Information Management software

IBM

IBM DB2 Recovery Expert for z/OS User Scenarios

DB2 for z/OS recovery overview

Recovery Expert for z/OS architecture

User scenarios

Nagraj Alur
Hennie Mynhardt
Gregg Palmer
Alicia Wangelien

Redbooks

ibm.com/redbooks

International Technical Support Organization

IBM DB2 Recovery Expert for z/OS User Scenarios

November 2006

SG24-7226-00

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

First Edition (November 2006)

This edition applies to Version 1, Release 1, Modification 0 of IBM DB2 Recovery Expert for z/OS (product number 5697-H74).

Contents

Figures

Tables

Examples

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 illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

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

CICS®
DB2 Universal Database™
DB2®
DFSMSdss™
DFSMShsm™
IBM®
MVS™
RACF®
Redbooks (logo) ™
Redbooks™
z/OS®

The following terms are trademarks of other companies:

SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

Microsoft, 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.

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

Preface

This IBM® Redbook describes the procedures for using IBM DB2® Recovery Expert for z/OS® to manage the recovery of DB2 for z/OS objects in a data sharing and non-data sharing environment. IBM DB2 Recovery Expert for z/OS is an easy-to-use, automated recovery solution that enables database recovery operations with minimal disruption. Its graphical user interface (GUI) provides powerful reporting and automated recovery capabilities for productive database maintenance and high availability. This book is written for an audience of DB2 for z/OS database administrators (DBA).

This book gives you an overview of DB2 Recovery Expert for z/OS, its main features, architecture, recovery semantics, and processing flow. It also describes some of the key considerations in choosing a particular topology to address a business requirement. We provide user scenarios with step-by-step approaches to recover single DB2 for z/OS objects, multiple DB2 for z/OS objects, dropped DB2 for z/OS tables, and DB2 for z/OS subsystems using the IBM Recovery Expert for z/OS tool.

We discuss backup and recovery in a DB2 for z/OS environment, the types of failures that might occur, the types of recovery that are supported, the types of objects that can be recovered, the elements and tools of recovery, and the recovery flow. We also provide a brief tutorial of the DB2 Recovery Expert for z/OS GUI client and an overview of the schema level repository (SLR) with recommendations for its maintenance.

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.

Nagraj Alur is a Project Leader with the IBM ITSO, San Jose Center. He holds a Master's degree in Computer Science from the Indian Institute of Technology (IIT), Mumbai, India. He has more than 30 years of experience in database management systems (DBMSs), and has been a programmer, systems analyst, project leader, independent consultant, and researcher. His areas of expertise include DBMSs, data warehousing, distributed systems management, database performance, information integration, and client/server and Internet computing. He has written extensively on these subjects and has taught classes and presented at conferences all around the world. Before joining the ITSO in November 2001, he was on a two-year assignment from the Software Group to the IBM Almaden Research Center, where he worked on Data Links solutions and an eSourcing prototype.

Hennie Mynhardt is an IT Specialist with IBM Software Group. Hennie has 24 years of database experience, 15 years of them with DB2. He has worked on technical projects and with companies in various countries overseas doing consulting, database administration, performance tuning, disaster recovery (DR) implementations, and problem determination. His special interests are DB2 system performance and backup and recovery. He is currently a member of the IBM Software Group (SWG) DB2 Database Engine and Tools technical sales team responsible for presales and postsales technical support. His areas of expertise include DB2 system, database administration, tools, and utilities.

Gregg Palmer is the Level 2 support for Recovery Expert for z/OS. He joined IBM in 1979 as a program support representative (PSR) in downtown Chicago, and moved to California in December 1980 as IBM Customer Information Control System (CICS®) Level 2 until 1984. From that time onward until 2006, he was Level 2 support for DB2 for z/OS. DB2 Level. After a two-year stint (1989 - 1990) as part of the change team in the subcomponent DATA MANAGER, he returned to DB2 for z/OS Level 2 support.

Alicia Wangelien is a DB2 DBA for z/OS for IBM Integrated Technology Delivery in Sterling Forest, NY. She has been with IBM for 24 years, 17 years of them providing DB2 DBA z/OS production support for commercial accounts, IBM internal accounts, and SAP® solutions in various industry sectors including advertising, benefits, finance, human resources, insurance, and manufacturing. She is currently a DB2 DBA z/OS technical team lead, Server Systems Operations (SSO) DBA Service Line Metric team lead for Distributed and Mainframe DBMS technologies, and the SSO DBA Service Line Metric representative for the Americas (U.S. and Canada) on the Global Data Management Metric team. She is an IBM Certified Database Administrator in IBM DB2 Universal Database™ V8.1 for z/OS. She holds a Master of Business Administration degree with Honors majoring in MIS, from IONA College.

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

Cynthia Browne
Carol DeGrace
Joel Farber
Donald Gosiewski
June Kim
David Schwartz
Frank Sherwin
Bryan F Smith
Pierre Washington
IBM Silicon Valley Laboratory, San Jose

Richard Conway
ITSO, Poughkeepsie Center

Paul Bartholf
Jeff Celander
Thomas Hubbard
John Pittman
Mark Wallace
Shawn Wikle
Rocket Software

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 have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

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 email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

IBM DB2 Recovery Expert for z/OS architecture

In this chapter, we describe the main features of IBM DB2 Recovery Expert for z/OS, its architecture, the recoverable objects, types of recovery, recovery semantics, and processing flow. We also describe some of the key considerations in choosing a particular topology to address a business requirement.

The topics covered are:

- Main features
- Architecture
- Recovery objects, recovery types, and recovery semantics
- Processing flow

1.1 Main features

The IBM DB2 Recovery Expert for z/OS tool provides an easy-to-use, automated recovery solution that enables database recovery operations with minimal disruption, and enables you you to maintain high availability for database users. Its easy-to-use graphical user interface (GUI) provides powerful reporting and automated recovery capabilities for productive database maintenance and high availability.

Menus make recovery to a specific point-in-time (PIT) simple and easy to effect. The tool creates recovery options that include rolling changes forward or backward, whichever is the most efficient in a given situation. Not only does the tool provide options to recovery scenarios, it also makes recommendations as to which option is relatively the least expensive in any given situation. This expert functionality saves you time and money by helping you make better decisions.

IBM DB2 Recovery Expert for z/OS also makes versioning easy, because by using the tool you can track object versions and data dependencies. It also supports recovery through the RESTORE SYSTEM utility of DB2 for z/OS Version 8. It simplifies complex and laborious operations such as reversing undesired data changes including those that have cascaded to related tables, and rebuilding database assets that have been accidentally dropped and therefore do not exist in the DB2 system catalog.

You can also use object profiles developed with the IBM DB2 Automation Tool to recover a set of objects using IBM DB2 Recovery Expert for z/OS. In addition, the tool generates all the necessary job control language (JCL), submits it, and helps you to track progress as recovery proceeds. You can also use the log analysis function to determine *quiet times*, thereby ensuring that the objects you recover have no activity occurring against them.

IBM DB2 Recovery Expert for z/OS also includes the IBM DB2 Grouper component, which provides a method to discover related sets of tables. This ensures that all relevant objects are recovered together thereby maintaining integrity of your data. IBM DB2 Recovery Expert for z/OS restores all missing objects related to the objects selected for recovery. By default, Recovery Expert for z/OS always recovers missing objects related through DB2 object dependencies.

To summarize, the main features of IBM DB2 Recovery Expert for z/OS include:

- Windows®-based simple, easy-to-use GUI with menus that make recoveries to a point-in-time and current quick and precise.
- Multiple options for recovery and recommendations about the least expensive option for a given situation; but leaves the decision to you to choose the most appropriate option for your situation
- Schema level repository (SLR) that inspects version levels available for restoration; this includes related dependent objects even if they no longer exist in the DB2 system catalog
- Support for the RESTORE SYSTEM utility of DB2 for z/OS Version 8
- IBM DB2 Grouper is included as a component, which helps you to maintain data integrity by discovering related sets of tables and ensuring that all related objects are recovered together
- Object profiles developed with any version of IBM DB2 Automation Tool that you can use to recover a set of objects
- Log analysis function that helps you to determine quiet times to ensure that recovered objects to that time do not have any activity occurring against them

- Supports non-data sharing and data sharing environments
- Supports DB2 for z/OS Version 7 and DB2 for z/OS Version 8 (CM, ENFM, and NFM)
- Does not run under Interactive System Productivity Facility (ISPF) or invoke ISPF services

For further information about the functions and constraints, see *DB2 Recovery Expert for z/OS, V1R1, User's Guide*, SC18-9822.

1.2 Architecture

Figure 1-1 shows the client/server nature of the architecture of IBM DB2 Recovery Expert for z/OS. A single instance of IBM DB2 Recovery Expert for z/OS can manage one or more DB2 subsystems located on one or more machines or sysplexes that might be geographically distributed.

> **Attention:** We have referred to the default name for the product libraries throughout this book. If you change the name of the product libraries, you must substitute the appropriate names that are used in your installation.

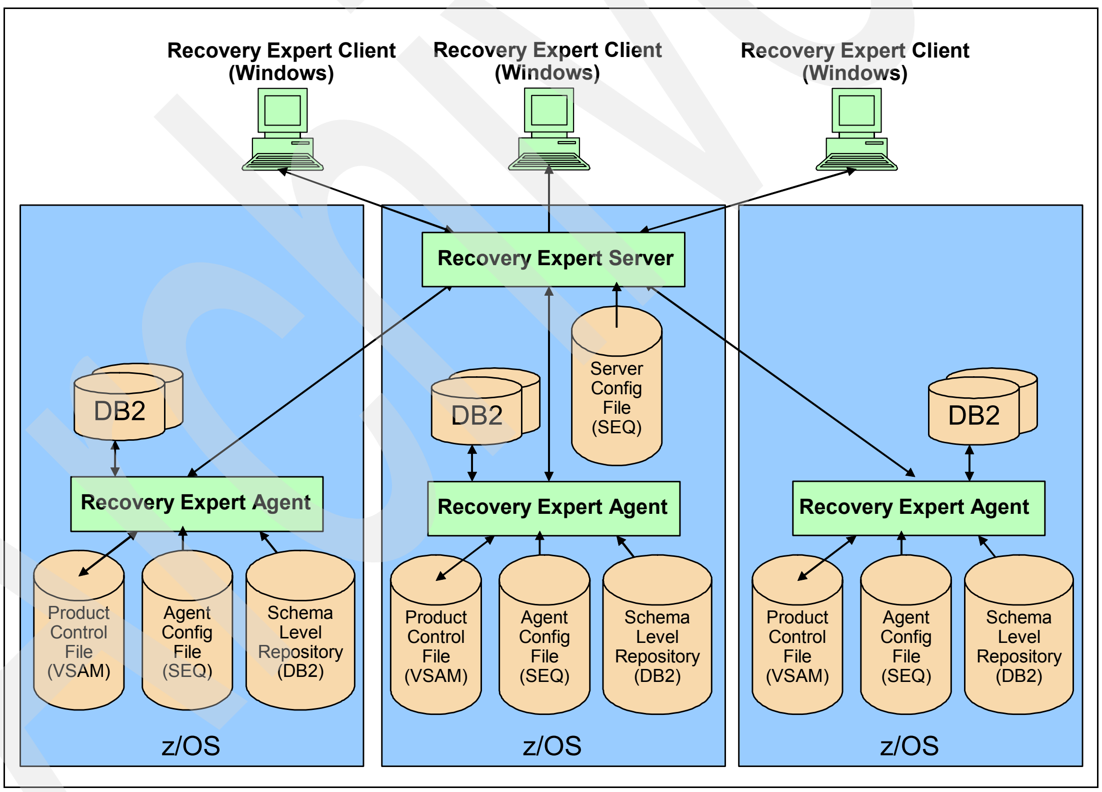

Figure 1-1 IBM DB2 Recovery Expert for z/OS architecture

The main components of IBM DB2 Recovery for z/OS are:

- Recovery Expert server

  The DB2 Recovery Expert for z/OS server centrally manages and controls all DB2 Recovery Expert for z/OS functions that are performed on behalf of user requests. You must run one instance of the server to manage all of your DB2 subsystems and data sharing groups and to support all of your DB2 Recovery Expert for z/OS user clients.

Using TCP/IP connections, the DB2 Recovery Expert for z/OS server, clients, and agents communicate with each other to perform the recovery functions.

The DB2 Recovery Expert for z/OS server is associated with a server configuration file, which identifies the DB2 Recovery Expert for z/OS client listener port (default 9875) and DB2 Recovery Expert for z/OS agent listener port (default 9876) along with the logging level, and whether to include all log messages, all log messages with warning severity or higher, all log messages with error severity or higher, and so on. Example 1-1 shows a sample DB2 Recovery Expert for z/OS server configuration file, which is a sequential z/OS data set or partitioned data set (PDS) member.

Example 1-1 Partial contents of a sample DB2 Recovery Expert for z/OS server configuration file

```
<server-config>
<client-listener-port>9875</client-listener-port>
<agent-listener-port>9876</agent-listener-port>
<description>wtsc59.itso.ibm.com</description>
<community-string>wtsc59server</community-string>
<multicast-address>236.1.2.3</multicast-address>
<multicast-port>19875</multicast-port>
.....................................
<log-level>I</log-level> </server-config>
<bind-retry-max>30</bind-retry-max>
<bind-retry-delay>10</bind-retry-delay>
<trace-network>false</trace-network>
<trace-xml>false</trace-xml>
<trace-events>false</trace-events>
<trace-config>false</trace-config>
</server-config>
```

The DB2 Recovery Expert for z/OS server runs as a batch job under IBM Multiple Virtual Storage (MVS). You can manually submit this job, schedule it to be run automatically, or run it as an MVS started task. If the server job terminates for any reason, you must restart it, or take steps to have it automatically restarted.

Important: Start the Recovery Expert for z/OS server before you start the agent.

- Recovery Expert for z/OS agent

  The DB2 Recovery Expert for z/OS agent provides access to database and system services, in support of the DB2 Recovery Expert for z/OS server and remote clients. You must run one instance of the agent on every system or logical partition (LPAR) that hosts DB2 subsystems or data sharing groups that you want to access with DB2 Recovery Expert for z/OS. Each agent communicates with the DB2 Recovery Expert for z/OS server to provide services.

  The DB2 Recovery Expert for z/OS agent is associated with two files:

  – An agent configuration file, which identifies the DB2 Recovery Expert for z/OS server and corresponding port number that this agent is associated with, and the data definition (DD) name allocated to the product control file associated with this agent. Other parameters include the logging level, maximum number of DB2 Recovery Expert for z/OS server connection retries, delay between retries, and so on. Example 1-2 shows a sample DB2 Recovery Expert for z/OS agent configuration file, which is a sequential MVS data set.

Note: You can also configure the Recovery Expert for z/OS server, clients, and agents to connect to each other automatically, as described in 1.4.1, “Configuration flow” on page 23.

Example 1-2 Sample DB2 Recovery Expert for z/OS agent configuration file

```
<agent-config>
<server-address>wtsc59.itso.ibm.com</server-address>
<server-port>9876</server-port>
<community-string>wtsc59server</community-string>
<multicast-address>236.1.2.3</multicast-address>
<multicast-port>19875</multicast-port>
<control-file-dd>DB2PARMS</control-file-dd>
<log-level>I</log-level>
<server-connect-retry-max>30</server-connect-retry-max>
<server-connect-retry-delay>10</server-connect-retry-delay>
<request-thread-timeout>300</request-thread-timeout>
<uppercase-passwords>true</uppercase-passwords>
<job-poll-rate>5</job-poll-rate>
<job-cancel-timeout>5</job-cancel-timeout>
<check-ownership-external>true</check-ownership-external>
<trace-config>false</trace-config>
<trace-csi>false</trace-csi>
<trace-db2-attachment>false</trace-db2-attachment>
<trace-sql>false</trace-sql>
<trace-ifi>false</trace-ifi>
<trace-network>false</trace-network>
<trace-xml>false</trace-xml>
<trace-events>false</trace-events>
<trace-config>false</trace-config>
</agent-config>
```

– A product control file, which identifies target DB2 subsystems, and specifies product execution options, and data set attributes. The product control file is a Virtual Storage Access Method (VSAM) key-sequenced data set (KSDS), which is updated using the sample JCL in the member ARYSJ001 of the sample library SARYSAMP. The product control file itself is created by the member ARYSJ000 in the sample library SARYSAMP. Example 1-3 shows the partial contents of a sample DB2 Recovery Expert for z/OS product control file, which identifies a single DB2 subsystem DB8A as the target.

Important: Take great care to ensure that the contents of this file are defined accurately. Otherwise, you might experience unpredictable results, or 0C1, 0C4, or other abends. For a detailed description of the product control file parameters, see Appendix D, “Potential Recovery Expert for z/OS recovery plans” on page 321.

Example 1-3 Partial contents of a sample DB2 Recovery Expert for z/OS product control file in ARYSJ001

```
...........................
...........................
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*---------------------------------------------------------------------
```

```
*
*
 SET DB2 SSID           = DB8A
 UPDATE DB2 ZPARMS      = DSNZPARM
 UPDATE DB2 BOOTSTRAP1  = DB8AU.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8AU.BSDS02
 UPDATE DB2 LOADLIB1    = DB8A8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8A8.SDSNLOAD
......................
......................
*
*-------------------------------------------------------------------
* Sample statements to add/update ARY product plans
*-------------------------------------------------------------------
*
 SET DB2 SSID           = DB8A
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
*
.........................................
.........................................
*-------------------------------------------------------------------
* Sample statements to add/update data set prefix generation
*-------------------------------------------------------------------
*
*  The DSN PREFIX maximum length is 17 characters. If NULL
*  is specified then user id is used as data set prefix. Use &USERID
*  in the prefix to insert user id. Example: TEST.&USERID will
*  generate a data set prefix of 'TEST.MYID' where the user id is
*  'MYID'.
*
*UPDATE ARY DSN PREFIX  = xxxxxxxx.xxxxxxxx
*
 UPDATE ARY DSN PREFIX  = NALUR
*
*-------------------------------------------------------------------
................................
................................
*-------------------------------------------------------------------
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*-------------------------------------------------------------------
*
*  The ROWDATA VSAM data set is dynamically created by the log
*  analysis services when creating SQL from the log.
*
*  The DSN PREFIX maximum length is 21 characters. The following
*  set of product controls are required and must be properly set
*  to ensure proper log data recoveries. The VOLSERS statement
```

```
*  value can be set to blanks, if required. A maximum of 3 volsers
*  can be specified.
*
*UPDATE LAS DSN PREFIX  = xxxxxxxxxxxxxxxxxxxx
 UPDATE LAS DSN PREFIX  = NALUR
 UPDATE LAS VOLSERS     = OP1TSA,OP1TSB,OP1TSC
 UPDATE LAS DATA AUNIT  = C
 UPDATE LAS DATA PQTY   = 00005
 UPDATE LAS DATA SQTY   = 00005
 UPDATE LAS INDEX AUNIT = C
 UPDATE LAS INDEX PQTY  = 00005
 UPDATE LAS INDEX SQTY  = 00005
*
......................................

.......................................
```

The DB2 Recovery Expert for z/OS agent must be running for DB2 Recovery Expert for z/OS users to perform any functions against DB2 subsystems on that LPAR. The DB2 Recovery Expert for z/OS agent runs as a batch job under MVS. You can manually submit this job, schedule it to be run automatically, or run it as an MVS started task. If the agent job terminates for any reason, you must restart it, or take steps to have it automatically restarted.

