at the bottom set of buttons, in the group box labelled Discovering Groups. These buttons are not enabled until you have completed the preparation steps (which were not yet complete when we captured the window shown in Figure 10-42 on page 345).

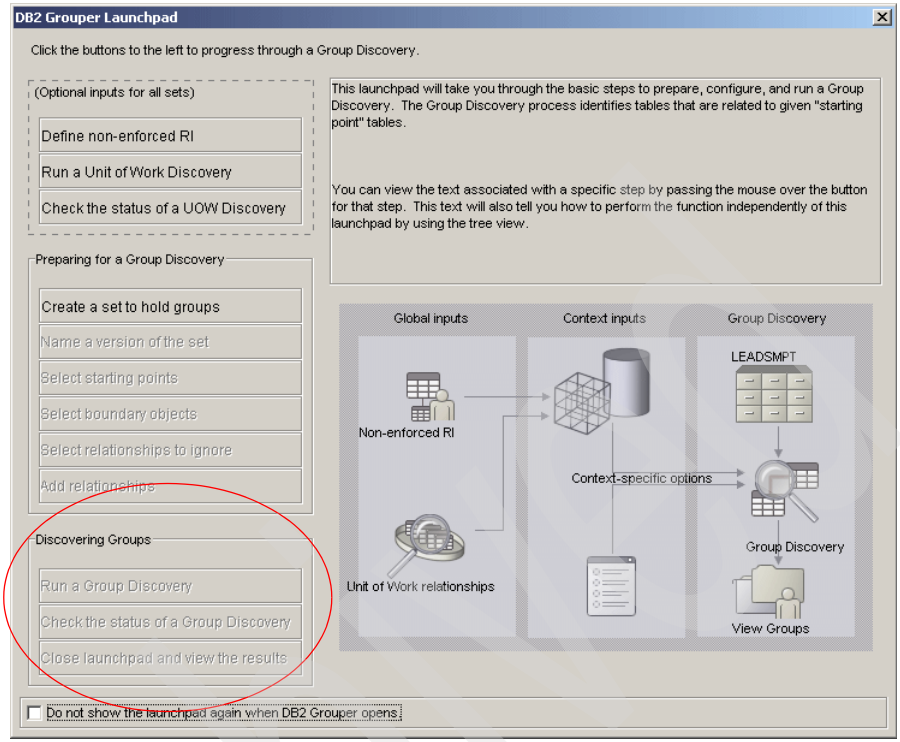


Figure 10-42 DB2 Grouper Launchpad

Running Group Discovery

The Group Discovery is really just the execution of the stored procedure that was set up when configuring the database for DB2 Grouper. To run Group Discovery, follow these steps:

1. Click **Run a Group Discovery**.

   You might be asking what is left to discover in this scenario, because we already defined all the relationships manually. This might be true, but the procedure still needs to run in order to populate the DB2 Grouper tables properly. We show a scenario later which does indeed discover the DB2-enforced referential integrity.

2. Figure 10-43 shows the Run Group Discovery window. Click **OK** to initiate.

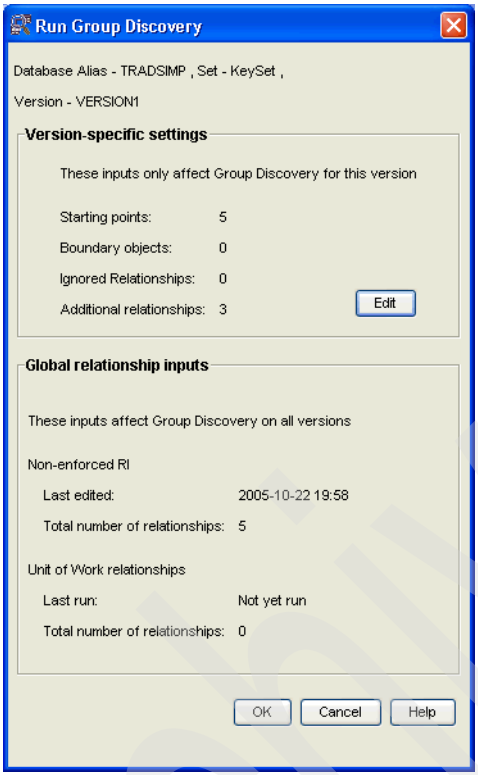


Figure 10-43 Run Group Discovery window

You are prompted to give the job a name and to confirm the Group Discovery, as shown in Figure 10-44.

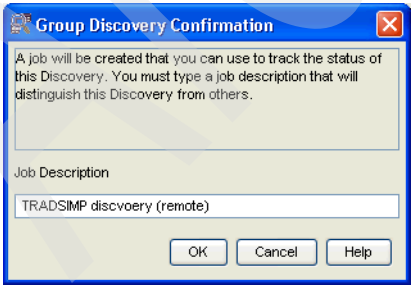


Figure 10-44 Group Discovery Confirmation

3. Enter the description and click **OK**.

4. You see yet another confirmation window, as shown in Figure 10-45. This window displays the job number identifier (not the name that you gave in the previous window). You should remember this number, because you might need to know it later.

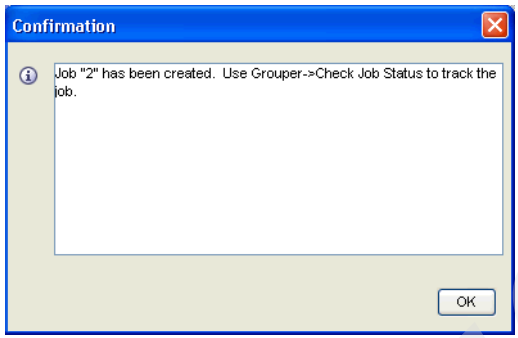


Figure 10-45 Group Discovery job confirmation

The Confirmation window says only that the Group Discovery Job has been *created*, but the job has not yet been *submitted*.

5. Click **OK** to continue with Group Discovery.
6. Figure 10-46 shows the next window. Here, we can see the job number from and some instructions for how to perform the discovery.

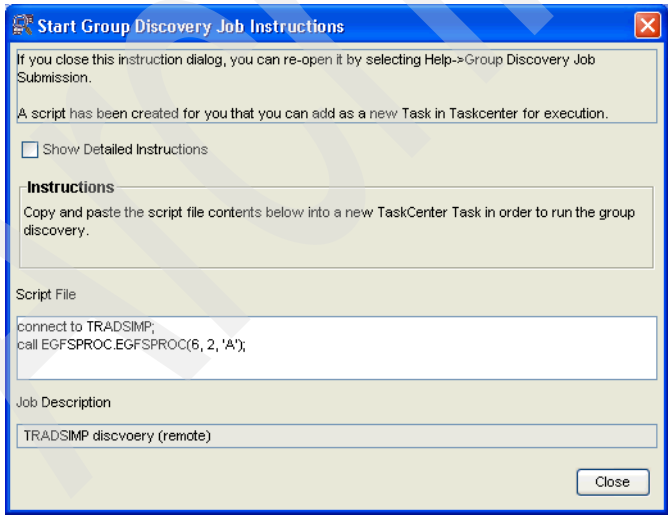


Figure 10-46 Start Group Discovery Job Instructions

You must run the discovery *manually* by executing the stored procedure and by sending it the correct parameters as shown in the window.