Important: Start the Recovery Expert for z/OS server before you start the agent.

Attention: The agent also requires UNIX® System Services access. The user ID under which the agent job runs must have an OMVS segment in its IBM Resource Access Control Facility (RACF®) profile.

- Recovery Expert for z/OS GUI client

  To launch the Recovery Expert for z/OS GUI client on Microsoft® Windows platform, select **Start** → **Programs** → **IBM DB2 Recovery Expert** → **IBM DB2 Recovery Expert**. This opens the prompt to connect to the Recovery Expert for z/OS server, as shown in Figure 1-2.

  You can type the host name (or IP address) and port number (if known), or click the ellipsis to obtain a list of available servers[1] (Figure B-2 on page 233) to select from. The prompt window reflects the last server that you connected to. The drop-down list provides the list of servers that you connected to previously, as shown in Figure 1-2.

Figure 1-2   Prompt to connect to the DB2 Recovery Expert for z/OS server

1 To use this feature, enable automatic discovery, as described in “Automatic discovery” on page 24.

The most recently used server is selected by default. You can select a listed server or manually enter the host name and port of a server that is not listed, for example:

```
wtsc53.itso.ibm.com:9875
```

After you connect to the DB2 Recovery Expert for z/OS server, the main IBM DB2 Recovery Expert for z/OS window is displayed, as shown in Figure 1-3. The Messages pane shows messages, which indicate the status of running or completed tasks that are initiated by the current user, if any (Figure 1-20 on page 36 shows an example with messages in it). There are three buttons to the right that are used to operate on the selected task or tasks. The buttons, from left to right, are used to perform the following tasks:

– Open (You can also double-click a task to open it.)
– Cancel
– Delete

You can select various options such as recovery and log analysis, as shown in Figure 1-3.

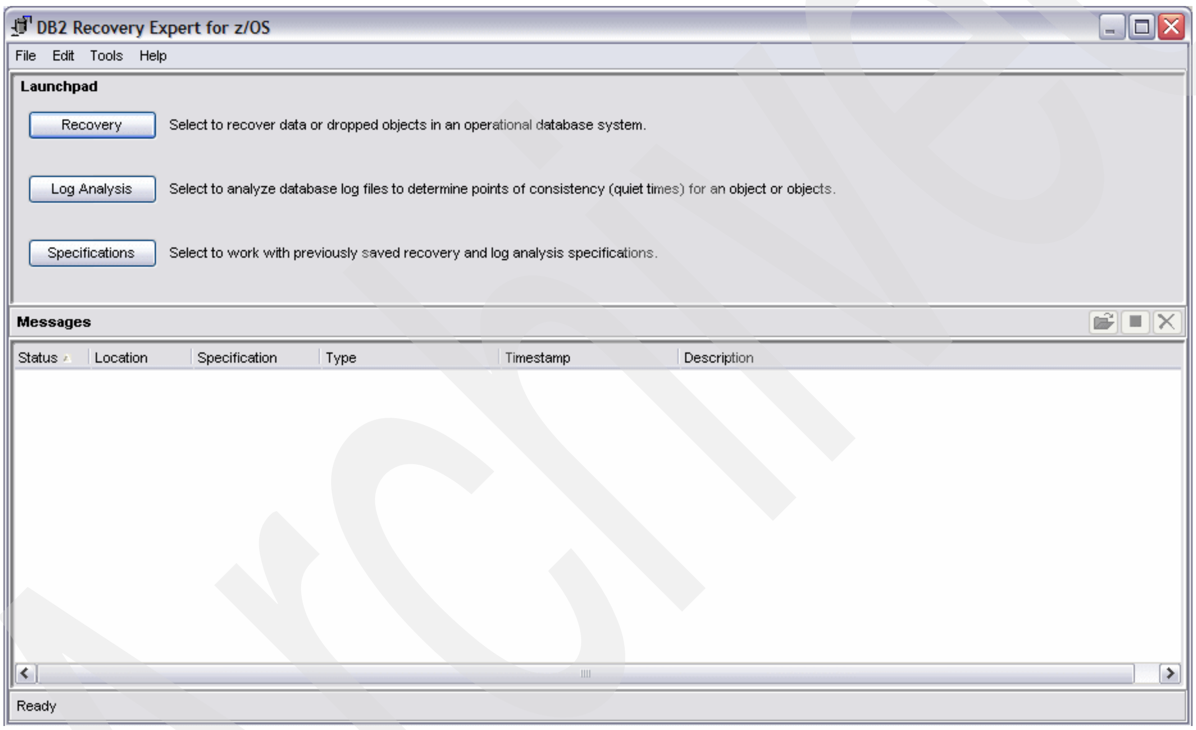

Figure 1-3 Main menu of the DB2 Recovery Expert for z/OS GUI client

- Schema level repository

  IBM DB2 Recovery Expert for z/OS captures DB2 system catalog information and stores it in a set of DB2 tables referred to as the schema level repository. The schema level repository is an archive to hold object definitions and alterations to object definitions. The sample library member ARYSJ002 contains JCL to update the schema level repository. The initial run of job ARYSJ002 might take multiple hours to copy the contents of the DB2 system catalog. Run times vary depending on the DB2 system catalog size. You must run the schema level repository update job at least daily. If important application object definition updates are performed, then run the schema level repository update after the object definition updates.

Note: When new objects are created in the DB2 subsystem that are not yet reflected in the schema level repository, the navigation pane in the DB2 Recovery Expert for z/OS GUI client highlights these new objects in blue. DB2 objects that are dropped and therefore only reflected in the schema level repository but not in the DB2 system catalog are highlighted in red in the navigation pane.

The Data Definition Language (DDL) to create the DB2 Recovery Expert for z/OS DB2 objects is shipped with the product to install in the IBM DB2 Tools SYSTOOLS database. An object creator schema name of SYSTOOLS is used.

- DB2 subsystems

  This is the list of DB2 subsystems that are managed by the DB2 Recovery Expert for z/OS agent as listed in the product control file associated with the agent.

1.3 Recovery objects, recovery types, and recovery semantics

This section describes the recovery objects, types of recovery, and recovery semantics supported by Recovery Expert for z/OS that extend the basic capabilities of the DB2 for z/OS engine. These Recovery Expert for z/OS features are discussed in detail in this section.

1.3.1 Recovery objects

The basic objects of recovery that are supported by Recovery Expert for z/OS are tables, table spaces, and table space partitions. The user can use the Recovery Expert for z/OS GUI client to specify aggregate objects such as databases, storage groups, plans and packages, DB2 Grouper profiles, and DB2 Automation Tool object profiles, which are then disassembled to the basic table space level unit of recovery. Recovery Expert for z/OS also supports the recovery of the entire DB2 subsystem using the RESTORE SYSTEM utility.

Currently, you *cannot* use the Recovery Expert for z/OS GUI client to specify the recovery of a data set of a non-partitioned table space, or the recovery of an index.

1.3.2 Recovery types

Recovery Expert for z/OS extends the capabilities of DB2 for z/OS engine by supporting the ability to recover dropped objects, and the ability to recover an object to any one of its prior object definition levels. It also provides the ability to identify potential recovery points or quiet times in the DB2 log using the log analysis facility.

Important: The requirement to keep the SLR current is critical to the ability of Recovery Expert for z/OS to extend the recovery capabilities of the DB2 for z/OS engine so that it accurately reflects the history of the creation of the object and any changes to its definition. This includes changes such as the addition of a column to a table, change in the data type of a column, or the establishment of recoverable points such as image copies and quiesce points.

Recovery Expert for z/OS supports the following types of recovery, as shown in Figure 1-4.

- Recovery of objects to current[2]
- Recovery of objects to a point-in-time
- Recovery of dropped objects
- Recovery of a subsystem

Attention: In all the following discussions about the types of recovery supported by Recovery Expert for z/OS, we assume that the SLR reflects an up-to-date history of events required to perform the requested recovery actions.

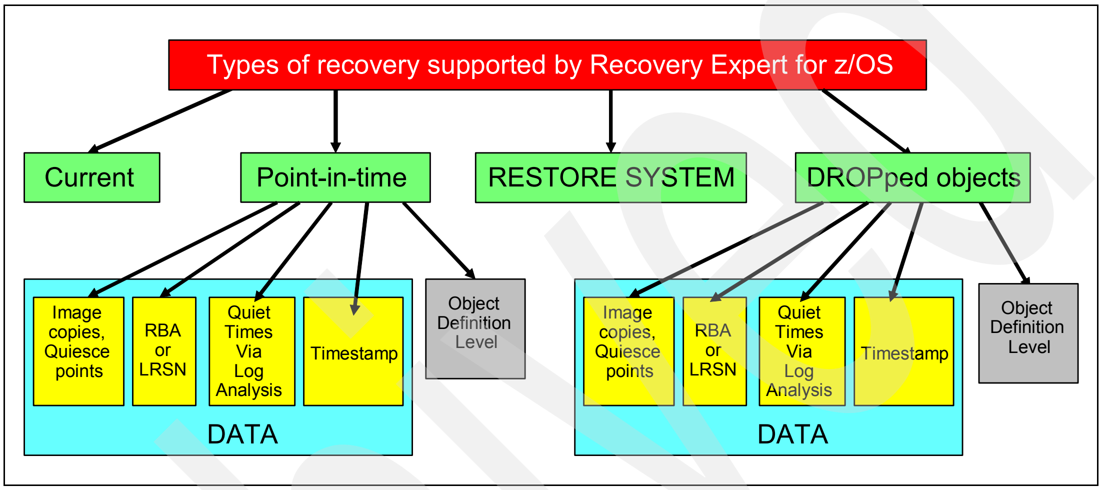

Figure 1-4 Types of recovery supported by Recovery Expert for z/OS

The following sections describe in detail each type of recovery shown in Figure 1-4.

Recovery of objects to current

The recovery of one or more objects to current usually occurs when there is a media failure, such as an input/output (I/O) error or a volume failure.

Note: Nowadays, due to the reliability of media, recovery of objects to the point of failure (POF) constitutes a small (single digit, perhaps) percentage of all recoveries that are performed within an organization; the majority of recoveries are performed due to operator errors, application errors, or site failures (disaster recovery).

Recovery of objects to a point-in-time

Recovery of objects to a point-in-time is the most common form of recovery performed today. It is triggered by operator errors and application errors. As shown in Figure 1-4, you can recover to the following points-in-time with Recovery Expert for z/OS:

- To a recoverable point identified in SYSIBM.SYSCOPY: This can be a full or incremental image copy (share level reference or change), or a quiesce point. In the Recovery Expert for z/OS GUI client, you can view SYSIBM.SYSCOPY recoverable points, as shown in Figure 1-5. To see image copies taken with share level change, select the **Show events that are not points of consistency** check box in Figure 1-5.

[2] This is sometimes called the point of failure (POF).

Note: You can obtain the list of recovery history events from the SLR and SYSIBM.SYSCOPY.

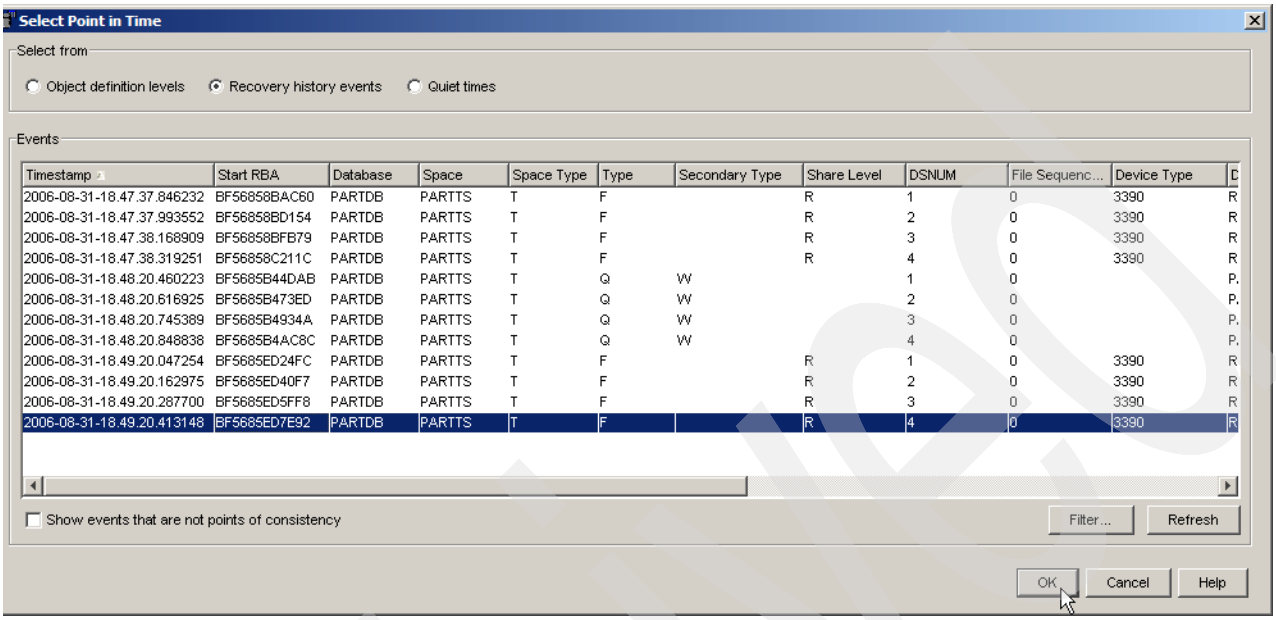

Figure 1-5 Recovery history events

- To a timestamp of your choosing, as shown in Figure 1-6: Here we assume that you have chosen the timestamp based on information gathered from some external source such as corresponding to when the DB2 subsystem was shut down for maintenance.

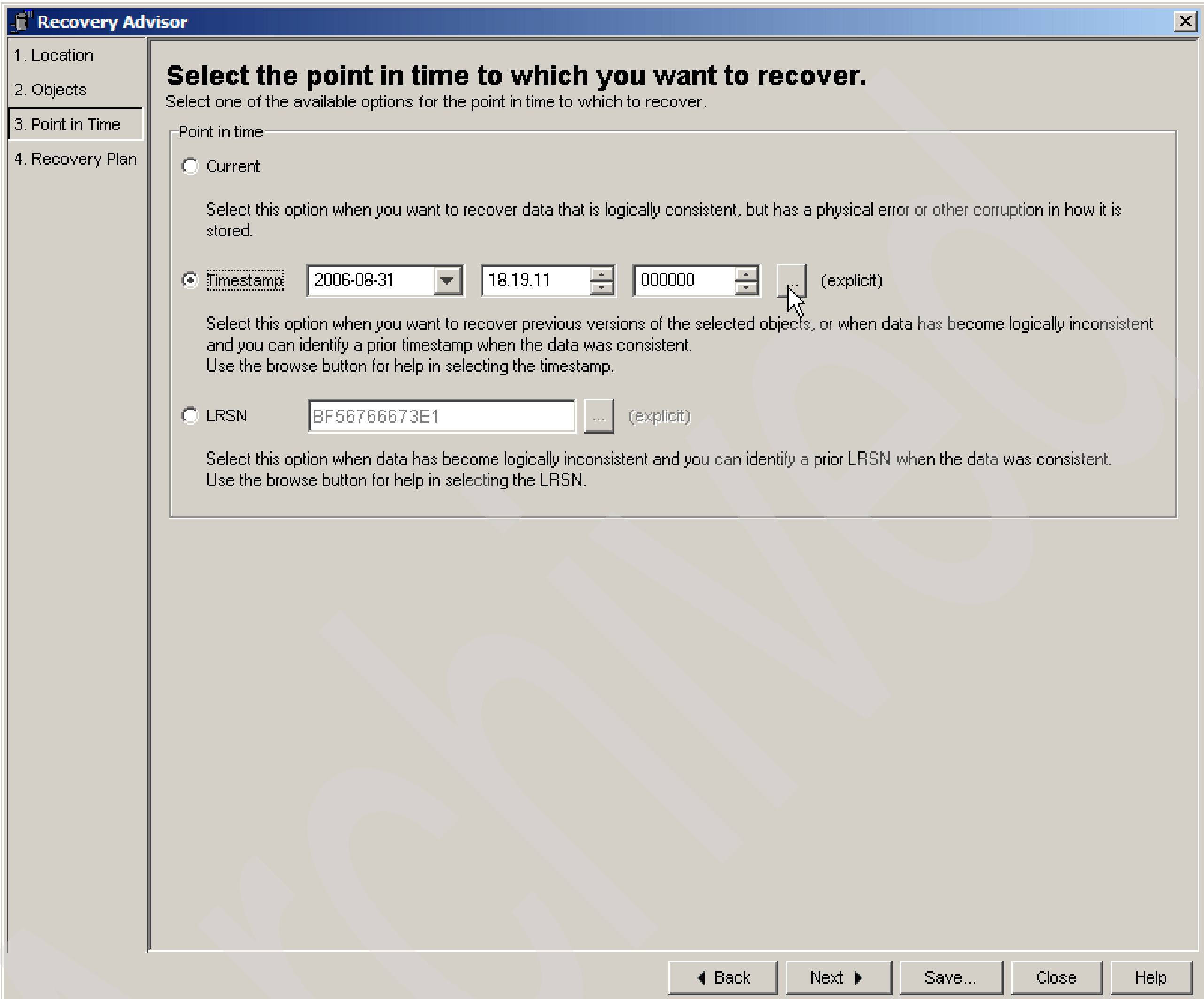

Figure 1-6 Recovery to timestamp

- To a relative byte address (RBA) (or log record sequence number (LRSN) in the case of data sharing) of your choosing, as shown in Figure 1-7: Here too, we assume that you have chosen the RBA or LRSN based on information gathered from some external source.