7. Select the check box to expand the instructions to see more detail, as shown in Figure 10-47.

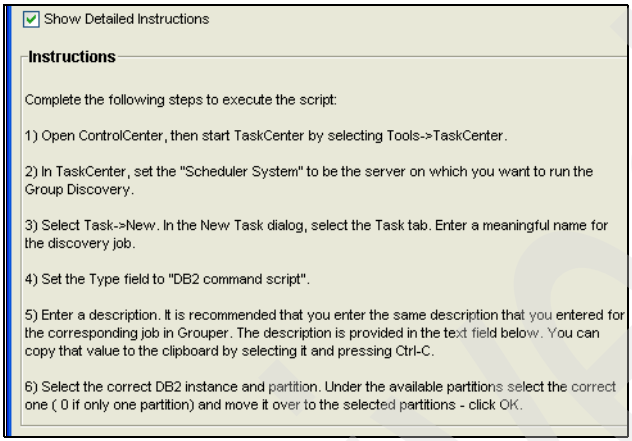


Figure 10-47 Start Group Discovery Job Detailed Instructions

There are several ways that you can run the job. The instructions tell you to create a task in the DB2 Task Center. This method does work and can be useful when you are dealing with a remote server. However, we found there were two simpler ways to accomplish the same task, as follows:

- You can copy the command as instructed on the screen, but you can also use the DB2 Command Editor to execute it. This method is convenient for immediate execution and if you do not need the other features of Task Center.
- You can run the command from the command line or from a script directly on the server. This method requires that you can open a command prompt on the server via telnet for example. You must connect to the database, and you must first make the DB2 Grouper stored procedure directory to be the current directory so that the stored procedure can find the jar file.

Regardless of the method that you use, the objective is to execute the stored procedure using the parameters as displayed in the Job Instructions window.

8. After you execute the stored procedure, you can check the job status from the DB2 Grouper client. You can do this from the DB2 Grouper launchpad or from the DB2 Grouper menu by clicking **Grouper** → **Check Job Status**. This action opens a window such as the one shown in Figure 10-48 on page 349, where you can see the various discovery jobs that were run.

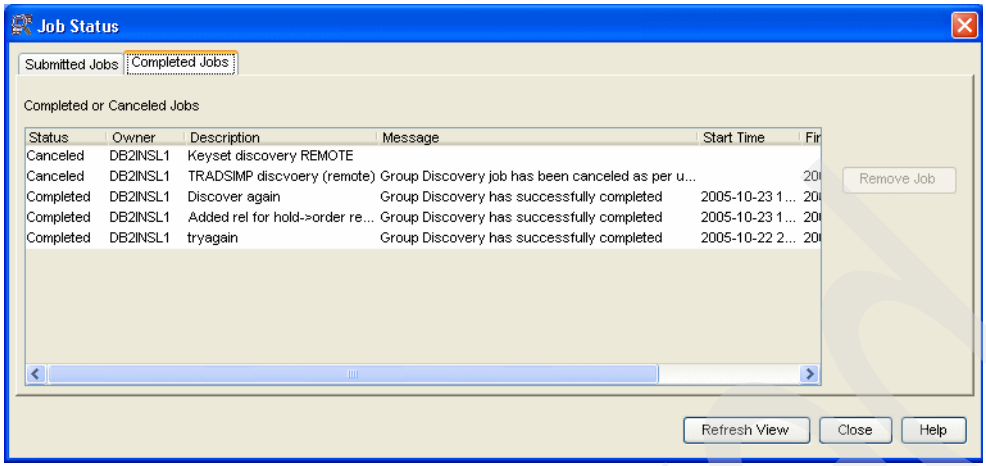


Figure 10-48 Check discovery job status

9. When your job is completed with a successful result, you can go to the DB2 Grouper client and look at the output. Close the launchpad if it is still open.
10. In the client, you can select the version on the left panel, then use the menu to select **Selected** → **View Group Discovery Results**. Alternatively, you can expand the Version and the new-found group should display. Figure 10-49 on page 350 shows that DB2 Grouper has identified the relationships correctly and has included them in the set named KeySet, the Version named Version1, and the group named Group_1.
11. The Group name, Group_1, was created automatically by DB2 Grouper. You can rename the group by right-clicking the group and opening the Properties window.

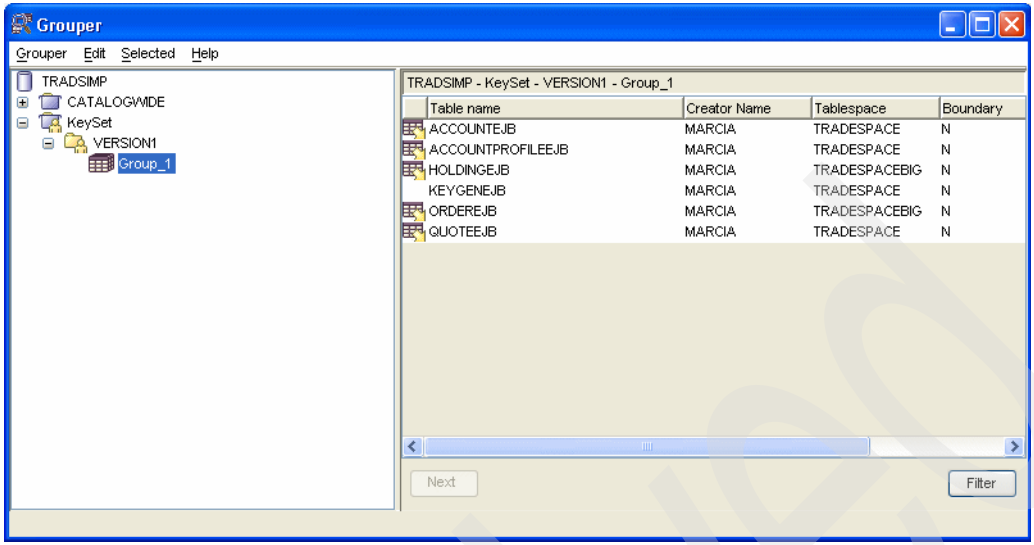


Figure 10-49 DB2 Grouper found relationships

12. Now, we almost have a group that we can use for DB2 RE. The last step is to mark the particular version as available. To do this, on the Version, right-click and select **Make available**.

    When you do this, DB2 Grouper marks it internally so that DB2 RE can see it, and DB2 Grouper also marks the version visually by adding a green triangle to the version icon, as shown in Figure 10-50.

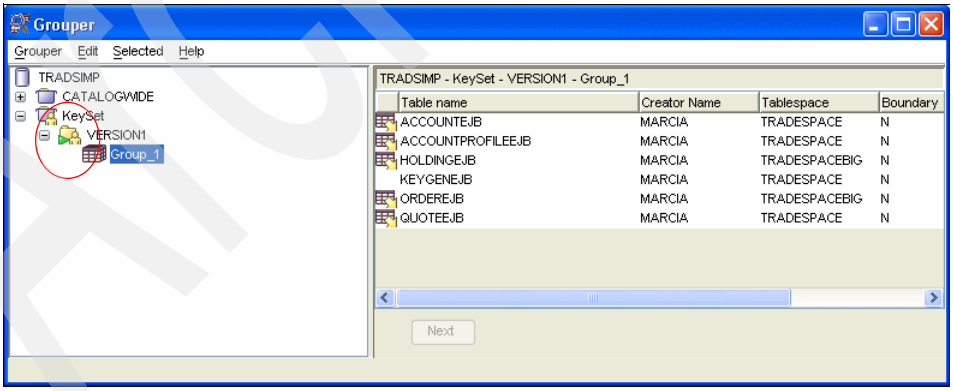


Figure 10-50 Group is now available for DB2 RE

When you are first learning DB2 Grouper, it is quite common to forget the last step of making the version available. When you use DB2 RE, if you click **Show Relationships** for some object and you do not see the relationships,

you should first check to see that the version is marked available in DB2 Grouper. If you forget to make it available, you do not have to re-run the discovery to make it available. Just open DB2 Grouper and make the version available.

The DB2 Grouper set that we created in this example is used in a recovery scenario in 6.6, “Point-in-time recovery with referential integrity and DB2 Grouper” on page 166.

10.3.3 Creating DB2 Grouper sets for DB2-enforced referential integrity

This section shows a simpler example as that of the previous section where we allow DB2 Grouper to find all the relationships in the database, given very few parameters. This time, we re-run a Group Discovery using the DB2 Grouper sample tables. These tables, and their referential integrity relationships, are documented in *IBM DB2 Grouper User's Guide*, SC18-7409.

We do not show a figure for every step here, only a few. The steps are:

1. Open DB2 Grouper.
2. Create a set. We named ours RedbookSet.
3. Create a version. We named ours ITSO_Version1.
4. Set the Group Discovery Options with three starting point tables, see Figure 10-51 on page 352.

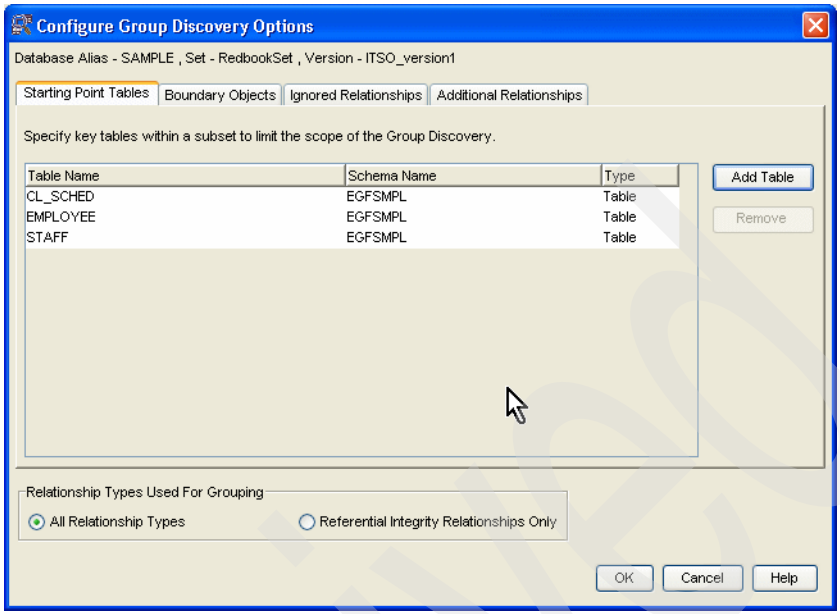


Figure 10-51 Set starting point tables

5. Run Group Discovery, and note the Job Start Instructions (Figure 10-52).

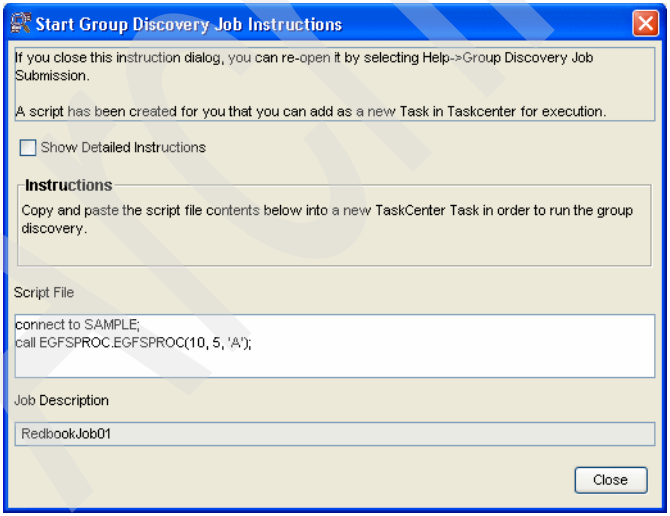


Figure 10-52 Group Discovery Job Instructions

6. Run the discovery job from the command line, see Example 10-4.

Example 10-4 Run Group Discovery from command line

```
C:\DB2Tools\grouper\server\sp\db2v8>db2 connect to sample

   Database Connection Information

 Database server        = DB2/NT 8.2.1
 SQL authorization ID   = MARCIA
 Local database alias   = SAMPLE

C:\DB2Tools\grouper\server\sp\db2v8>db2 call EGFSPROC.EGFSPROC(10, 5, 'A')

  Return Status = 0

C:\DB2Tools\grouper\server\sp\db2v8>
```