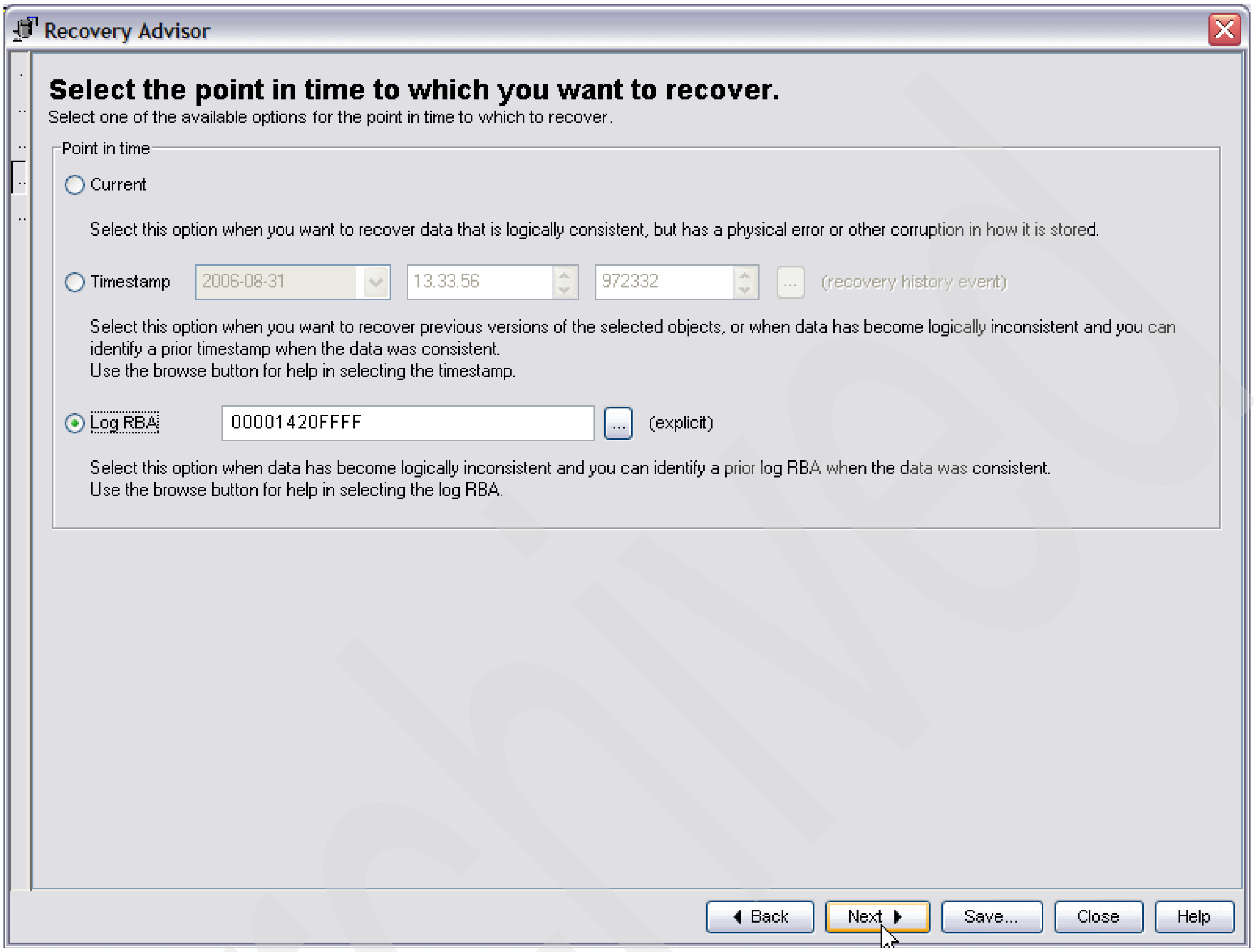

Figure 1-7 Recovery to RBA or LRSN

- To a quiet time as determined by running the log analysis component of Recovery Expert for z/OS: The quiet times are RBAs or LRSNs recorded in the SLR by Recovery Expert for z/OS, and are viewed as shown in Figure 1-8.

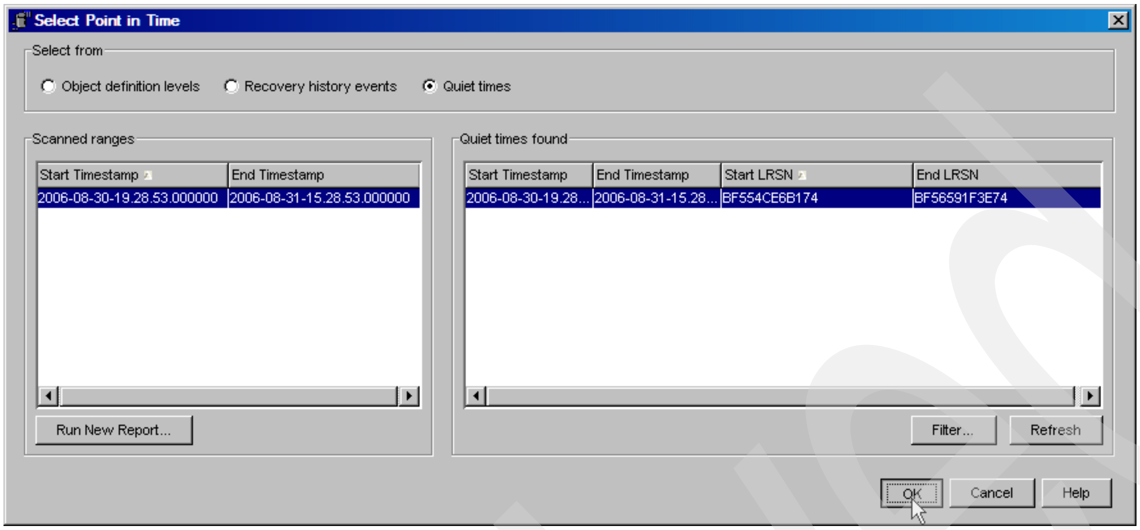

Figure 1-8 Recovery to quiet time

- To an object definition level that is recorded in the SLR by running the SLR update job (ARYSJ002 member in the ARY.V1R1M0.SARYSAMP library): This is a special construction of Recovery Expert for z/OS that is not available in the base DB2 for z/OS engine. The list of possible object definition levels that you can recover to is shown in Figure 1-9.

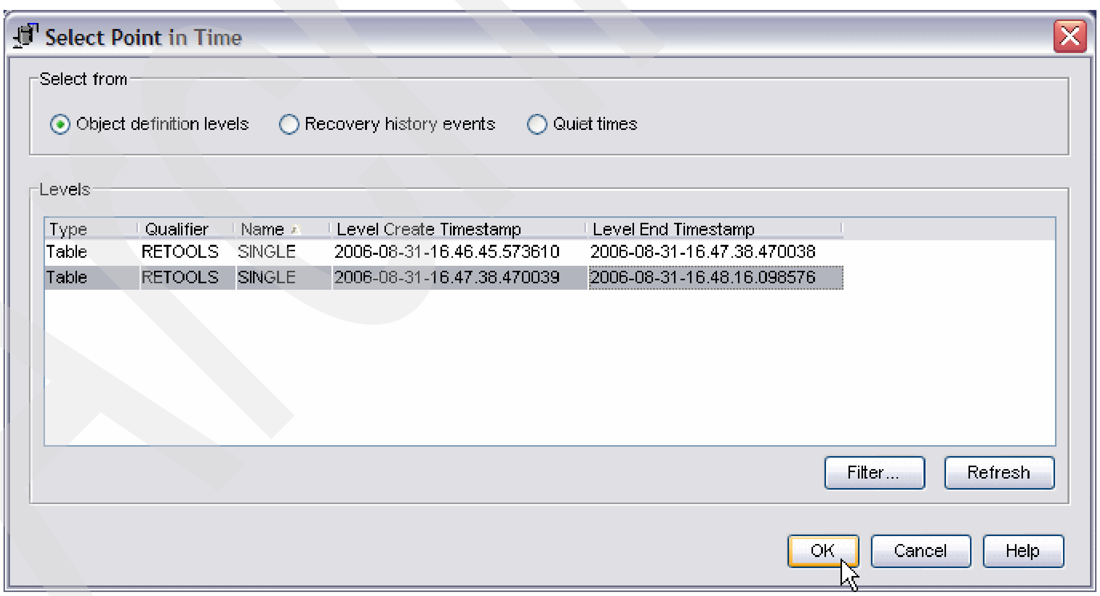

Figure 1-9 Recovery to an object definition level

Attention: Given the semantics of object definition levels, as discussed in 1.3.3, “Recovery semantics” on page 16, you must attempt to recover an object to an object definition level after you obtain a thorough understanding of the object definition levels recorded in your SLR tables.

Important: The semantics of recovering an object or set of objects to a point-in-time as implemented by Recovery Expert for z/OS differs from that implemented by the DB2 for z/OS engine. This is described in detail in 1.3.3, “Recovery semantics” on page 16.

In all of these cases, if Recovery Expert for z/OS cannot determine the recovery point to be a point-of-consistency[3], it issues a warning through a pop-up window (shown in Figure 1-10) that the recovery point chosen might not represent a point-of-consistency.

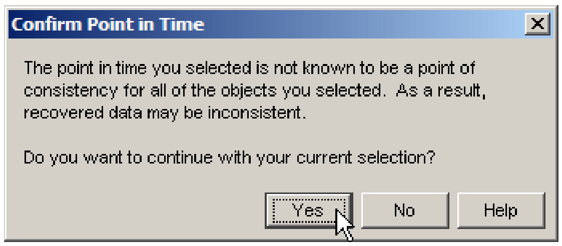

Figure 1-10 Point-of-consistency warning

Recovery of dropped objects

The recovery of a dropped object is invariably the consequence of a user error. When an object gets dropped accidentally, the various rows in the DB2 system catalog tables and DB2 directory are also deleted.

Recovery Expert for z/OS relies on the contents of the SLR to recover dropped objects. Therefore, it is critical that the SLR contains the most up-to-date history of recoverable information (such as object creation, alterations and drops, and information about image copies, quiesce points, and quiet times) to recover dropped objects. In particular, the SLR must contain a record of at least one full image copy for the dropped object to recover that object. Therefore, ensure that the SLR is updated frequently and also when critical application changes occur. For a more detailed discussion about how to update the SLR, see Appendix C, “Overview of schema level repository and maintenance considerations” on page 305.

Recovery of a subsystem

Recovery of a subsystem corresponds to the recovery of the entire DB2 subsystem to a prior point-in-time corresponding to when a backup was taken. DB2 for z/OS Version 8 introduced the BACKUP SYSTEM and RESTORE SYSTEM utilities to facilitate DB2 subsystem recovery using Data Facility Storage Management Subsystem (DFSMS) managed volumes of DB2 data and logs. Recovery Expert for z/OS uses the RESTORE SYSTEM utility of the DB2 for z/OS engine to provide DB2 subsystem recovery.

Note: When a BACKUP SYSTEM utility is issued, relevant information is written to the bootstrap data set (BSDS). For Recovery Expert for z/OS agent to become aware of it, the agent must be recycled.

[3] Point-of-consistency (POC) is a point when there are no outstanding units of work for the objects that are being recovered. A POC is guaranteed when the recovery point is a quiesce point, or a full image copy with share level of reference, or a quiet time as determined by running log analysis.

1.3.3 Recovery semantics

This section describes the semantics of recovering an object or set of objects to a point-in-time with the help of scenarios. Figure 1-11 illustrates a possible timeline of table creation, table alteration, image copies, quiesce points, an optional DROP table (to explain the particular scenario), and SLR updates. It shows near optimal SLR updates that ensure that almost all table creation, alteration, and image copies are recorded in the SLR.

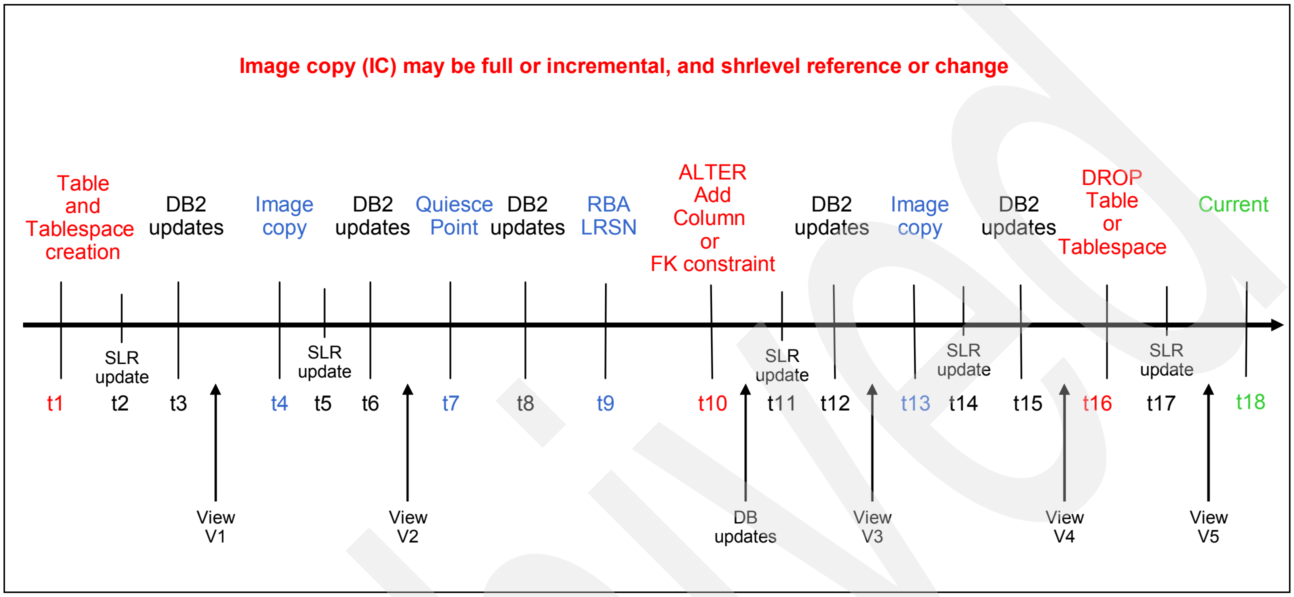

Figure 1-11 Timeline of object creation, alterations, image copies, quiesce points, and SLR updates

Table 1-1 lists the potential recovery scenarios and the semantics associated with each one.

Table 1-1 Recovery Expert for z/OS point-in-time recovery semantics

Action requested	Base DB2 semantics		Recovery Expert semantics			
			Existing table		Dropped table	
	DDL	Data	DDL	Data	DDL	Data
Recover table to IC at time t4	ALTER DDL at time t10	IC at time t4	Table creation at time t1	IC at time t4	Table creation at time t1	IC at time t4
Recover table to quiesce point at time t7	ALTER DDL at time t10	Quiesce point at time t7	Table creation at time t1	Quiesce point at time t7	Table creation at time t1	Quiesce point at time t7[a]
Recover table to IC at time t13	ALTER DDL at time t10	IC at time t13	ALTER DDL at time t10	IC at time t13	ALTER DDL at time t10	IC at time t13
Recover table to current at time t18	ALTER DDL at time t10	Current at time t18	ALTER DDL at time t10	Current at time t18	ALTER DDL at time t10	Data as just before DROP table at time t16

Action requested	Base DB2 semantics		Recovery Expert semantics			
			Existing table		Dropped table	
	DDL	Data	DDL	Data	DDL	Data
Recover table to object definition level corresponding to table creation (time t1)	Not supported	Not applicable	Table creation at time t1	Data as just before ALTER table at time t10	Table creation at time t1	Data as just before ALTER table at time t10
Recover table to object definition level corresponding to ALTER action (time t10)	Not supported	Not applicable	ALTER DDL at time t10	Current at time t18	ALTER DDL at time t10	Data as just before DROP table at time t16
Recover PARTITION to IC at time t4	ALTER DDL at time t10	IC at time t4	ALTER DDL at time t10	IC at time t4	Not applicable	Not applicable
Recover non-partitioned table space DSNUM to RBA at time t9	ALTER DDL at time t10	RBA at time t9	Not supported	Not applicable	Not applicable	Not applicable
Recover DROPped partitioned table space to the point where it was dropped by recovering to the object definition level corresponding to the table space (not table)	Not supported	Not applicable	Not applicable	Not applicable	Table space creation at time t16	Data as just before DROP of table space at time t16

a. For multi-table table space, only full image copy recovery supported for a DROPped table

For each scenario, Table 1-1 shows the semantics of the requested recovery action in the base DB2 for z/OS engine, and Recovery Expert for z/OS semantics when recovering an existing object or a dropped object.

- Recover table to the image copy taken at time t4

  The recovery action is a point-in-time to an image copy at time t4 which is before the time when the table was altered at time t10. The semantics at the completion of the recovery action are:

  – In base DB2 for z/OS, the table structure is as it was after the table alteration at time t10, though the data corresponds to the image copy taken at time t4. DB2 for z/OS automatically handles the missing or transformation data values for the modified table structure. All indexes, views, synonyms, privileges, and so on remain as they were just before the recovery action was initiated.

  – In Recovery Expert for z/OS, when the recovery action is for the existing table (assuming no drop at time t16), the table structure is as it was at table creation at time t1, though the data corresponds to the image copy taken at time t4. View V1 is restored but not views V2, V3, V4, and V5 as per the rationale described in “Algorithm for restoring object dependencies and data” on page 22. The same applies to indexes, synonyms, and so on.

> **Note:** Recovery Expert for z/OS drops the existing table, and re-creates it (using the information it has in the SLR) and its dependencies such as indexes, views, and synonyms as per the rationale described in "Algorithm for restoring object dependencies and data" on page 22. However, it does *not* restore any of the privileges that existed on the table and views; it is your responsibility to re-grant these privileges. The SLR tables maintain information about grants, and you must be able to retrieve this information to restore the required privileges.

- In Recovery Expert for z/OS, when the recovery action is for the DROPped table (assuming drop at time t16), the table structure is as it was at table creation at time t1, though the data corresponds to the image copy taken at time t4. In this case too, view V1 is restored but not views V2, V3, and V4 as per the rationale described in "Algorithm for restoring object dependencies and data" on page 22 (view V5 does not apply). The same applies to indexes, synonyms, and so on.

> **Note:** Recovery Expert for z/OS re-creates the DROPped table (using the information it has in the SLR) and its dependencies such as indexes, views, and synonyms as per the rationale described in "Algorithm for restoring object dependencies and data" on page 22. However, it does *not* restore any of the privileges that existed on the table and views; it is your responsibility to re-grant these privileges. The SLR tables maintain information about grants, and you must be able to retrieve this information to restore the required privileges.

- Recover table to the quiesce point taken at time t7

  The recovery action is a PIT to a quiesce point at time t7 which is before the time when the table was altered at time t10. The semantics at the completion of the recovery action are:

  - In base DB2 for z/OS, the table structure is as it was after the table alteration at time t10, though the data corresponds to the RBA corresponding to the quiesce point taken at time t7. DB2 for z/OS automatically handles the missing or transformation data values for the modified table structure. All indexes, views, synonyms, privileges, and so on remain as they were just before the recovery action was initiated.
  - In Recovery Expert for z/OS, when the recovery action is for the existing table (assuming no drop at time t16), the table structure is as it was at table creation at time t1, though the data corresponds to the RBA corresponding to the quiesce point taken at time t7. View V1 is restored, but not views V2, V3, V4, and V5 as per the rationale described in "Algorithm for restoring object dependencies and data" on page 22. The same applies to indexes, synonyms, and so on.

> **Note:** Recovery Expert for z/OS drops the existing table, and re-creates it (using the information it has in the SLR) and its dependencies such as indexes, views, and synonyms as per the rationale described in "Algorithm for restoring object dependencies and data" on page 22. However, it does *not* restore any of the privileges that existed on the table and views; it is your responsibility to re-grant these privileges. The SLR tables maintain information about grants, and you must be able to retrieve this information to restore the required privileges.