7. Check Job Status (optional), see Figure 10-53.

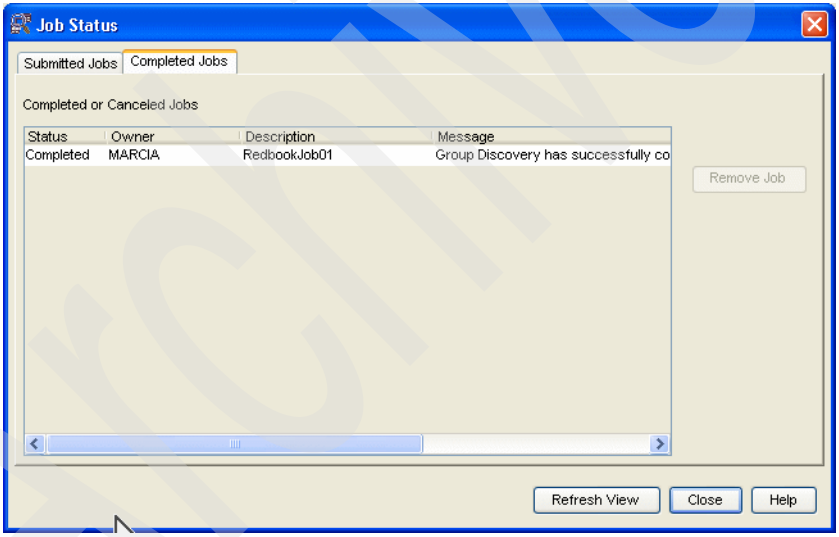


Figure 10-53 Check Discovery Job Status

8. See the groups that were captured during job (Figure 10-54).

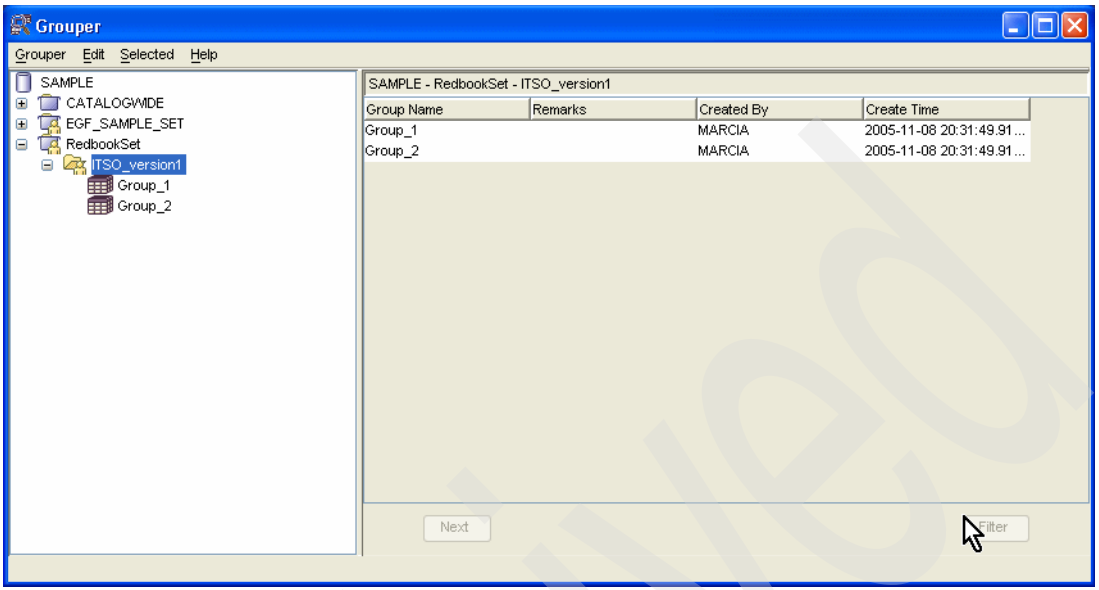


Figure 10-54 Group Discovery created two groups

Figure 10-55 and Figure 10-56 on page 355 show the contents of the two groups.

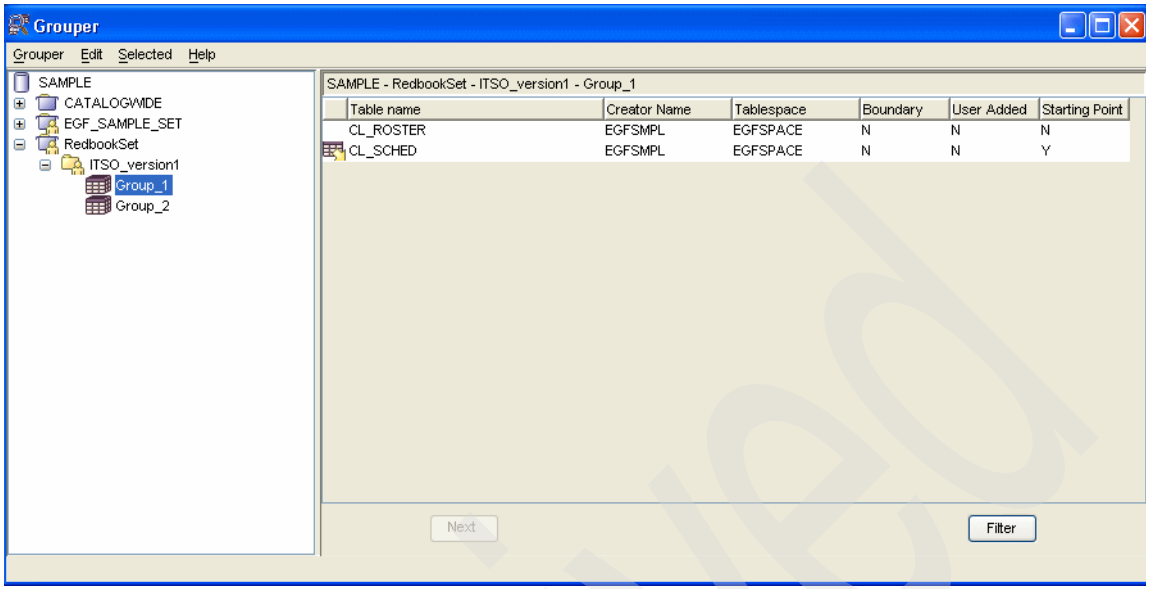


Figure 10-55 RedbookSet Group_1

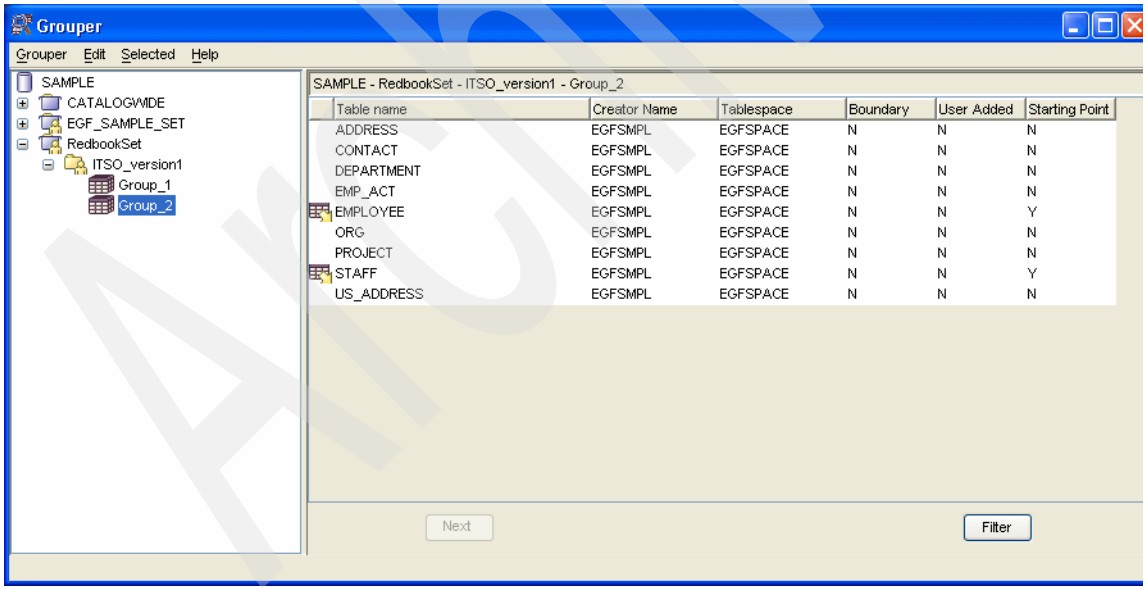


Figure 10-56 RedbookSet Group_2

We do not show a recovery scenario with these groups here. We leave that as an exercise for you.

A

Sample applications

This appendix lists the scripts that we used to create table space and tables that we used in the Fast Backup scenarios for this book.

A.1 Fast Backup sample applications

We used the following sample applications in the Fast Backup scenarios in this book.

A.1.1 The DEMOSPACE application

DEMOSPACE table space is generated by the following scripts.

- makesms.sh
- makedms.sh
- demospace.sh
- demospace.ddl

makesms.sh

Example A-1 shows the script which creates a system managed table space DEMOSPACE.

Example: A-1 makesms.sh

```
#!/bin/ksh
# login as instance owner (sanjose1: db2inst1)

# create new tablespace on the ESS /ess2 mountpoint
db2 connect to sample
db2 drop tablespace demospace > /dev/null
echo "creating a tablesspace DEMOSPACE ..."
db2 "create regular tablespace demospace pagesize 4096 managed by system using
('/ess2/demosms')"
db2 connect reset
```

makedms.sh

Example A-2 shows the script which creates a database managed table space DEMOSPACE.

Example: A-2 makedms.sh

```
#!/bin/ksh
# login as instance owner (sanjose1: db2inst1)

# create new tablespace on the ESS /ess2 mountpoint
db2 connect to sample
db2 drop tablespace demospace > /dev/null
echo "creating a tablespace DEMOSPACE ..."
db2 "create regular tablespace demospace pagesize 4096 managed by database
using (file '/ess2/demodms' 400000)"
db2 connect reset
```

demospace.sh

Example A-3 shows the script which exports and imports tables to DEMOSPACE table space.

Example: A-3 demospace.sh

```
#!/bin/ksh
# login as instance owner (sanjose1: db2inst1)

# exports all the tables to IXF files
db2 connect to sample
db2 "export to  cl_sched.ixf    of ixf  select * from cl_sched    "  >> export.out
db2 "export to  department.ixf  of ixf  select * from DEPARTMENT  "  >> export.out
db2 "export to  emp_act.ixf     of ixf  select * from emp_act     "  >> export.out
db2 "export to  emp_photo.ixf   of ixf modified by lobsinfile   select * from emp_photo" >>
export.out
db2 "export to  emp_resume.ixf  of ixf  select * from emp_resume  "  >> export.out
db2 "export to  employee.ixf    of ixf  select * from employee    "  >> export.out
db2 "export to  in_tray.ixf     of ixf  select * from in_tray     "  >> export.out
db2 "export to  org.ixf         of ixf  select * from org         "  >> export.out
db2 "export to  project.ixf     of ixf  select * from project     "  >> export.out
db2 "export to  sales.ixf       of ixf  select * from sales       "  >> export.out
db2 "export to  staff.ixf       of ixf  select * from staff       "  >> export.out
db2 connect reset

# drop the SAMPLE tables and re-create them in that new tablespace.
db2 -tvf demospace.ddl > demospace.out

# imports the data from step1 back into the new tables.
db2 connect to sample
db2 "import from cl_sched.ixf    of ixf  messages importmsg.txt insert into  cl_sched"
db2 "import from department.ixf  of ixf  messages importmsg.txt insert into  DEPARTMENT"
db2 "import from emp_act.ixf     of ixf  messages importmsg.txt insert into  emp_act"
db2 "import from emp_photo.ixf of ixf modified by lobsinfile messages importmsg.txt insert into
emp_photo"
db2 "import from emp_resume.ixf  of ixf  messages importmsg.txt insert into  emp_resume"
db2 "import from employee.ixf    of ixf  messages importmsg.txt insert into  employee"
db2 "import from in_tray.ixf     of ixf  messages importmsg.txt insert into  in_tray"
db2 "import from org.ixf         of ixf  messages importmsg.txt insert into  org"
db2 "import from project.ixf     of ixf  messages importmsg.txt insert into  project"
db2 "import from sales.ixf       of ixf  messages importmsg.txt insert into  sales"
db2 "import from staff.ixf       of ixf  messages importmsg.txt insert into  staff"
db2 connect reset

db2 list db directory

echo "*** create Versioning Repository of Recovery Expert, ***"
echo "*** then set logretain recovery and backup SAMPLE.   ***"
```

demospace.ddl

Example A-4 shows the DDL script which create tables on DEMOSPACE table space.

Example: A-4 demospace.ddl

```
-- This CLP file was created using DB2LOOK Version 8.2
-- Timestamp: Thu Sep 29 13:13:35 CDT 2005
-- Database Name: SAMPLE
-- Database Manager Version: DB2/6000 Version 8.2.2
-- Database Codepage: 819
-- Database Collating Sequence is: UNIQUE

CONNECT TO SAMPLE;

DROP TABLE CL_SCHED;
DROP table department;
drop table emp_act;
drop table emp_photo;
drop table emp_resume;
drop table employee;
drop table in_tray;
drop table org;
drop table project;
drop table sales;
drop table staff;

------------------------------------------------
-- DDL Statements for table ORG
------------------------------------------------

 CREATE TABLE ORG  (
                   DEPTNUMB SMALLINT NOT NULL ,
                   DEPTNAME VARCHAR(14) ,
                   MANAGER SMALLINT ,
                   DIVISION VARCHAR(10) ,
                   LOCATION VARCHAR(13) )
                  IN DEMOSPACE ;

-- DDL Statements for primary key on Table ORG

ALTER TABLE ORG
        ADD PRIMARY KEY
                (DEPTNUMB);
```

```
------------------------------------------------
-- DDL Statements for table STAFF
------------------------------------------------

 CREATE TABLE STAFF  (
                  ID SMALLINT NOT NULL ,
                  NAME VARCHAR(9) ,
                  DEPT SMALLINT ,
                  JOB CHAR(5) ,
                  YEARS SMALLINT ,
                  SALARY DECIMAL(7,2) ,
                  COMM DECIMAL(7,2) )
                 IN DEMOSPACE ;

-- DDL Statements for primary key on Table STAFF

ALTER TABLE STAFF
        ADD PRIMARY KEY
                (ID);

------------------------------------------------
-- DDL Statements for table DEPARTMENT
------------------------------------------------

 CREATE TABLE DEPARTMENT  (
                  DEPTNO CHAR(3) NOT NULL ,
                  DEPTNAME VARCHAR(29) NOT NULL ,
                  MGRNO CHAR(6) ,
                  ADMRDEPT CHAR(3) NOT NULL ,
                  LOCATION CHAR(16) )
                 IN DEMOSPACE ;

-- DDL Statements for primary key on Table DEPARTMENT

ALTER TABLE DEPARTMENT
        ADD PRIMARY KEY
                (DEPTNO);

------------------------------------------------
-- DDL Statements for table EMPLOYEE
------------------------------------------------

 CREATE TABLE EMPLOYEE  (
                  EMPNO CHAR(6) NOT NULL ,
                  FIRSTNME VARCHAR(12) NOT NULL ,
                  MIDINIT CHAR(1) NOT NULL ,
                  LASTNAME VARCHAR(15) NOT NULL ,
                  WORKDEPT CHAR(3) ,
                  PHONENO CHAR(4) ,
```

```
                  HIREDATE DATE ,
                  JOB CHAR(8) ,
                  EDLEVEL SMALLINT NOT NULL ,
                  SEX CHAR(1) ,
                  BIRTHDATE DATE ,
                  SALARY DECIMAL(9,2) ,
                  BONUS DECIMAL(9,2) ,
                  COMM DECIMAL(9,2) )
                 IN DEMOSPACE ;

-- DDL Statements for primary key on Table EMPLOYEE

ALTER TABLE EMPLOYEE
        ADD PRIMARY KEY
                (EMPNO);

------------------------------------------------
-- DDL Statements for table EMP_ACT
------------------------------------------------

 CREATE TABLE EMP_ACT  (
                  EMPNO CHAR(6) NOT NULL ,
                  PROJNO CHAR(6) NOT NULL ,
                  ACTNO SMALLINT NOT NULL ,
                  EMPTIME DECIMAL(5,2) ,
                  EMSTDATE DATE ,
                  EMENDATE DATE )
                 IN DEMOSPACE ;

------------------------------------------------
-- DDL Statements for table PROJECT
------------------------------------------------

 CREATE TABLE PROJECT  (
                  PROJNO CHAR(6) NOT NULL ,
                  PROJNAME VARCHAR(24) NOT NULL ,
                  DEPTNO CHAR(3) NOT NULL ,
                  RESPEMP CHAR(6) NOT NULL ,
                  PRSTAFF DECIMAL(5,2) ,
                  PRSTDATE DATE ,
                  PRENDATE DATE ,
                  MAJPROJ CHAR(6) )
                 IN DEMOSPACE ;

-- DDL Statements for primary key on Table PROJECT

ALTER TABLE PROJECT
        ADD PRIMARY KEY
                (PROJNO);
```

```
------------------------------------------------
-- DDL Statements for table EMP_PHOTO
------------------------------------------------

 CREATE TABLE EMP_PHOTO  (
                  EMPNO CHAR(6) NOT NULL ,
                  PHOTO_FORMAT VARCHAR(10) NOT NULL ,
                  PICTURE BLOB(102400) LOGGED NOT COMPACT )
                 IN DEMOSPACE ;

-- DDL Statements for primary key on Table EMP_PHOTO

ALTER TABLE EMP_PHOTO
        ADD PRIMARY KEY
                (EMPNO,
                 PHOTO_FORMAT);

------------------------------------------------
-- DDL Statements for table EMP_RESUME
------------------------------------------------

 CREATE TABLE EMP_RESUME  (
                  EMPNO CHAR(6) NOT NULL ,
                  RESUME_FORMAT VARCHAR(10) NOT NULL ,
                  RESUME CLOB(5120) LOGGED NOT COMPACT )
                 IN DEMOSPACE ;

-- DDL Statements for primary key on Table EMP_RESUME

ALTER TABLE EMP_RESUME
        ADD PRIMARY KEY
                (EMPNO,
                 RESUME_FORMAT);

------------------------------------------------
-- DDL Statements for table SALES
------------------------------------------------

 CREATE TABLE SALES  (
                  SALES_DATE DATE ,
                  SALES_PERSON VARCHAR(15) ,
                  REGION VARCHAR(15) ,
                  SALES INTEGER )
                 IN DEMOSPACE ;
```

```
------------------------------------------------
-- DDL Statements for table CL_SCHED
------------------------------------------------

 CREATE TABLE CL_SCHED  (
                  CLASS_CODE CHAR(7) ,
                  DAY SMALLINT ,
                  STARTING TIME ,
                  ENDING TIME )
                 IN DEMOSPACE ;

------------------------------------------------
-- DDL Statements for table IN_TRAY
------------------------------------------------

 CREATE TABLE IN_TRAY  (
                  RECEIVED TIMESTAMP ,
                  SOURCE CHAR(8) ,
                  SUBJECT CHAR(64) ,
                  NOTE_TEXT VARCHAR(3000) )
                 IN DEMOSPACE ;

COMMIT WORK;

CONNECT RESET;

TERMINATE;
```