  - In Recovery Expert for z/OS, when the recovery action is for the DROPped table (assuming drop at time t16), the table structure is as it was at table creation at time t1, though the data corresponds to the RBA corresponding to the quiesce point taken at time t7. In this case too, view V1 is restored, but not views V2, V3, and V4 as per the rationale described in “Algorithm for restoring object dependencies and data” on page 22 (view V5 does not apply). The same applies to indexes, synonyms, and so on.

> **Note:** Recovery Expert for z/OS re-creates the DROPped table (using the information it has in the SLR) and its dependencies such as indexes, views, and synonyms as per the rationale described in “Algorithm for restoring object dependencies and data” on page 22. However, it does *not* restore any of the privileges that existed on the table and views; it is your responsibility to re-grant these privileges. The SLR tables maintain information about grants, and you must be able to retrieve this information to restore the desired privileges.

- Recover table to the image copy taken at time t13

  The recovery action is a PIT to an image copy at time t13, which is after the time when the table was altered at time t10. The semantics at the completion of the recovery action are:

  – In base DB2 for z/OS, the table structure is as it was after the table alteration at time t10, though the data corresponds to the image copy taken at time t13. All indexes, views, synonyms, privileges, and so on remain as they were just before the recovery action was initiated.

  – In Recovery Expert for z/OS, when the recovery action is for the existing table (assuming no drop at time t16), the table structure is as it was after the table alteration at time t10, though the data corresponds to the image copy taken at time t13. Views V1, V2, and V3 remain, although views V4 and V5 are dropped as per the rationale described in “Algorithm for restoring object dependencies and data” on page 22. The same applies to indexes, synonyms, and so on.

  – In Recovery Expert for z/OS, when the recovery action is for the DROPped table (assuming drop at time t16), the table structure is as it was after the table alteration at time t10, though the data corresponds to the image copy taken at time t13. In this case too, views V1, V2, and V3 are restored, but not view V4 as per the rationale described in “Algorithm for restoring object dependencies and data” on page 22 (view V5 does not apply). The same applies to indexes, synonyms, and so on.

> **Note:** Recovery Expert for z/OS re-creates the DROPped table (using the information it has in the SLR) and its dependencies such as indexes, views, and synonyms as per the rationale described in “Algorithm for restoring object dependencies and data” on page 22. However, it does *not* restore any of the privileges that existed on the table and views; it is your responsibility to re-grant these privileges. The SLR tables maintain information about grants, and you must be able to retrieve this information to restore the required privileges.

- Recover table to current at time t18

  The recovery action is to current at time t18. The semantics at the completion of the recovery action are:

  – In base DB2 for z/OS, the table structure is as it was after the table alteration at time t10, though the data corresponds to the values at time t18. All indexes, views, synonyms, privileges, and so on remain as they were just before the recovery action was initiated.

  – In Recovery Expert for z/OS, when the recovery action is for the existing table (assuming no drop at time t16), the table structure is as it was after the table alteration at time t10, though the data corresponds to the values at time t18. All indexes, views, synonyms, privileges, and so on remain as they were just before the recovery action was initiated.

– In Recovery Expert for z/OS, when the recovery action is for the DROPped table (assuming drop at time t16), the table structure is as it was after the table alteration at time t10, though the data corresponds to the values just before the drop of the table at time t16. Views V1, V2, and V3 remain, but not view V4 as per the rationale described in “Algorithm for restoring object dependencies and data” on page 22 (view V5 does not apply). The same applies to indexes, synonyms, and so on.

> **Note:** Recovery Expert for z/OS re-creates the DROPped table (using the information it has in the SLR) and its dependencies such as indexes, views, and synonyms as per the rationale described in “Algorithm for restoring object dependencies and data” on page 22. However, it does *not* restore any of the privileges that existed on the table and views; it is your responsibility to re-grant these privileges. The SLR tables maintain information about grants, and you must be able to retrieve this information to restore the required privileges.

- Recover table to object definition level corresponding to table creation at time t1

  The recovery action is a PIT to an object definition level at time t1, which corresponds to the table creation time. The semantics at the completion of the recovery action are:

  – Base DB2 for z/OS does not support such a concept.

  – In Recovery Expert for z/OS, when the recovery action is for the existing table (assuming no drop at time t16), the table structure is as it was at table creation at time t1, though the data corresponds to the point just before the table alteration at time t10. Views V1 and V2 are restored, but not views V3, V4, and V5 as per the rationale described in “Algorithm for restoring object dependencies and data” on page 22. The same applies to indexes, synonyms, and so on.

> **Note:** Recovery Expert for z/OS drops the existing table, and re-creates it (using the information it has in the SLR) and its dependencies such as indexes, views, and synonyms as per the rationale described in “Algorithm for restoring object dependencies and data” on page 22. However, it does *not* restore any of the privileges that existed on the table and views; it is your responsibility to re-grant these privileges. The SLR tables maintain information about grants, and you must be able to retrieve this information to restore the required privileges.

  – In Recovery Expert for z/OS, when the recovery action is for the DROPped table (assuming drop at time t16), the table structure is as it was at table creation at time t1, though the data corresponds to the point just before the table alteration at time t10. In this case too, views V1 and V2 are restored, but not views V3 and V4 as per the rationale described in “Algorithm for restoring object dependencies and data” on page 22 (view V5 does not apply). The same applies to indexes, synonyms, and so on.

> **Note:** Recovery Expert for z/OS re-creates the DROPped table (using the information it has in the SLR) and its dependencies such as indexes, views, and synonyms as per the rationale described in “Algorithm for restoring object dependencies and data” on page 22. However, it does *not* restore any of the privileges that existed on the table and views; it is your responsibility to re-grant these privileges. The SLR tables maintain information about grants, and you must able to retrieve this information to restore the required privileges.

- Recover table to object definition level corresponding to table alteration at time t10

  The recovery action is a PIT to an object definition level at time t10 which corresponds to the table alteration time. The semantics at the completion of the recovery action are:

  – Base DB2 for z/OS does not support such a concept.

  – In Recovery Expert for z/OS, when the recovery action is for the existing table (assuming no drop at time t16), the table structure is as it was at table alteration at time t10, though the data corresponds to current at time t18. This corresponds to a recovery to current for Recovery Expert for z/OS. Views V1, V2, V3, V4, and V5 remain. The same applies to indexes, synonyms, and so on.

  – In Recovery Expert for z/OS, when the recovery action is for the DROPped table (assuming drop at time t16), the table structure is as it was at table alteration at time t10, though the data corresponds to the point just before the drop at time t16. Views V1, V2, V3 are restored, but not view V4 as per the rationale described in “Algorithm for restoring object dependencies and data” on page 22 (view V5 does not apply). The same applies to indexes, synonyms, and so on.

  Note: Recovery Expert for z/OS re-creates the DROPped table (using the information it has in the SLR) and its dependencies such as indexes, views, and synonyms as per the rationale described in “Algorithm for restoring object dependencies and data” on page 22. However, it does *not* restore any of the privileges that existed on the table and views; it is your responsibility to re-grant these privileges. The SLR tables maintain information about grants, and you must be able to retrieve this information to restore the required privileges.

- Recover a partition to the image copy at time t4

  The recovery action is a PIT of a single partition of partitioned table space to an image copy at time t4, which is before the time when the table was altered (at time t10). The semantics at the completion of the recovery action are:

  – In base DB2 for z/OS, the table structure is as it was after the table alteration at time t10, though the data corresponds to the image copy taken at time t4. DB2 for z/OS automatically handles the missing or transformation data values for the modified table structure. All indexes, views, synonyms, privileges, and so on remain as they were just before the recovery action was initiated.

  – In Recovery Expert for z/OS, when the recovery action is for the existing table (assuming no drop at time t16), the semantics are the same as for base DB2: The table structure is as it was after the table alteration at time t10, though the data corresponds to the image copy taken at time t4. All indexes, views, synonyms, privileges, and so on remain as they were just before the recovery action was initiated.

  – In Recovery Expert for z/OS, this recovery action is not supported because it is not possible to just drop a partition in the DB2 for z/OS engine.

- Recover a data set of a non-partitioned table space to an RBA at time t9

  The recovery action is a PIT of a data set of a multi data set non-partitioned table space to an RBA at time t9. The semantics at the completion of the recovery action are:

  – In base DB2 for z/OS, the table structure is as it was after the table alteration at time t10, though the data corresponds to the image copy taken at time t4. DB2 for z/OS automatically handles the missing or transformation data values for the modified table structure. All indexes, views, synonyms, privileges, and so on remain as they were just before the recovery action was initiated.

  – In Recovery Expert for z/OS, this semantics is not supported.

- Recover a dropped partitioned table space to the point just before the drop

  This is achieved by recovering to the latest object definition level for the table space as recorded in the SLR. The semantics at the completion of the recovery action are:

  – Base DB2 for z/OS does not support such a concept.

  – In Recovery Expert for z/OS, when the recovery action is for the DROPped partitioned table space, the table structure is as it was at table alteration at time t10, though the data corresponds to the point just before the drop at time t16. Views V1, V2, and V3 are restored, but not view V4 as per the rationale described in “Algorithm for restoring object dependencies and data” (view V5 does not apply). The same applies to indexes, synonyms, and so on.

> **Note:** Recovery Expert for z/OS re-creates the DROPped table (using the information it has in the SLR) and its dependencies such as indexes, views, and synonyms as per the rationale described in “Algorithm for restoring object dependencies and data”. However, it does *not* restore any of the privileges that existed on the table and views; it is your responsibility to re-grant these privileges. The SLR tables maintain information about grants, and you must be able to retrieve this information to restore the required privileges.

Algorithm for restoring object dependencies and data

The key to understanding the algorithm for restoring object dependencies is the start and end timestamp stored in the SLR for a particular object definition level. With reference to Figure 1-11 on page 16, when a new object is created at time t1 and the SLR update is run at time t2, an entry is created in the SLR with a start timestamp corresponding to time t1, and an end timestamp that is open-ended (blank).

When an ALTER is done to the object at time t10 and an SLR update is performed at time t11, the following events occur:

1. A new entry is created for this object in the SLR with a start timestamp of t11 and an end timestamp that is open-ended (blank).
2. The earlier object entry that had a start timestamp of t1 and an open-ended timestamp (blank), is now provided an end timestamp that is one microsecond less than timestamp t11, in other words, it has an end timestamp of t11 - 1 microsecond.

There are now two object definition levels (or versions for ease of understanding) for this object in the SLR: The first one that is bounded by a start timestamp t1 and end timestamp (t11 - 1 microsecond), and the current one which is bounded by a start timestamp t11 and open-ended (blank) as far as the end timestamp is concerned.

When deciding what objects to restore or drop when a PIT recovery action is requested, Recovery Expert for z/OS performs the following tasks:

1. Determines what object definition level (version) applies to this PIT recovery action, for example, when the recovery action is to the image copy at time t4 in Figure 1-11 on page 16, the object definition level boundary is the one corresponding to the start timestamp t1 and end timestamp (t11 - 1 microsecond).
2. Determines the objects and dependencies that it knows about in this bounded interval, in effect this corresponds to the information collected by the SLR updates that are run in this interval. For the bounded interval corresponding to this object definition level, the SLR updates were performed at time t2 and t5. Therefore, in this interval, the SLR only knows about views V1 and V2.

3. Determines the objects defined before the chosen recovery point and only restore those objects. For the chosen recovery point of the image copy taken at time t4, the only object restored is view V1, which is known in the SLR as created before time t4. View V2 is not restored because it is known in the SLR as created after time t4.

Note: For the object definition recovery point corresponding to time t11 (when the SLR was updated) after an accidental drop of the table, view V4 is not restored. This is because the SLR update that is performed after the drop does not have information about view V4 recorded in the SLR. View V4 is dropped when the table was dropped at time t16 and is not available to be recorded when the SLR update was performed at time t17. View V5 does not apply because no table exists after the drop.

Attention: The boundary interval *also* applies to the data when the recovery point is an object definition level. Therefore, when the recovery point is to the object definition level corresponding to table creation at time t1, the Data Manipulation Language (DML) updates that occur between time t10 and t11 are considered for inclusion in the data, even though in the scenario these updates are performed on an updated DDL version (after time t10), but before the SLR update that was performed at time t11.

1.4 Processing flow

This section describes the processing flow involved during configuration, starting, and stopping the DB2 Recovery Expert for z/OS environment, and the runtime execution of the DB2 Recovery Expert for z/OS environment.

1.4.1 Configuration flow

The steps involved in configuring DB2 Recovery Expert for z/OS are well documented in Chapter 3, “Configuring DB2 Recovery Expert for z/OS,” of the manual *DB2 Recovery Expert for z/OS, V1R1, User's Guide*, SC18-9822.

As shown in Figure 1-1 on page 3, the various configuration files such as the server configuration file, the agent configuration file, and the product control file have to be correctly configured for successful operation of the environment. The configuration can be either manual configuration or automatic discovery.

Manual configuration

The client-server configuration uses a different method from the method for configuring the server-agent configuration. This configuration is described in the following section.

Client-server configuration

You must configure the DB2 Recovery Expert for z/OS server to run on a particular host machine. You can additionally configure the server with a TCP/IP port number on which to listen for incoming connection requests from instances of the client. This value is specified by the client-listener-port option (default value 9875) in the server configuration file, as shown in Example 1-1 on page 4. When you start the DB2 Recovery Expert for z/OS GUI client, manually enter the host name or IP address on which the DB2 Recovery Expert for z/OS server is running, and the port number on which that server is configured to listen for client connections.

Server-agent configuration

In addition to configuring the DB2 Recovery Expert for z/OS server for incoming client requests, you must specify a different port on which to listen for incoming connection requests from instances of the agent. This value is specified using the agent-listener-port configuration option (default value 9876) in the server configuration file, as shown in Example 1-1 on page 4.

You must configure an instance of the agent to run on every machine hosting a DB2 subsystem that is accessed by users. When you configure each agent instance, specify the host name or IP address and the port number on which the server is running. These values are specified using the server-address and server-port configuration options in the agent configuration file, as shown in Example 1-2 on page 5. When the DB2 Recovery Expert for z/OS agent is started, it uses this configuration information to connect to that particular DB2 Recovery Expert for z/OS server.

Automatic discovery

You can configure the server, clients, and agents to connect to each other automatically. Automatic discovery eases the manual configuration effort that is required and allows users to connect to the server even if they do not know the specific connection information.

When the DB2 Recovery Expert for z/OS server starts, it begins sending notifications in the network that it is up and running. These notifications include the port numbers on which it is listening for incoming client and agent connections, as specified in the server configuration file shown in Example 1-1 on page 4. The notifications are sent using IP multicasting. Briefly, multicasting is similar in concept to broadcasting, where a message is potentially sent to every destination address on a network. However, with multicasting, instead of sending messages to every address on a network, messages are sent only to addresses that register interest in those particular messages.

In the case of DB2 Recovery Expert for z/OS, the client and agent register interest in the messages sent by the server, therefore the server notifications are routed to them. There are a wide variety of options related to the use of multicasting in IP network software and hardware.

> **Attention:** In some cases, it is possible that due to software or hardware configuration, multicast packets are not forwarded from the machine on which the server is running to the machines on which one or more clients or agents are running. You must consult your network administrator and configuration documentation to ensure that proper forwarding of multicast packets is enabled.

The exact configuration process to use these server notifications differs between client-server and agent-server connections.

Client-server connections

When the DB2 Recovery Expert for z/OS GUI client starts, you are prompted to enter the host name or IP address and port number of the server to which you want to connect. In addition to manually entering a value, you can also select from a list of values that you specified during prior sessions, as shown in Figure 1-2 on page 7.

Rather than entering a value or selecting a recently used value, you can click the browse button (**...**) in the login window shown in Figure 1-2 on page 7. This opens the Select Server window shown in Figure 1-12.

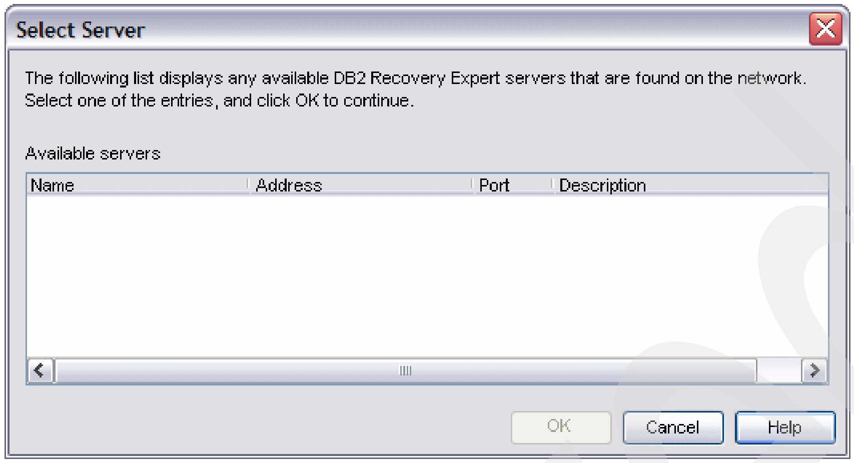

Figure 1-12 Select Server window

Figure 1-12 lists all of the currently running instances of DB2 Recovery Expert for z/OS server, as detected by listening for server notifications in the network. You can select any one of the listed server instances, and the DB2 Recovery Expert for z/OS client uses the connection information in the notification to connect to that server. If no servers are detected, no servers are listed as is the case in Figure 1-12. Click **Cancel** to return to the login window and manually enter a host name and port number, or choose one from the list of values that you specified during prior sessions, as shown in Figure 1-2 on page 7.

Agent-server connections

As described previously, you can specify the host name and port number of the server to which an agent connects in the agent configuration file, as shown in Example 1-2 on page 5. If you *omit* the server-address configuration option, the agent listens for server notifications in the network.

When a notification for a DB2 Recovery Expert for z/OS server is received, the agent uses the information in the notification to connect to that server if the community-string value specified in the agent configuration file matches the community-string in the server notification. If you specify a value for the server-address configuration option in the agent configuration file, as shown in Example 1-2 on page 5 (which has a value of wtsc59.itso.ibm.com), the agent attempts to connect to that address, and does *not* perform any automatic server discovery.The administrator uses automatic discovery to change which machine the server runs on and does not have to perform changes in the agent configuration files.

1.4.2 Starting the DB2 Recovery Expert for z/OS environment

After you configure the server and agents, start the server, agents, and clients in the following sequence:

1. Start the DB2 Recovery Expert for z/OS server.

   The DB2 Recovery Expert for z/OS server must be running for DB2 Recovery Expert for z/OS users to perform any functions against the DB2 subsystems.

DB2 Recovery Expert for z/OS server runs as a batch job under MVS™, which you can submit manually, schedule automatically, or run as an MVS started task. Issue the following command from the MVS console if DB2 Recovery Expert for z/OS server is defined as a started task:

```
/s ARYSRV53
```

In this command, ARYSRV53 is the name of the member in your installation's PROCLIB library (typically SYS1.PROCLIB).

> **Important:** You must start the DB2 Recovery Expert for z/OS server before starting any of the agents.

The primary output of the DB2 Recovery Expert for z/OS server job is log messages that provide status information about the ongoing operation of the server. The log messages are written to the SYSPRINT DD. In addition to providing status information about the operation of the server, these messages record if and when errors occur. In the event of exceptional conditions, additional messages might be written to the SYSOUT DD. If an abend occurs, dump information might be written to the CEEDUMP, SYSUDUMP DDs, or to both. Product support can use this information for diagnosis.

2. Start the DB2 Recovery Expert for z/OS agents.

   The DB2 Recovery Expert for z/OS agent provides access to database and system services in support of the DB2 Recovery Expert for z/OS server and remote clients. You must run one instance of the agent on every system or LPAR that hosts DB2 subsystems or data sharing groups, which you want to access with DB2 Recovery Expert for z/OS. Each agent communicates with the DB2 Recovery Expert for z/OS server to provide services. The agent must be running for DB2 Recovery Expert for z/OS users to perform any functions against DB2 subsystems on that LPAR.

   As in the case of the DB2 Recovery Expert for z/OS server, the DB2 Recovery Expert for z/OS agent runs as a batch job under MVS. You can manually submit this job, schedule it to be run automatically, or run it as an MVS started task. Issue the following command from the MVS console if DB2 Recovery Expert for z/OS agent is defined as a started task:

   ```
   /s ARYAGT53
   ```

   In this command, ARYAGT53 is the name of the member in your installation's PROCLIB library (typically SYS1.PROCLIB).

   > **Important:** You must start the DB2 Recovery Expert for z/OS server before starting any of the agents.

   The primary output of the DB2 Recovery Expert for z/OS agent job is log messages that provide status information about the ongoing operation of the agent. The log messages are written to the SYSPRINT DD. In addition to providing status information about the operation of the agent, these messages record if and when errors occur. In the event of exceptional conditions, additional messages might be written to the SYSOUT DD. If an abend occurs, dump information might be written to the CEEDUMP, SYSUDUMP DDs, or to both. Product support can use this information for diagnosis.

3. Start the DB2 Recovery Expert for z/OS GUI clients.

   To launch the DB2 Recovery Expert for z/OS client, from the Windows system, click **Start** → **Programs** → **IBM DB2 Recovery Expert**. It detects available DB2 Recovery Expert for z/OS servers and lists them in a server list, as shown in Figure 1-2 on page 7. Further processing occurs as described in the list item "Recovery Expert for z/OS GUI client" on page 7.

1.4.3 Stopping the DB2 Recovery Expert for z/OS environment

You can stop the agents and clients independent of the server, but stopping the server requires that you quiesce or force off the agents and clients before you stop the server.

1. Stop the DB2 Recovery Expert for z/OS server.

   The DB2 Recovery Expert for z/OS server accepts standard MVS /MODIFY and /STOP commands (from System Display and Search Facility (SDSF) or elsewhere that supports them) and provides multiple options for shutting down the server as follows:

   – Graceful shutdown of the DB2 Recovery Expert for z/OS server using any of the following commands:

     ```
     /MODIFY ARYSRV53,STOP SERVER
     /MODIFY ARYSRV53,STOP
     /STOP ARYSRV53
     ```

     The termination of Recovery Expert for z/OS stops if a Recovery Expert for z/OS GUI client is still connected. A force might be more appropriate to shut down Recovery Expert for z/OS.

   – Stop the server and all agents currently connected to it using the command:

     ```
     /MODIFY ARYSRV53,STOP ALL
     ```

   – Force the server and all agents currently connected to it to drop immediately using the command:

     ```
     /MODIFY ARYSRV53,FORCE
     ```

   – Terminate the server using the following command:

     ```
     /p ARYSRV53
     ```

   You can also use the /MODIFY ARYSRV53,QUIESCE command to quiesce the server (continue servicing existing connections but not accept new ones), and the /MODIFY ARYSRV53,RESUME command to resume the quiesced server.

2. Stop the DB2 Recovery Expert for z/OS agents.

   You can stop a single agent associated with a DB2 Recovery Expert for z/OS server or all the agents associated with it as follows:

   – Stop an individual agent:

     ```
     /MODIFY ARYSRV53,STOP AGENTS id
     ```

     You can determine the required ID value by issuing the following command:

     ```
     /MODIFY ARYSRV53,LIST SESSIONS
     ```

     You can also stop an agent by issuing the following command:

     ```
     /p ARYAGT53
     ```

     In this command, ARYAGT53 is the name of the member in your installation's PROCLIB library (typically SYS1.PROCLIB).

   – Stop all the agents associated with a particular DB2 Recover Expert server:

     ```
     /MODIFY ARYSRV53,STOP AGENTS
     ```

3. Stop the DB2 Recovery Expert for z/OS clients.

   Exiting the DB2 Recovery Expert for z/OS client application on Windows causes its connection to the DB2 Recovery Expert for z/OS server to be terminated.

1.4.4 Runtime flow

This section briefly documents the processing that occurs when a database administrator (DBA[4]) logs on to a DB2 Recovery Expert for z/OS server from a DB2 Recovery Expert for z/OS GUI client, and navigates the menu to perform functions such as recovery on objects and log analysis.

Attention: We assume that the DBAs have all the prerequisite privileges on the DB2 Recovery Expert for z/OS packages and plans, and on the underlying objects in the DB2 subsystem that they are attempting to manage. By default, the DB2 Recovery Expert for z/OS packages and plans are granted access to PUBLIC during product installation. It is the DBAs' responsibility to ensure they have the necessary privileges on the DB2 objects in the appropriate DB2 subsystem which they expect to manage. The validate option does *not* check whether the user ID in the recovery plan has the necessary privileges to perform the various actions defined in the plan.

When DBAs log on to a DB2 Recovery Expert for z/OS server, they are presented with a main menu, as shown in Figure 1-3 on page 8. This section reviews the recovery and log analysis functions.

Recovery

When you click the **Recovery** button, you see a navigation window that lists all the currently active DB2 subsystems that the DB2 Recovery Expert for z/OS agent is authorized to manage, as shown in Figure 1-13.

[4] In this context, DBA refers to a person with appropriate skills to perform the tasks described. We recognize that the responsibilities of a person designated as a DBA in your organization might vary from those described here.

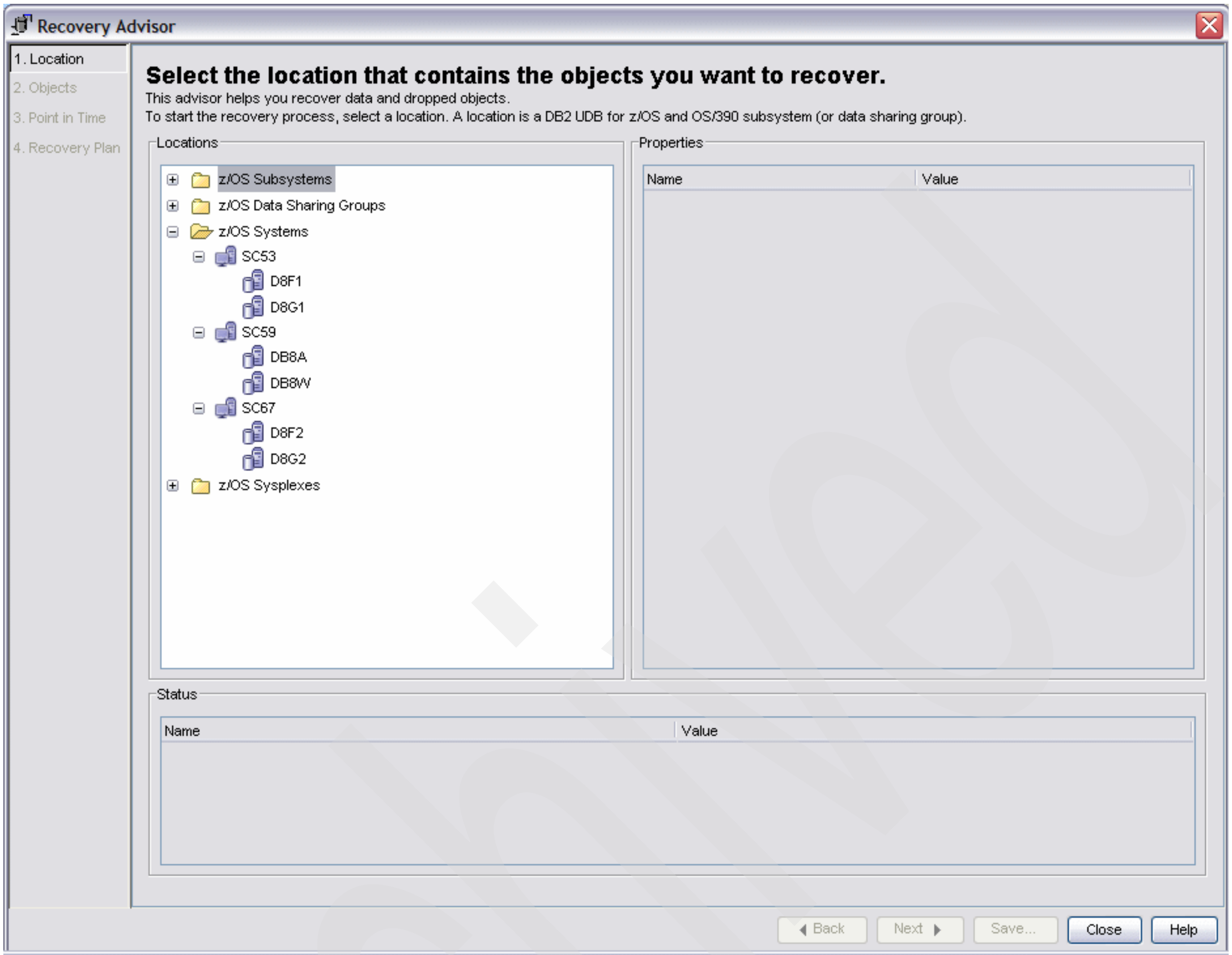

Figure 1-13 Navigation pane including active DB2 subsystems managed by the DB2 Recovery Expert for z/OS server

When the DB2 subsystem of interest is accessible by the DBAs, they can perform recovery functions on the DB2 objects in DB2 subsystem using the navigation pane. After selecting one or more DB2 objects to be recovered, they can select custom options selected for the recovery process, and generate and present a list of potential recovery plans in an ascending order of the expected cost of executing a particular recovery plan.

You (the DBA) can review the choices presented and choose the most appropriate recovery plan for execution. You can view the generated JCL for the recovery plan and make changes, if appropriate. You can also validate the recovery plan that causes DB2 Recovery Expert for z/OS to verify that all the objects listed in the recovery plan exist. For example, it ensures that if a point-in-time recovery is requested to a particular image copy, then that image copy is cataloged.

The validate option also checks for the existence of expected DB2 objects to ensure that objects that were dropped at the time of recovery plan generation are still dropped, and objects which existed still exist. It reduces the possibility that the generated JCL will fail at run time due to missing resources. However, this validation is not a full simulation, and therefore, a successful validation of the chosen recovery plan does *not* guarantee successful runtime execution because other conditions such as the DBA not having the required privileges on the target objects causes the runtime execution to fail.

On successful validation, you see the message shown in Figure 1-14, which makes it explicit that there is no guarantee of success.

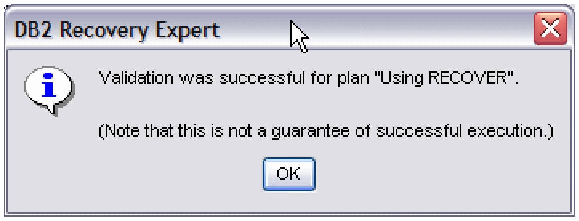

Figure 1-14 Validation successful message with caveat

You can start the selected recovery plan after the validation is successful.

> **Note:** Running the selected recovery plan when validation has failed can cause unpredictable results.

The status of the execution is recorded in the Messages portion of the main menu, as shown in Figure 1-3 on page 8. You can view the job output within the GUI or through z/OS based on your job card setup (typically using SDSF).

Log analysis

To analyze database log files to determine points of consistency, or quiet times, for objects, click the **Log Analysis** button (Figure 1-15 on page 31). After you select the location (Figure 1-16 on page 32), you can specify the log range to be considered (Figure 1-17 on page 33), which might be a start and end time or a duration preceding the current time.

> **Important:** The default time corresponds to the time on the Recovery Expert for z/OS GUI client. Take care to ensure that the time specified for the log range to be considered for quiet time detection must correspond to that of the target DB2 subsystem.

You can also specify the minimum duration (the default is 00:02:00, that is, 2 minutes) of the quiet time to be included in the output. You can use the SYSLGRNX option to optimize the log files that have to be read. Do *not* use this option if errors occur, or if the overhead of using it outweighs the savings it provides. Select the objects to look for (Figure 1-18 on page 34) and initiate log analysis. The quiet times that are detected by log analysis are written to SLR tables and job output, as shown in Figure 1-19 on page 35.

Select the quiet times that are detected by log analysis and written to SLR as a recovery point, as shown in Figure 1-20 on page 36 through Figure 1-25 on page 41. The quiet time is displayed as a timestamp recovery point.