B

Log Analysis reports

This appendix provides excerpts of the Log Analysis reports that are produced in Chapter 5, “Log analysis” on page 83.

B.1 Excerpt from general report created by db2la

Example B-1 is an excerpt from the general report that was created by the **db21a** command.

Example: B-1 Excerpt from the general report

```
********************************* TOP OF DATA ********************************
                      DB2 LOG ANALYSIS - GENERAL REPORT
                      BUILT ON 2005/09/29 AT 16:32:11
                      *********************************

FILTERS
-------
DATABASE   : VPS
START DATE : 2005/09/29
START TIME : 00:00:00
END DATE   : 2005/09/30
END TIME   : 00:00:00
ACTION     : U I D
OPTIONS    :

================================================================
URID         LSN          DATE       TIME     STATUS   AUTHID
------------ ------------ ---------- -------- -------- ----------
000000004435 000009C51C90 2005/09/29 13:49:50 COMPLETE DB2INSK1
----------------------------------------------------------------

NGNAME           TSNAME             TABLE OWNER  TABLE NAME         ACTION
LSN
---------------- ------------------ ------------ ------------------ ------
------------
IBMCATGROUP      SYSCATSPACE        SYSIBM       SYSTABLES          U
000009C50A4D
IBMCATGROUP      SYSCATSPACE        SYSIBM       SYSTABLES          U
000009C5105C
IBMCATGROUP      SYSCATSPACE        SYSIBM       SYSTABLES          U
000009C51B18

================================================================
URID         LSN          DATE       TIME     STATUS   AUTHID
------------ ------------ ---------- -------- -------- ----------
0000000044BA 00000ABEE40E 2005/09/29 16:07:35 COMPLETE DB2INSK1
----------------------------------------------------------------

NGNAME           TSNAME             TABLE OWNER  TABLE NAME         ACTION
LSN
---------------- ------------------ ------------ ------------------ ------
------------
```

```
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE0038
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE00FC
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE01BE
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE0286
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE0345
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE131D
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE13E1
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE14A1
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE1566
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE162C
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE16DC
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE17A3
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE1869
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE192F
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE19F5
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE1AA5
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE1B69
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE1C1B
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE1CDC
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE1DA4
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR               D
00000ABE1DA4
<< .... Cut out middle of the report >>
SUMMARY
-------


NGNAME                     UPDATES INSERTS DELETES
-------------------------- ------- ------- -------
```

```
IBMCATGROUP                     3
IBMDEFAULTGROUP                                 303

TSNAME                    UPDATES INSERTS DELETES
------------------------- ------- ------- -------
SMSDAT1                                         303
SYSCATSPACE                     3

TABLE                     UPDATES INSERTS DELETES
------------------------- ------- ------- -------
SYSIBM.SYSTABLES                3
VPS.CAR                                         303

TOTAL SUMMARY
-------------

TOTAL UPDATES : 3
TOTAL INSERTS : 0
TOTAL DELETES : 303
****************************** BOTTOM OF DATA
*******************************
```

B.2 Excerpt of detailed report created by db2la

Example B-2 is an excerpt from the detailed report that was created by the **db2la** command.

Example: B-2 Excerpt from the detailed report

```
DB2 LOG ANALYSIS - DETAILED REPORT
                    BUILT ON 2005/09/29 AT 16:41:07
                    ********************************
                 BASED ON PREVIOUSLY BUILT GENERAL REPORT

ADDITIONAL FILTERS
------------------
DATABASE   : VPS
START DATE : 2005/09/29
START TIME : 00:00:00
END DATE   : 2005/09/30
END TIME   : 00:00:00
ACTION     : U I D
TABLENAME  : CAR
OPTIONS    :

--------------------------------------------------------------------------------
```

```
ACTION DATE       TIME     URID         LSN          STATUS   AUTHID
------ ---------- -------- ------------ ------------ -------- ----------
DELETE 2005/09/29 16:07:35 0000000044BA 00000ABE0038 COMPLETE DB2INSK1

NGNAME             TSNAME             TABLE OWNER  TABLE NAME
------------------ ------------------ ------------ --------------------
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR

ROW STATUS  T_ID T_TYPE T_MODEL T_YEAR T_COLOR T_PACKAGE   T_AC T_POWERWINDOWS
T_AUTOMATIC T_DOORS T_PRICE  T_DATA
----------- ---- ------ ------- ------ ------- ----------- ---- --------------
----------- ------- -------- ----------------
POST-CHANGE -    -      -       -      -       -           -    -
-           -       -        -
PRE-CHANGE  4440 Sedan  Boran   1973   Green   Performance N    Y
N           3       62843.00 Needs a tune up.

------------------------------------------------------------------------------

ACTION DATE       TIME     URID         LSN          STATUS   AUTHID
------ ---------- -------- ------------ ------------ -------- ----------
DELETE 2005/09/29 16:07:35 0000000044BA 00000ABE00FC COMPLETE DB2INSK1

NGNAME             TSNAME             TABLE OWNER  TABLE NAME
------------------ ------------------ ------------ --------------------
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR

ROW STATUS  T_ID T_TYPE T_MODEL T_YEAR T_COLOR T_PACKAGE T_AC T_POWERWINDOWS
T_AUTOMATIC T_DOORS T_PRICE  T_DATA
----------- ---- ------ ------- ------ ------- --------- ---- --------------
----------- ------- -------- ----------------
POST-CHANGE -    -     -       -     -       -         -    -            -
-       -       -
PRE-CHANGE  5032 Wagon  Boran  1973  Purple  Off-Road  N    N            N
5       87786.00 Needs a tune up.

------------------------------------------------------------------------------

ACTION DATE       TIME     URID         LSN          STATUS   AUTHID
------ ---------- -------- ------------ ------------ -------- ----------
DELETE 2005/09/29 16:07:35 0000000044BA 00000ABE01BE COMPLETE DB2INSK1

NGNAME             TSNAME             TABLE OWNER  TABLE NAME
------------------ ------------------ ------------ --------------------
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR

ROW STATUS  T_ID T_TYPE T_MODEL T_YEAR T_COLOR T_PACKAGE   T_AC T_POWERWINDOWS
T_AUTOMATIC T_DOORS T_PRICE  T_DATA
```

```
----------- ---- ------ ------- ------ ------- ----------- ---- --------------
----------- ------- -------- ----------------------

POST-CHANGE -    -      -       -      -       -           -    -
-           -       -        -
PRE-CHANGE  5494 Van    Boran   1973   Green   Performance N    N
Y           4       97788.00 Minor engine knocking.


------------------------------------------------------------------------------

ACTION DATE       TIME     URID         LSN          STATUS   AUTHID
------ ---------- -------- ------------ ------------ -------- ----------
DELETE 2005/09/29 16:07:35 0000000044BA 00000ABE0286 COMPLETE DB2INSK1

NGNAME             TSNAME             TABLE OWNER  TABLE NAME
------------------ ------------------ ------------ --------------------
IBMDEFAULTGROUP    SMSDAT1            VPS          CAR
```