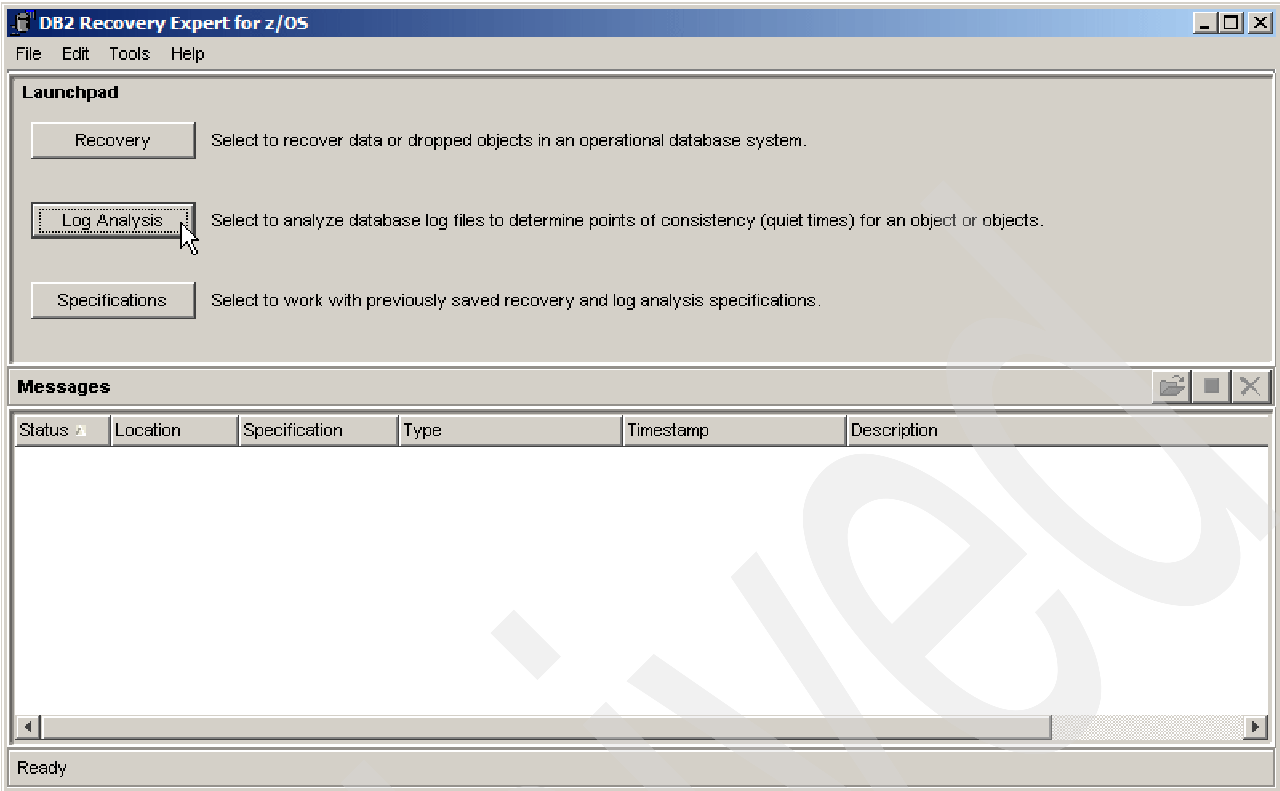

Figure 1-15 Launchpad menu: Selecting log analysis

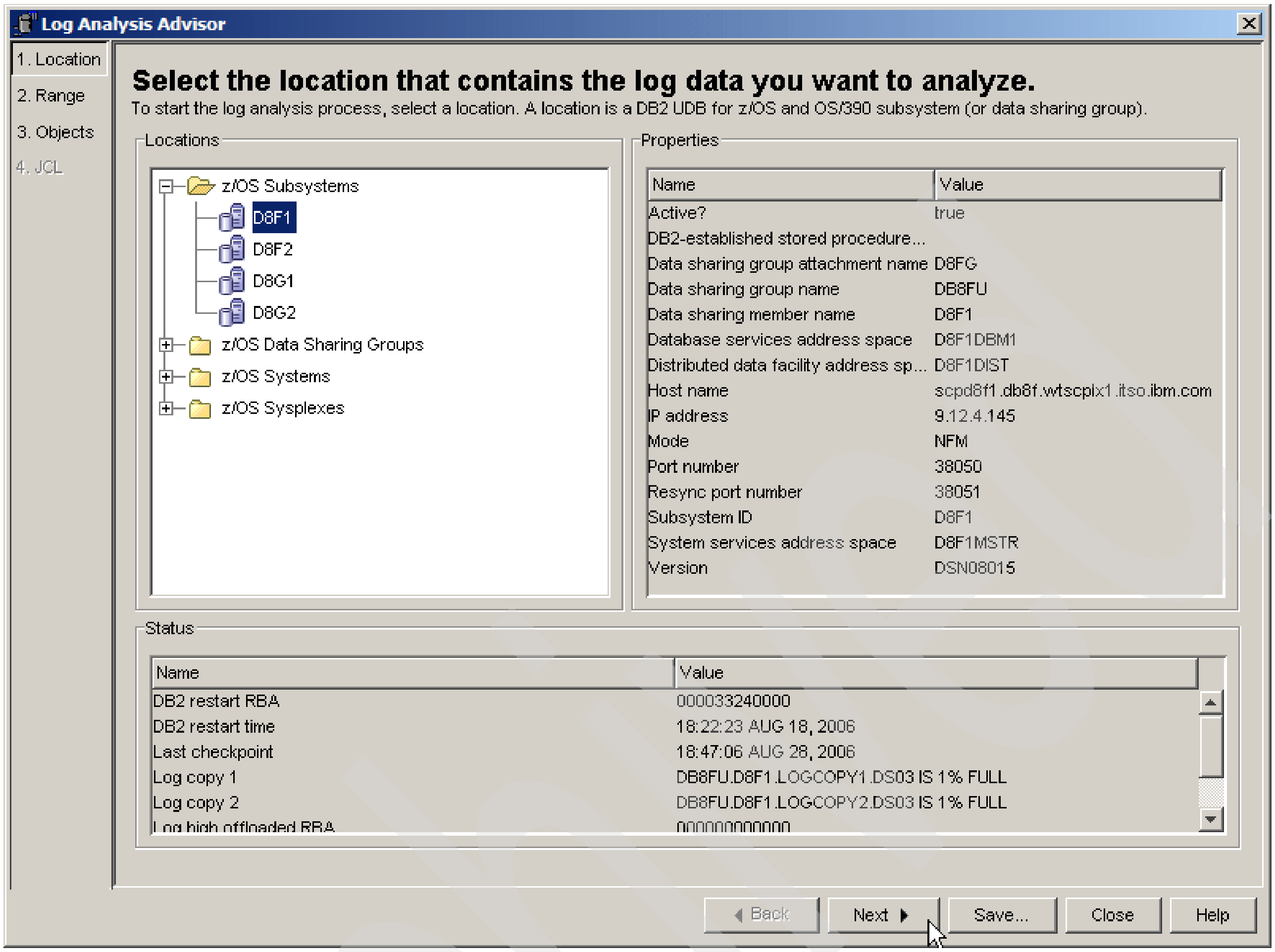

Figure 1-16 Selecting the location

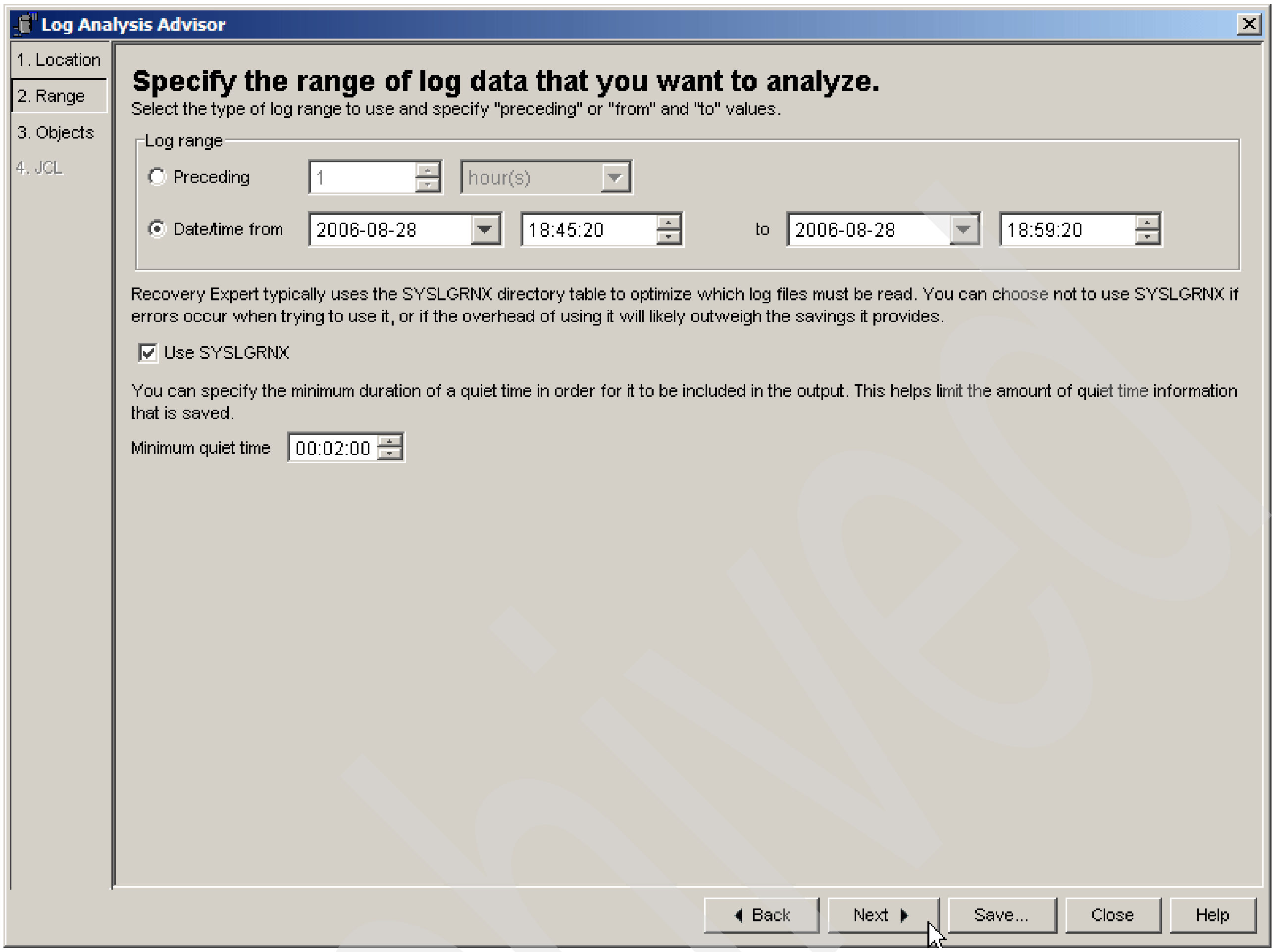

Figure 1-17 Specifying log range to analyze

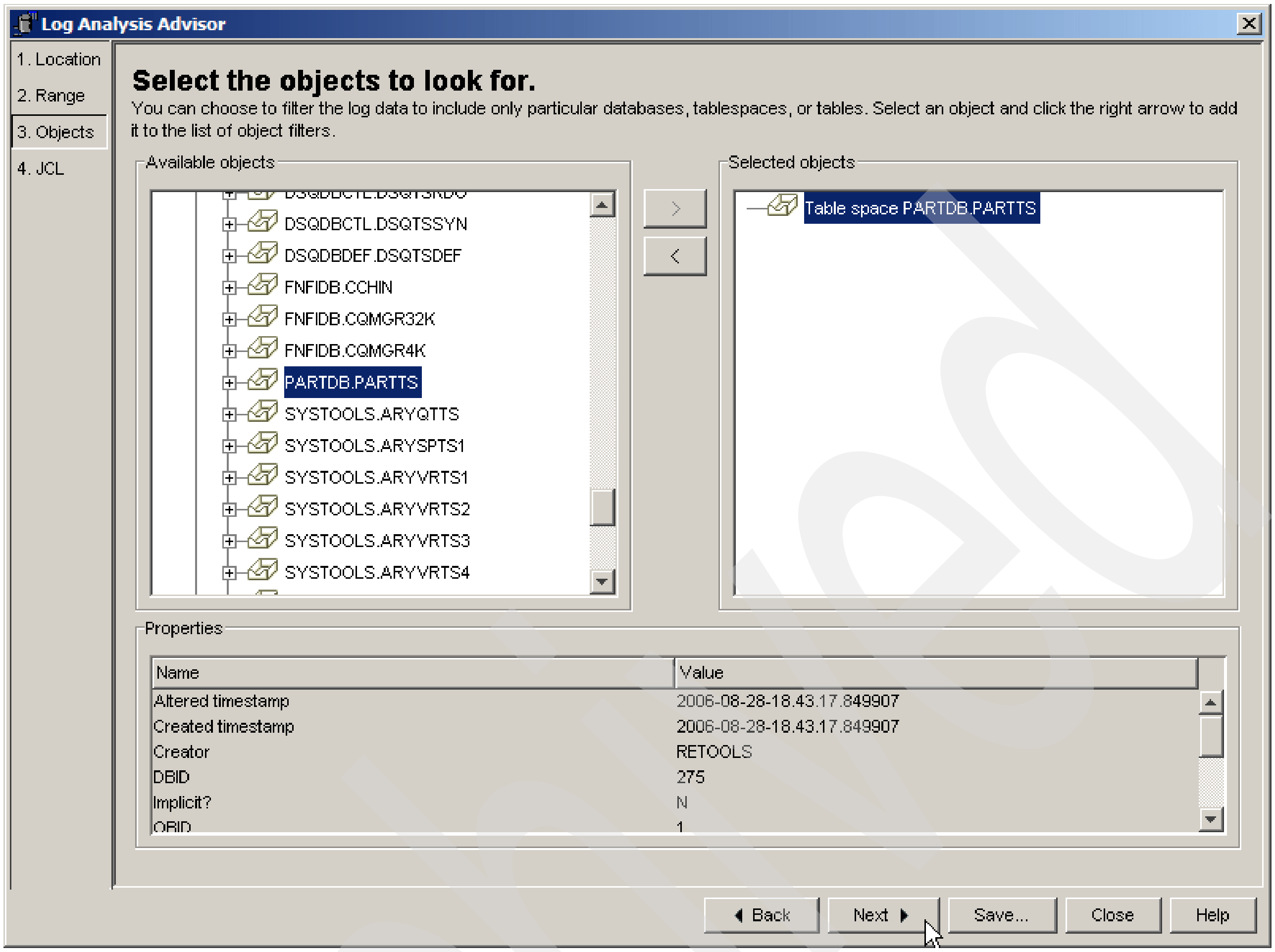

Figure 1-18 Selecting the objects to look for

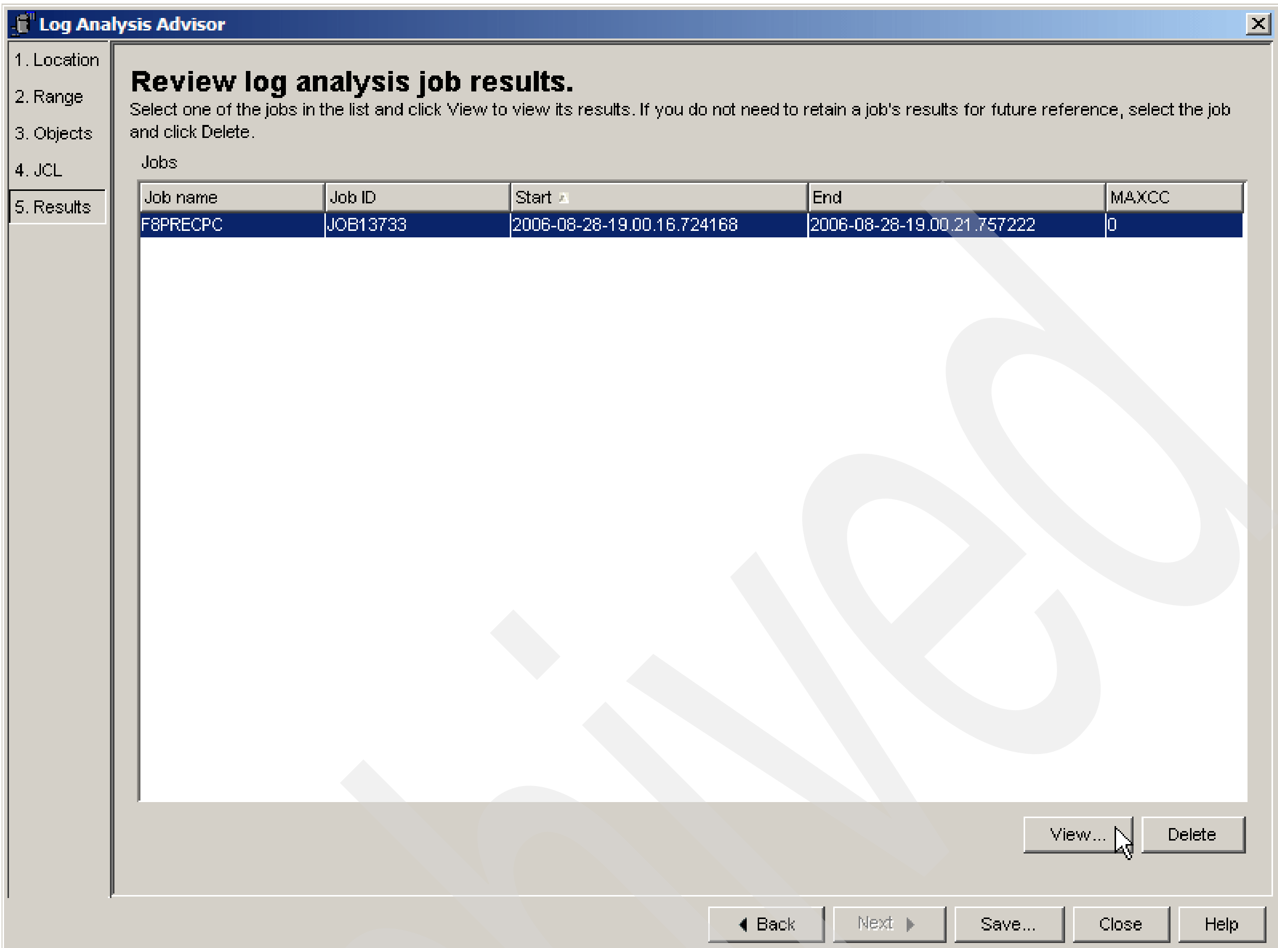

Figure 1-19 Reviewing log analysis job results

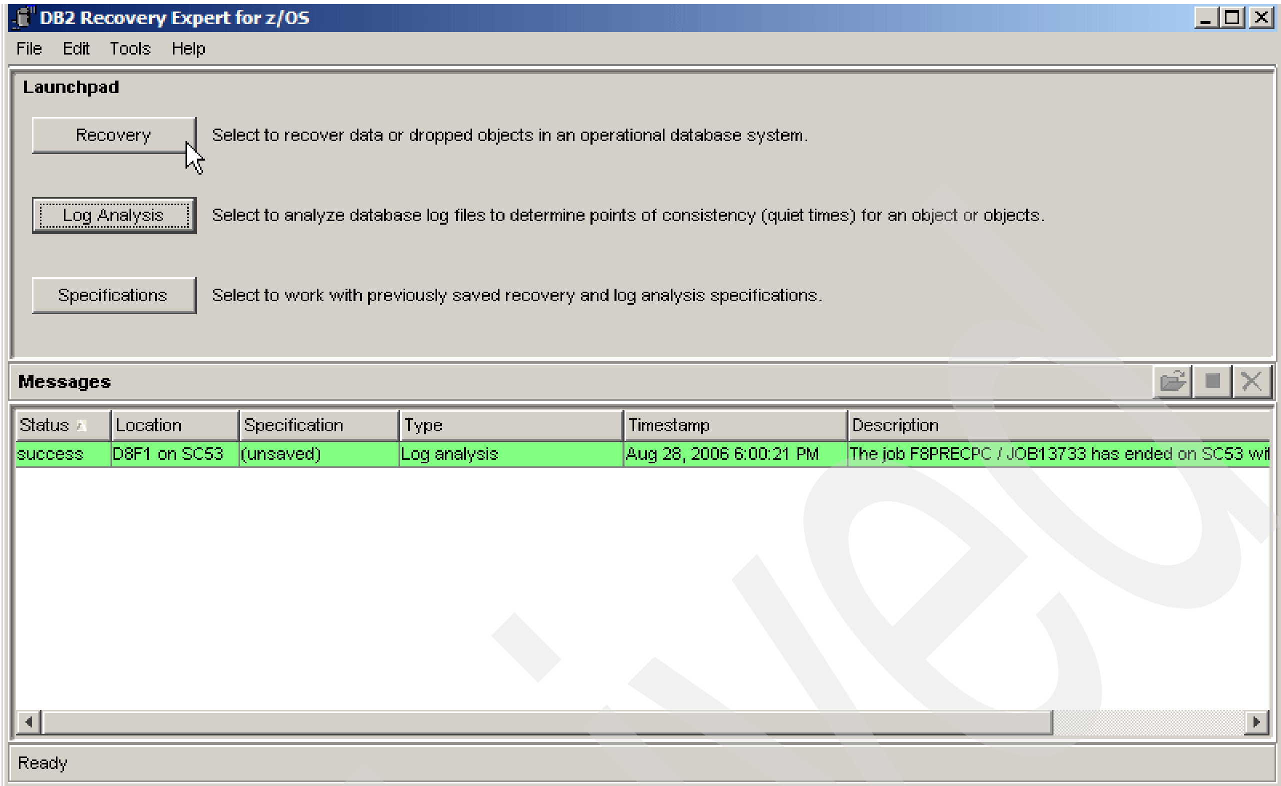

Figure 1-20 Launchpad with message pane and launching recovery

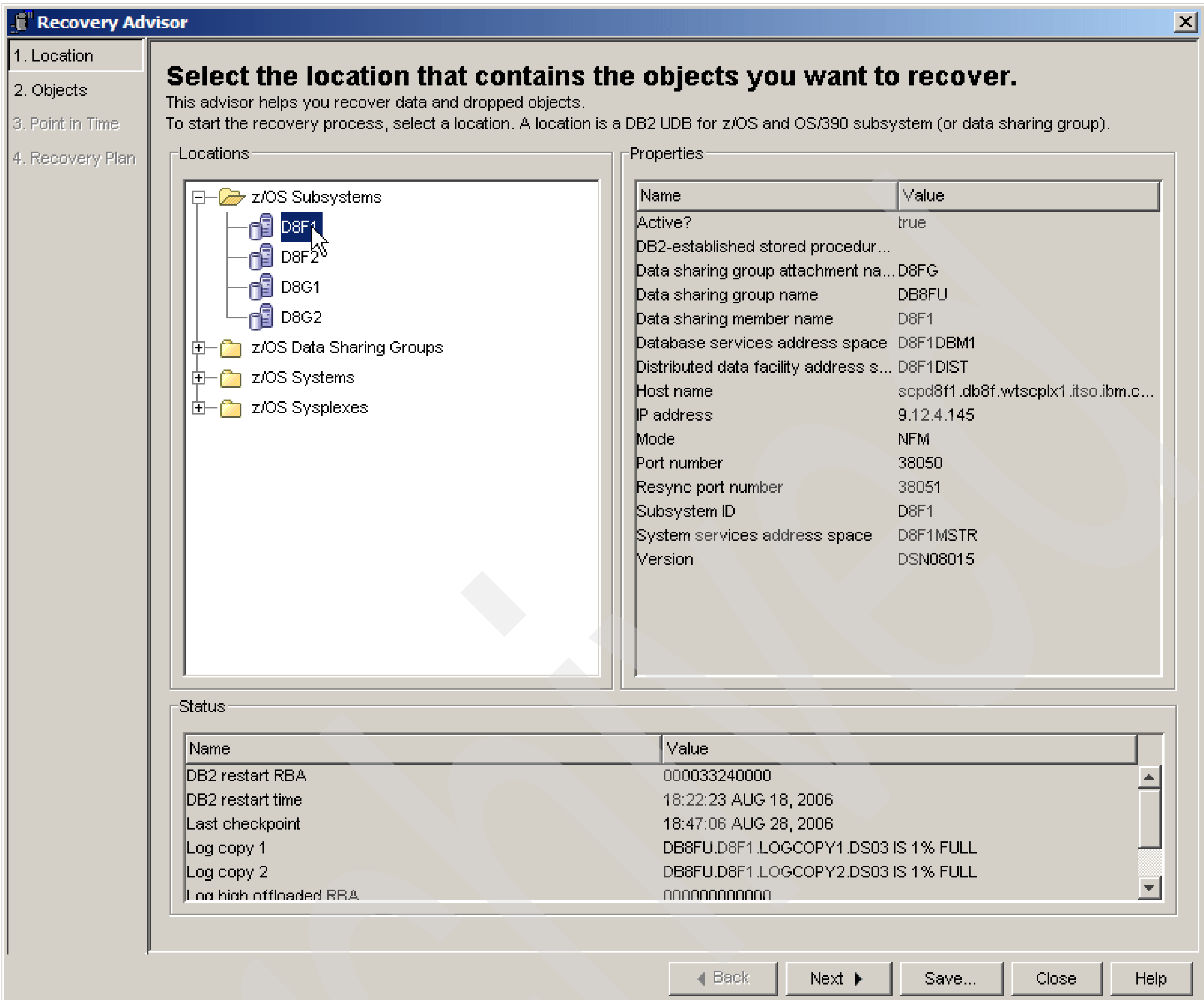

Figure 1-21 Selecting the location

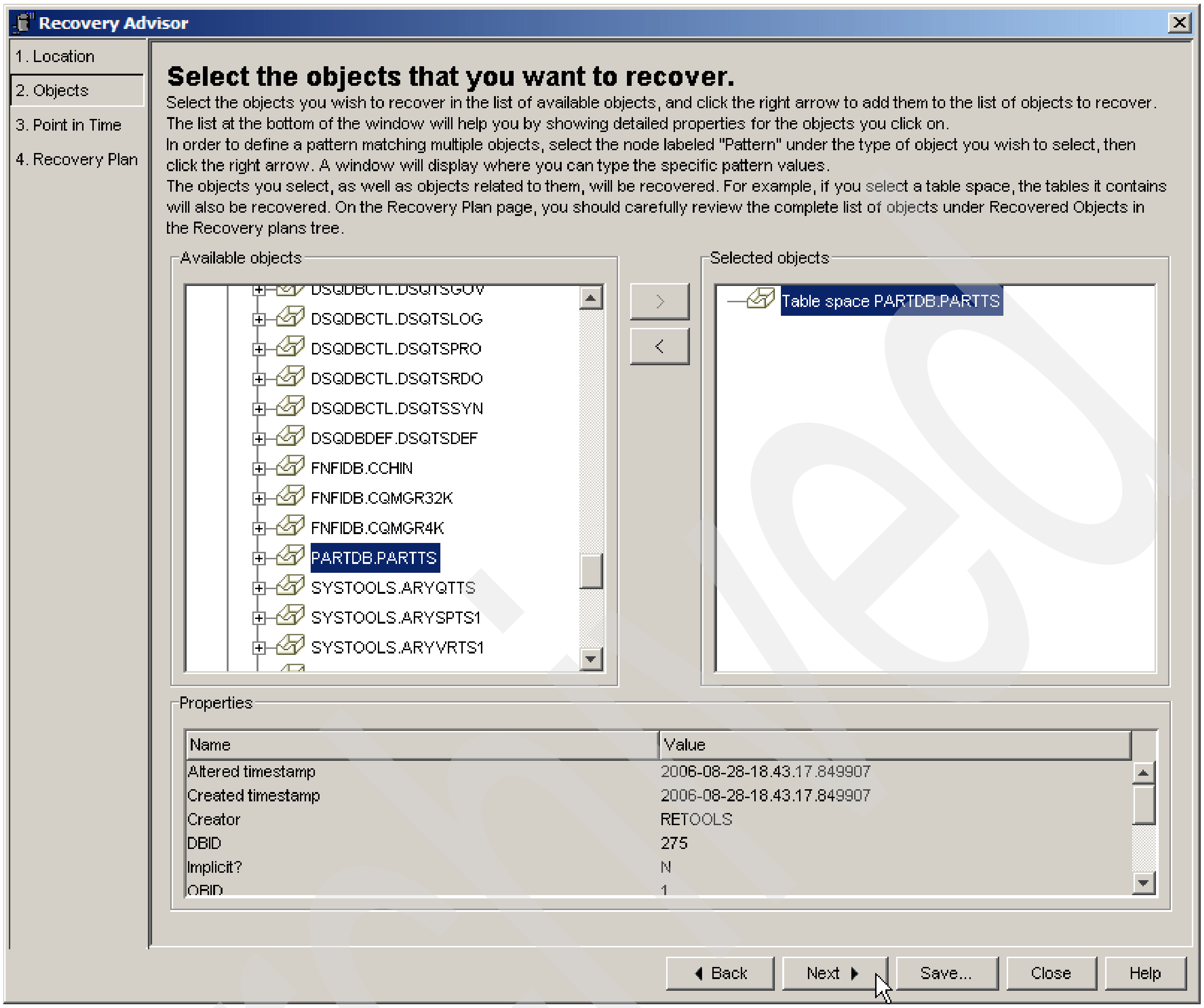

Figure 1-22 Selecting the objects

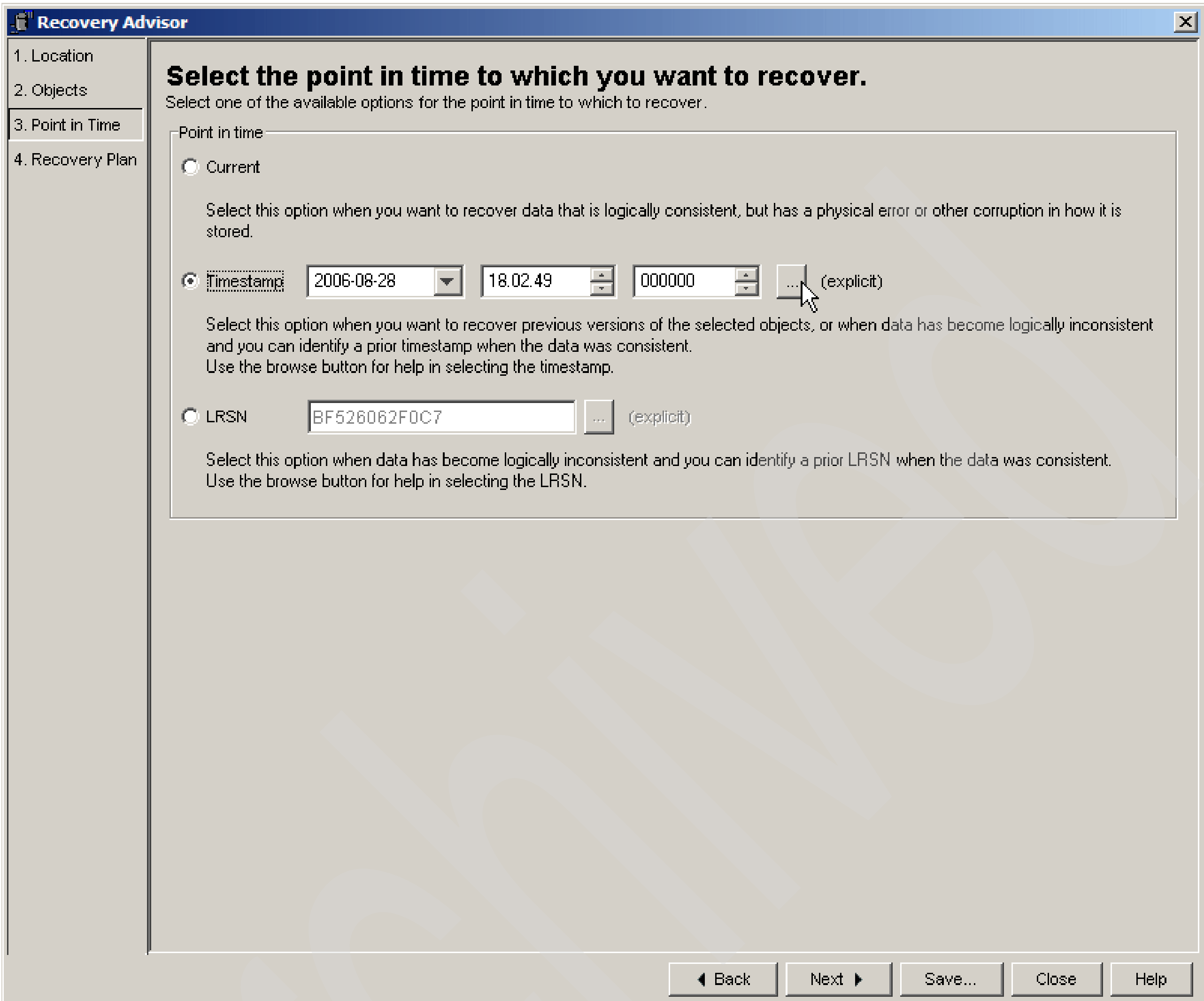

Figure 1-23 Selecting the timestamp explicit ellipsis

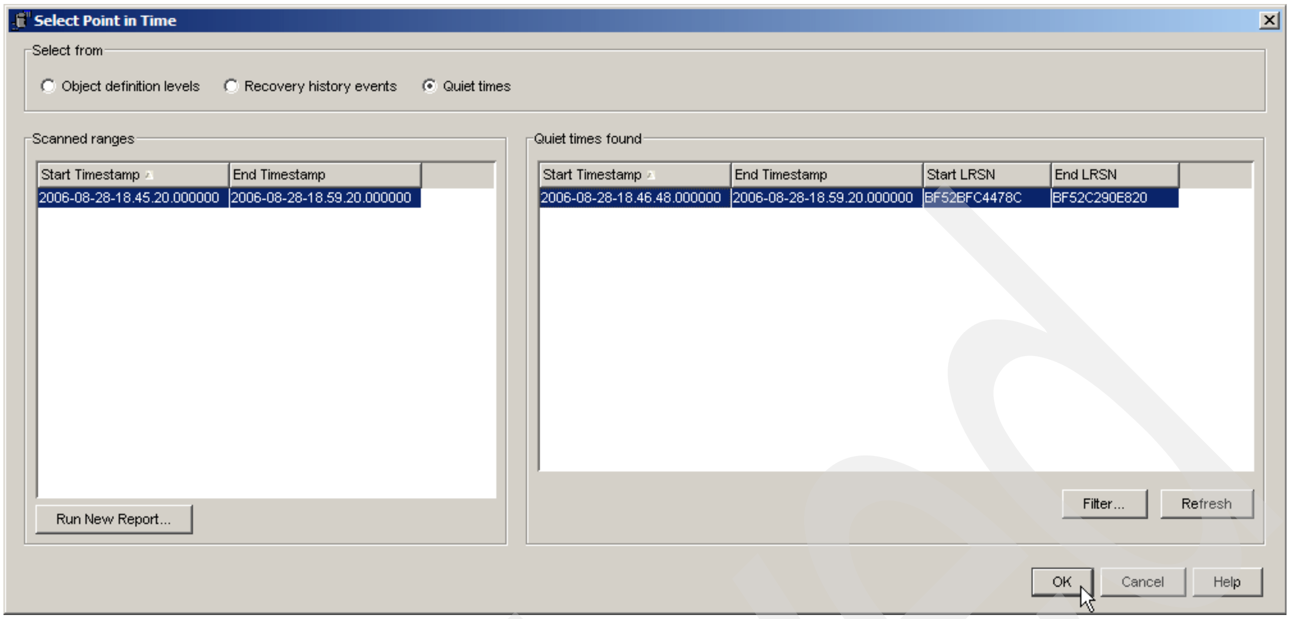

Figure 1-24   Selecting the quiet time