B.3 Excerpt of the undo SQL

Example B-3 is an excerpt from the undo SQL that was created.

Example: B-3 Excerpt of the undo SQL

```
-- UNDO DELETE FROM "VPS"."CAR" IN URID 0000000044BA AT LSN 00000ABEE349
INSERT INTO "VPS"."CAR"
           ("T_ID",
            "T_TYPE",
            "T_MODEL",
            "T_YEAR",
            "T_COLOR",
            "T_PACKAGE",
            "T_AC",
            "T_POWERWINDOWS",
            "T_AUTOMATIC",
            "T_DOORS",
            "T_PRICE",
            "T_DATA")
    VALUES (121014,
            'Pickup',
            'Boran',
            1973,
            'Blue',
            'Off-Road',
            'N',
            'N',
```

```
             'N',
             6,
             57501.00,
             'Needs an oil change.');
-- UNDO DELETE FROM "VPS"."CAR" IN URID 0000000044BA AT LSN 00000ABEE283
INSERT INTO "VPS"."CAR"
            ("T_ID",
             "T_TYPE",
             "T_MODEL",
             "T_YEAR",
             "T_COLOR",
             "T_PACKAGE",
             "T_AC",
             "T_POWERWINDOWS",
             "T_AUTOMATIC",
             "T_DOORS",
             "T_PRICE",
             "T_DATA")
     VALUES (121015,
             'Minivan',
             'Boran',
             1973,
             'Blue',
             'Towing',
             'N',
             'N',
             'N',
             4,
             29072.00,
             'Minor engine knocking.');
-- UNDO DELETE FROM "VPS"."CAR" IN URID 0000000044BA AT LSN 00000ABEE1C3
INSERT INTO "VPS"."CAR"
            ("T_ID",
             "T_TYPE",
             "T_MODEL",
             "T_YEAR",
             "T_COLOR",
             "T_PACKAGE",
             "T_AC",
             "T_POWERWINDOWS",
             "T_AUTOMATIC",
             "T_DOORS",
             "T_PRICE",
             "T_DATA")
     VALUES (121016,
             'Pickup',
             'Boran',
             1973,
             'Blue',
```

```
            'Towing',
            'N',
            'Y',
            'Y',
            6,
            99587.00,
            'Some paint chips.');
-- UNDO DELETE FROM "VPS"."CAR" IN URID 0000000044BA AT LSN 00000ABEE10A
INSERT INTO "VPS"."CAR"
           ("T_ID",
            "T_TYPE",
            "T_MODEL",
            "T_YEAR",
            "T_COLOR",
            "T_PACKAGE",
            "T_AC",
            "T_POWERWINDOWS",
            "T_AUTOMATIC",
            "T_DOORS",
            "T_PRICE",
            "T_DATA")
    VALUES (121017,
            'Van',
            'Boran',
            1973,
            'Blue',
            'Off-Road',
            'N',
            'N',
            'N',
            6,
            7828.00,
            'Sold as-is.');
-- UNDO DELETE FROM "VPS"."CAR" IN URID 0000000044BA AT LSN 00000ABEE04B
INSERT INTO "VPS"."CAR"
           ("T_ID",
            "T_TYPE",
            "T_MODEL",
            "T_YEAR",
            "T_COLOR",
            "T_PACKAGE",
            "T_AC",
            "T_POWERWINDOWS",
            "T_AUTOMATIC",
            "T_DOORS",
            "T_PRICE",
            "T_DATA")
    VALUES (121018,
            'Van',
```

```
            'Boran',
            1973,
            'Blue',
            'Off-Road',
            'N',
            'N',
            'N',
            6,
            75142.00,
            'Some paint chips.');
```

B.4 DDL of VPS.CAR table

Example B-4 is an example of the DDL of the VPS.CAR table.

Example: B-4 DDL of the VPS.CAR table

```
create table vps.CAR (
        T_ID integer not null,
        T_TYPE varchar(50) not null,
        T_MODEL varchar(50) not null,
        T_YEAR integer,
        T_COLOR varchar(50),
        T_PACKAGE varchar(50),
        T_AC char(1),
        T_POWERWINDOWS char(1),
        T_AUTOMATIC char(1),
        T_DOORS smallint,
        T_PRICE decimal(10,2) not null,
        T_DATA varchar(500),
primary key (T_ID))
```

C

Additional material

This redbook refers to additional material that you can download from the Internet as described in this appendix.

C.1 Locating the Web material

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

ftp://www.redbooks.ibm.com/redbooks/SG24-7133

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

ibm.com/redbooks

Select **Additional materials** and open the directory that corresponds with the redbook form number, SG24-7133.

C.2 Using the Web material

The additional Web material that accompanies this redbook includes the following file:

File name	*Description*
EasyOX.zip	Contains three bat files and one Perl script sample for running EasyOX.

C.2.1 System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space: 1 MB minimum
Operating System: Windows 2000/2003/XP, AIX, Linux
Processor: 386 or higher
Memory: 256 MB RAM

C.2.2 How to use the Web material

The sample `easyox` program assumes that you have Recovery Expert installed with the documented prerequisites.

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zipped file into this folder. Copy the three .bat files and the Perl script file to the BIN directory of the Recovery Expert installation path. Edit the .bat files according to the instructions in 8.4.8, “Using sample program EasyOX to automate cloning tables” on page 241.