> **Note:** You have to click the **Refresh** button after running log analysis to repopulate this panel with newly discovered quiet times.

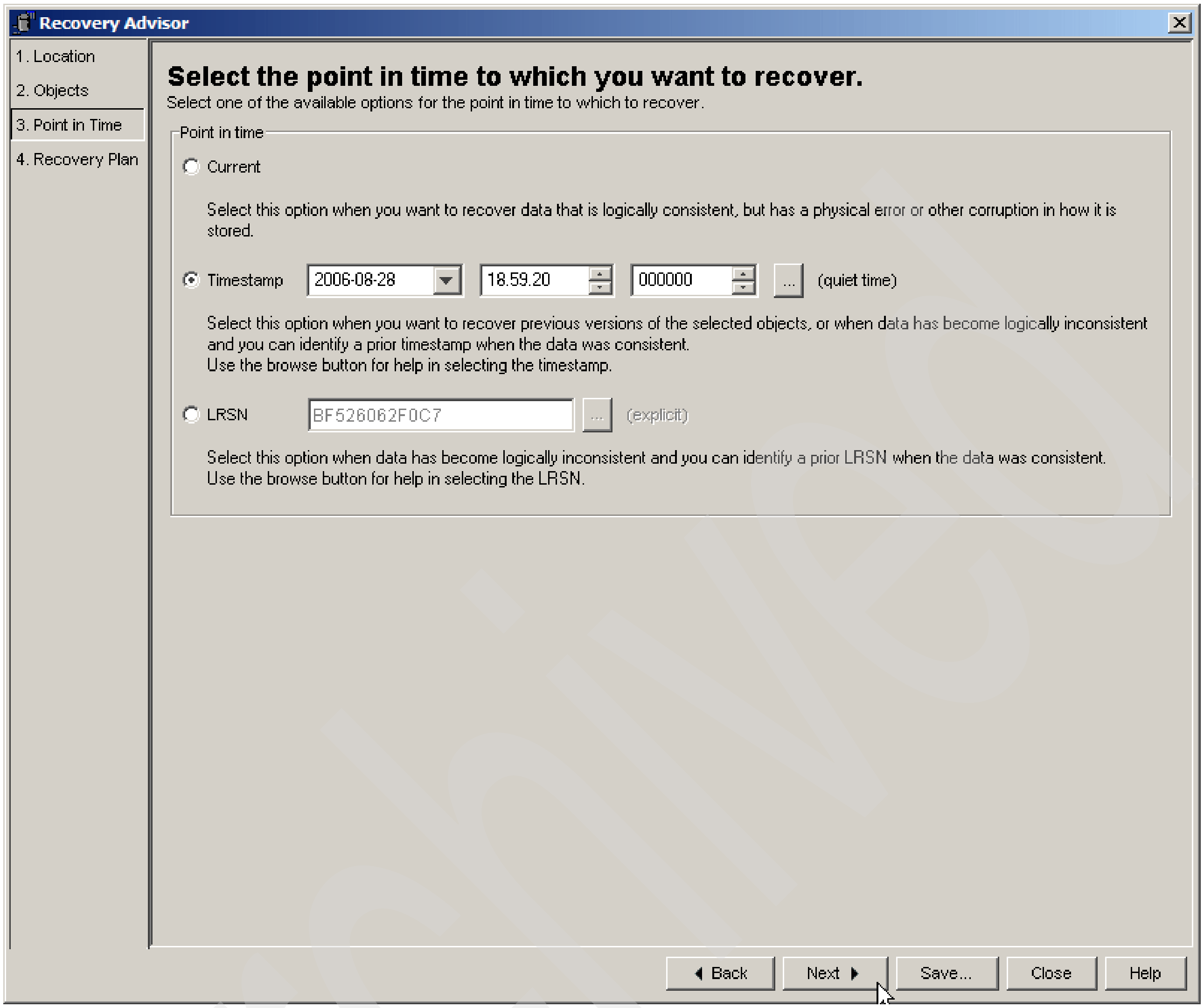

Figure 1-25 Selecting the timestamp corresponding to the quiet time

Specifications

Click **Specifications** (Figure 1-26 on page 42) to retrieve and use recovery specifications that are previously saved. After you select the location (Figure 1-27 on page 43) and click the **Refresh** button, you can see the available specifications (Figure 1-28 on page 43). You can highlight the specification of interest and open it, as shown in Figure 1-29 on page 44 and Figure 1-30 on page 45.

Other operations that are available are rename (Figure B-89 on page 299), copy (Figure B-90 on page 299), export (Figure B-91 on page 300), import (Figure B-92 on page 300), and delete (Figure B-93 on page 301).