Abbreviations and acronyms

AIX	Advanced Interactive eXecutive from IBM
ALID	Archive Log Inventory Data
APAR	authorized program analysis report
ARM	automatic restart manager
ASCII	American National Standard Code for Information Interchange
BLOB	binary large objects
CCA	client configuration assistant
CCSID	coded character set identifier
CD	compact disk
CEC	central electronics complex
CF	coupling facility
CFCC	coupling facility control code
CFRM	coupling facility resource management
CLI	call level interface
CLI	command line interface
CLP	command line processor
CPU	central processing unit
CSA	common storage area
CTT	created temporary table
DASD	direct access storage device
DB2	default DB2 instance
DB2 PM	DB2 performance monitor
DBA	database administrator
DBAT	database access thread
DBD	database descriptor
DBID	database identifier
DBRM	database request module
DCC	Data Capture Changes
DCL	data control language
DDCS	distributed database connection services
DDF	distributed data facility
DDL	data definition language
DLL	dynamic load library manipulation language
DML	data manipulation language
DNS	domain name server
DPF	database partitioning feature
DRDA®	distributed relational database architecture
DSC	dynamic statement cache, local or global
DTT	declared temporary tables
EA	extended addressability
EBCDIC	extended binary coded decimal interchange code
ECS	enhanced catalog sharing
ECSA	extended common storage area
EDM	environment descriptor management
ERP	enterprise resource planning
ESA	Enterprise Systems Architecture
ESP	Enterprise Solution Package
ETR	external throughput rate, an elapsed time measure, focuses on system capacity
FAQ	frequently asked questions
FTD	functional track directory
FTP	File Transfer Program

GB	gigabyte (1,073,741,824 bytes)
GBP	group buffer pool
GRS	global resource serialization
GUI	graphical user interface
HPJ	high performance Java
I/O	input/output
IBM	International Business Machines Corporation
ICF	integrated catalog facility
ICMF	internal coupling migration facility
IFCID	instrumentation facility component identifier
IFI	instrumentation facility interface
IPLA	IBM Program Licence Agreement
IRLM	internal resource lock manager
ISPF	interactive system productivity facility
ISV	independent software vendor
IT	Information Technology
ITR	internal throughput rate, a processor time measure, focuses on processor capacity
ITSO	International Technical Support Organization
IVP	installation verification process
JDBC™	Java Database Connectivity
JFS	journaled file systems
JNDI	Java Naming and Directory Interface™
JVM	Java Virtual Machine
KB	kilobyte (1,024 bytes)
LA	Log Analysis
LOB	large object
LPL	logical page list
LRECL	logical record length
LRSN	log record sequence number
LUN	logical unit number
LUW	logical unit of work
LVM	logical volume manager
MB	megabyte (1,048,576 bytes)
NPI	non-partitioning index
ODB	object descriptor in DBD
ODBC	Open Data Base Connectivity
PAV	parallel access volume
PDS	partitioned data set
PIB	parallel index build
PIT	point-in-time
PSID	page set identifier
PSP	preventive service planning
PTF	program temporary fix
PUNC	possibly uncommitted
QA	Quality Assurance
QMF™	Query Management Facility
RACF®	Resource Access Control Facility
RBA	relative byte address
RE	Recovery Expert
RECFM	record format
RI	referential integrity
RID	record identifier
RR	repeatable read
RRS	resource recovery services
RRSAF	resource recovery services attach facility
RS	read stability
SDK	software developers kit
Short	--fm-- <$filename>

SMIT	System Management Interface Tool
SMS	System Managed Space
SU	switch user
UOW	unit of work
VR	Versioning Repository

Related publications

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

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 383. Note that some of the documents referenced here might be available in softcopy only:

- *DB2 Recovery Expert for Multiplatforms*, SG24-6421

Other publications

The publications that we list here are also relevant as further information sources.

IBM DB2 documentation

- *IBM DB2 Recovery Expert for Multiplatforms: User’s Guide*, SC18-9564

  http://publib.boulder.ibm.com/epubs/pdf/aryu1b12.pdf
- *IBM DB2 Grouper User’s Guide*, SC18-7409, available at:

  http://publib.boulder.ibm.com/epubs/pdf/gxquga16.pdf

IBM DB2 UDB documentation

- *What’s New V8,* SC09-4848
- *Administration Guide: Implementation V8*, SC09-4820
- *Administration Guide: Performance V8*, SC09-4821
- *Administration Guide: Planning V8*, SC09-4822
- *Application Development Guide: Building and Running Applications V8*, SC09-4825
- *Application Development Guide: Programming Client Applications V8*, SC09-4826
- *Application Development Guide: Programming Server Applications V8*, SC09-4827

- *Call Level Interface Guide and Reference, Volume 1*, *V8*, SC09-4849
- *Call Level Interface Guide and Reference, Volume 2, V8*, SC09-4850
- *Command Reference V8*, SC09-4828
- *Data Movement Utilities Guide and Reference V8*, SC09-4830-01
- *Data Recovery and High Availability Guide and Reference V8*, SC09-4831
- *Guide to GUI Tools for Administration and Development*, SC09-4851
- *Installation and Configuration Supplement V8*, GC09-4837
- *Quick Beginnings for DB2 Clients V8*, GC09-4832
- *Quick Beginnings for DB2 Servers V8*, GC09-4836
- *Replication and Event Publishing Guide and Reference*, SC18-7568
- *SQL Reference, Volume 1, V8*, SC09-4844
- *SQL Reference, Volume 2, V8*, SC09-4845
- *System Monitor Guide and Reference V8,* SC09-4847
- *Data Warehouse Center Application Integration Guide Version 8 Release 1*, SC27-1124-01
- *DB2 XML Extender Administration and Programming Guide Version 8 Release 1*, SC27-1234
- *Federated Systems PIC Guide Version 8 Release 1*, GC27-1224

IBM - AIX

- AIX operating system Library:

  http://www-1.ibm.com/servers/aix/library/index.html

- High Availability

  http://www-1.ibm.com/servers/eserver/pseries/ha/

- IBM AIX 5L operating system

  http://www-1.ibm.com/servers/aix/

Online resources

These Web sites and URLs are also relevant as further information sources:

- IBM Data Management Tools Home page

  http://www.ibm.com/software/data/db2imstools/

- Data Management Tools support page:

  http://www.ibm.com/software/data/db2imstools/support.html

- DB2 UDB Web page:

  http://www.ibm.com/software/data/db2/udb/

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications, and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

E

F

G

H

I

J

K

L

M

N

O

P

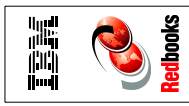
T

U

V

W

X



IBM DB2 Recovery Expert for Multiplatforms V2 Usage Guide

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

IBM DB2 Recovery Expert for Multiplatforms V2 Usage Guide

Install and implement DB2 Recovery Expert, Fast Backup, and DB2 Grouper

Learn how to manage recovery assets

Includes recovery and log analysis scenarios

IBM DB2 Recovery Expert for Multiplatforms provides an enhanced data recovery solution that enables more precise recovery operations while reducing disruption during the recovery process. DB2 Recovery Expert creates and maintains additional recovery assets that you can use for intelligent analysis of both DB2 and DB2 Recovery Expert assets to find the most efficient recovery path. DB2 Recovery Expert facilitates the process of rebuilding your database assets, such as tables, indexes, and data, to a specified point-in-time, often without taking the database or the business operations offline.

This IBM Redbook shows you how to install and implement DB Recovery Expert and the Fast Backup and Grouper components. It also describes the DB2 Recovery Expert assets and how to manage these assets.

In addition, this redbook explains the summary and detail reports of Log Analysis, including the DPF environment. It shows, in detailed examples, point-in-time recovery and cloning of tables, table spaces, and databases.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

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

For more information:
ibm.com/redbooks

SG24-7133-00 ISBN 0738496286