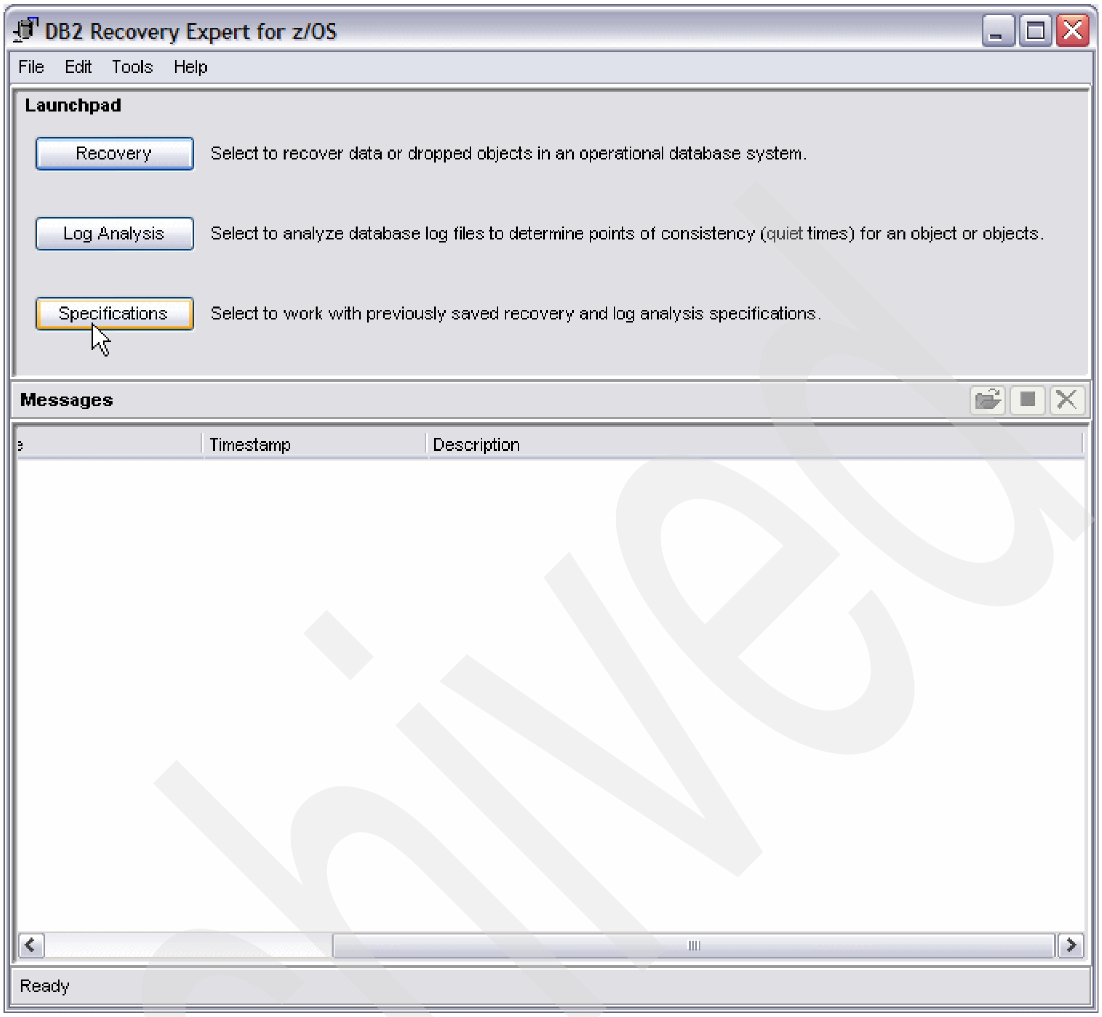

Figure 1-26 Launch specifications

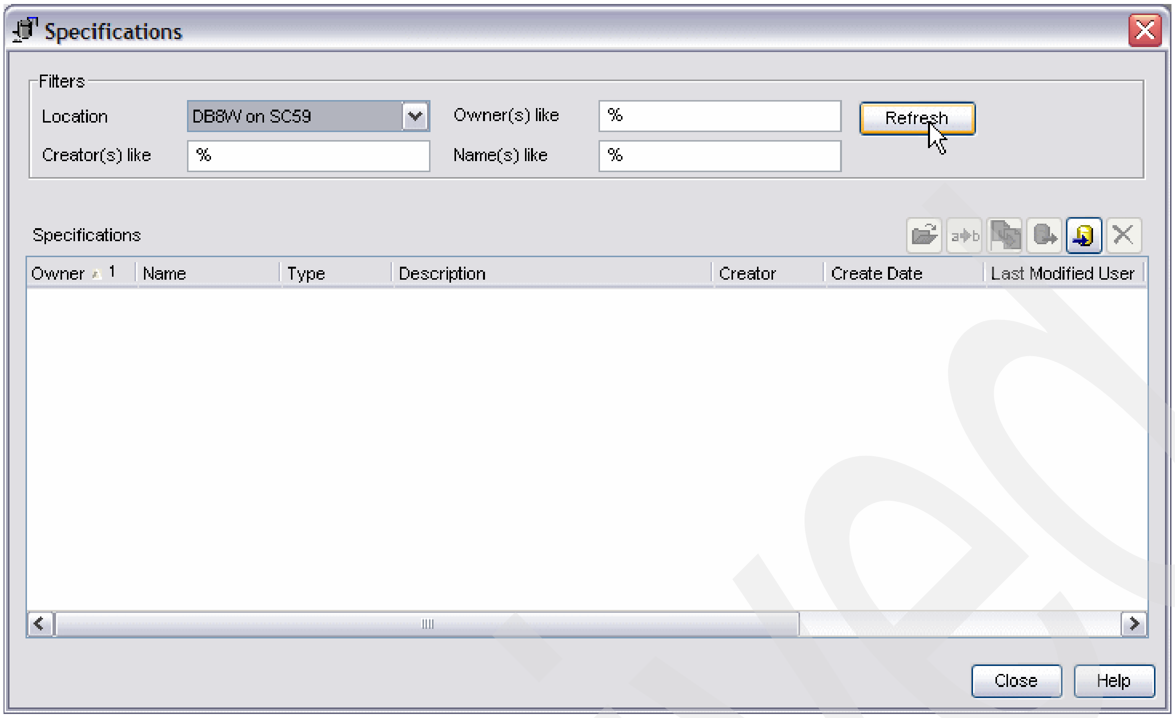

Figure 1-27 Filtering the specifications to view

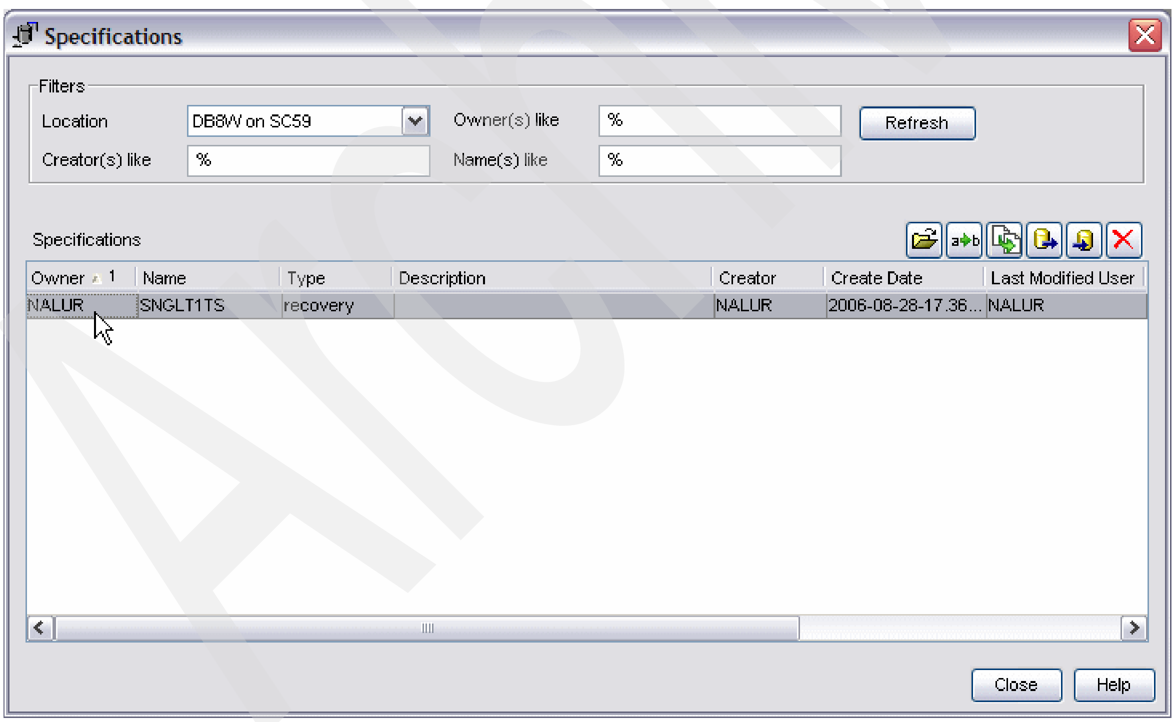

Figure 1-28 Available specifications

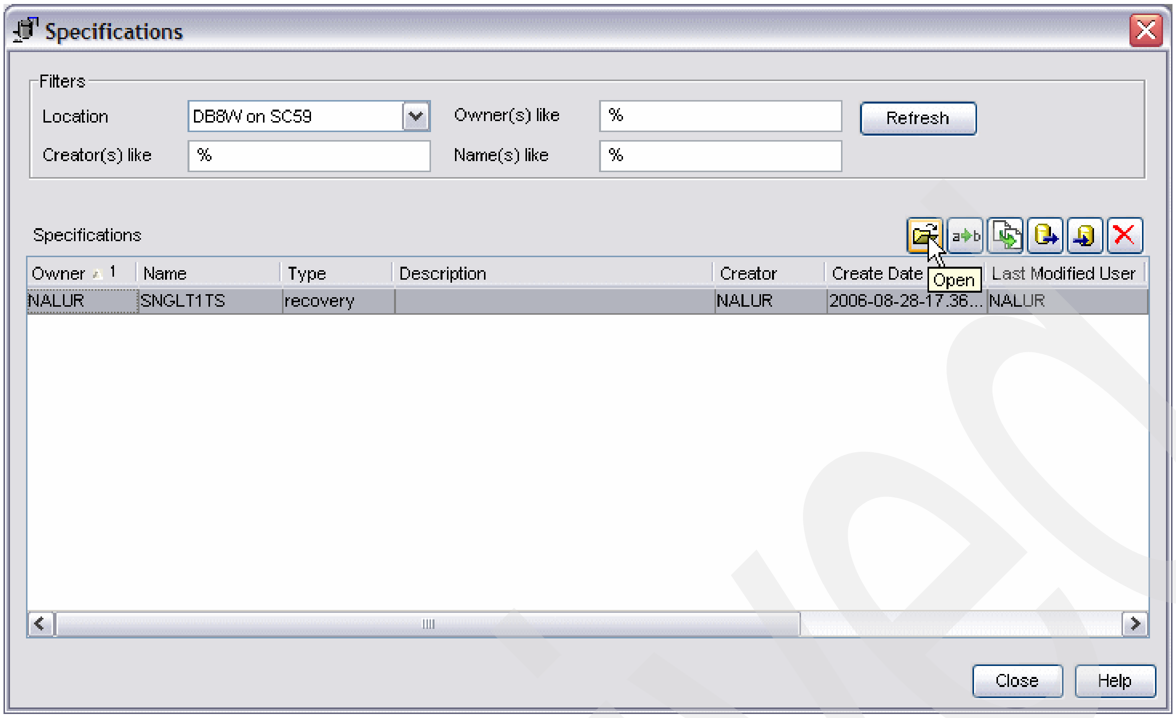

Figure 1-29 Opening the highlighted specification

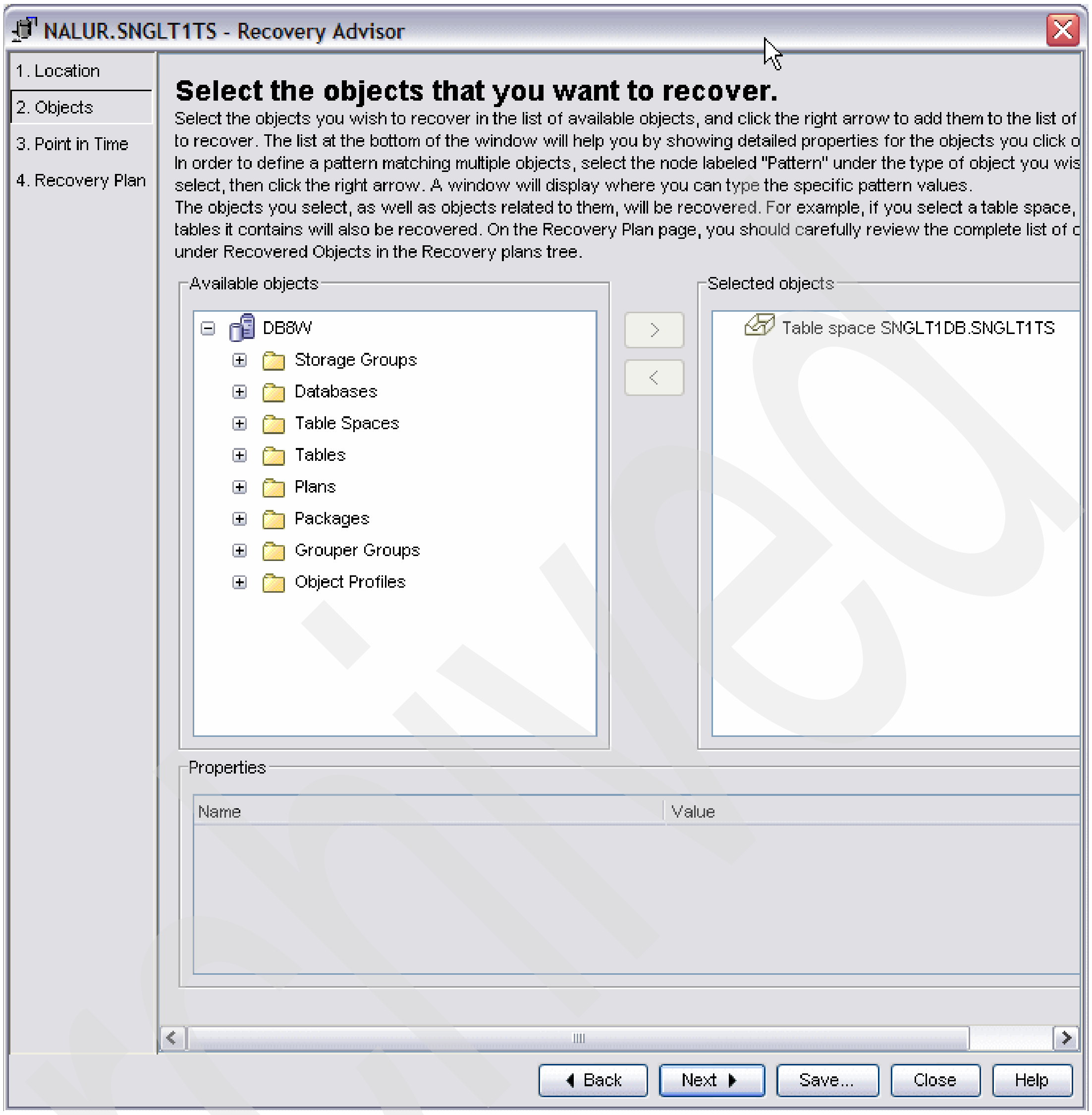

Figure 1-30 Retrieved specification

2

User scenarios

In this chapter, we provide a step-by-step approach on how to recover DB2 for z/OS objects to current or a point-in-time (PIT) using IBM DB2 Recovery Expert for z/OS in a data sharing and a non-data sharing environment.

The topics covered are:

- Environment configuration
- Recovering objects to current
- Recovering objects to a point-in-time
- Recovering dropped objects
- Recovering the entire subsystem

2.1 Introduction

The DB2 for z/OS environment can experience different kinds of failures such as Internal Resource Lock Manager (IRLM) failures, z/OS or power failures, disk failures, application errors, DB2 subsystem failures, site failures, and so on. When such failures occur, appropriate recovery procedures have to be executed swiftly and precisely to ensure minimal loss of data and minimal loss of system availability.

IBM Recovery Expert for z/OS supports recovery of the DB2 for z/OS environment after failures such as disk failures, application errors, and accidentally dropped objects. In this chapter, we provide a step-by-step approach to perform recovery of various DB2 objects using the IBM Recovery Expert for z/OS tool in our test environment, as described in the following sections:

- Environment configuration
- Recovering objects to current
- Recovering objects to a point-in-time
- Recovering dropped objects
- Recovering the entire subsystem

2.2 Environment configuration

Figure 2-1 shows the IBM DB2 Recovery Expert for z/OS (RE) configuration used in our user scenarios.

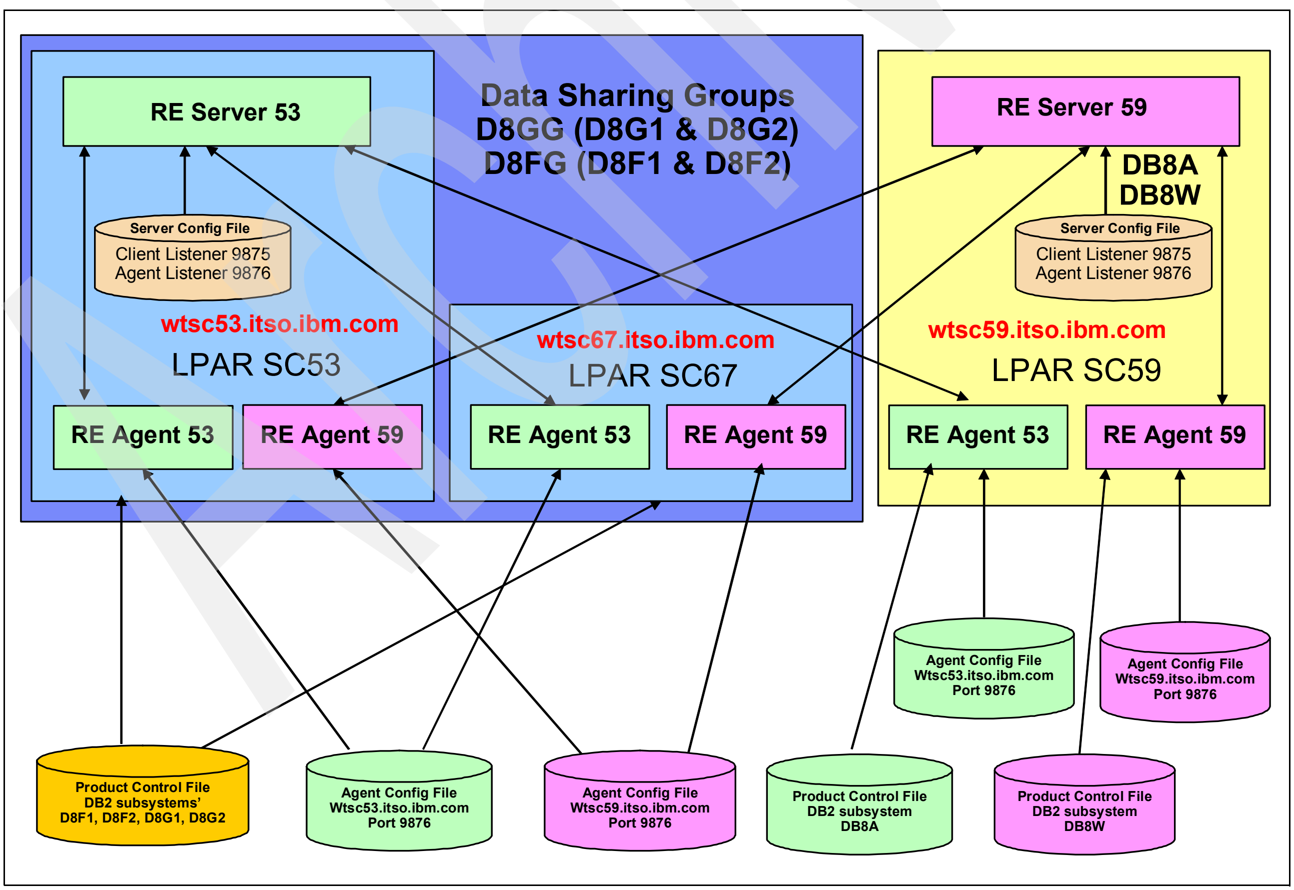

Figure 2-1 IBM DB2 Recovery Expert for z/OS configuration used in the user scenarios

Figure 2-1 shows the following components:

- Three logical partitions (LPARs) SC53, SC67, and SC59
- All the DB2 subsystems are at DB2 for z/OS Version 8
- LPARs SC53 and SC67 form a sysplex that has two data sharing groups D8GG and D8FG:
  - Data sharing group D8GG includes subsystems D8G1 and D8G2
  - Data sharing group D8FG includes subsystems D8F1 and D8F2
- LPAR SC59 is not part of any sysplex and has two DB2 subsystems DB8A and DB8W defined on it.
- There are two IBM DB2 Recovery Expert for z/OS servers installed in this configuration:
  - RE Server 53 on one member of the sysplex LPAR SC53
  - RE Server 59 on LPAR SC59 which is not part of any sysplex
- There are two agent instances on each of the three LPARs as follows:
  - RE Agent 53 on each of the LPARs reference Recovery Expert for z/OS Server 53 using an agent configuration file that identifies the host name (wtsc53.itso.ibm.com) and port number 9876 (default). The partial contents of this file are shown in Example 2-1, and the complete file is shown in Example E-3 on page 335.

Example 2-1 Partial contents of Recovery Expert for z/OS Agent 53 agent configuration file on all LPARs

```
<agent-config>
<server-address>wtsc53.itso.ibm.com</server-address>
<server-port>9876</server-port>
........................
</agent-config>
```

RE Agent 53 on LPARs SC53 and SC67 shares a common IBM Recovery Expert for z/OS product control file that identifies subsystems D8G1, D8G2, D8F1, and D8F2. The partial contents of this file are shown in Example 2-2, and the complete file is shown in Example E-4 on page 340.

Example 2-2 Partial contents of the shared Recovery Expert for z/OS Agent 53 product control file on LPARs SC53 and SC67

```
//ARYSJ002 JOB (999,POK),'CONWAY',CLASS=A,MSGCLASS=H,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
/*JOBPARM L=999,SYSAFF=SC53
................
................
 SET DB2 SSID          = D8G1
 UPDATE DB2 ZPARMS     = DSNZPAG1
 UPDATE DB2 BOOTSTRAP1 = DB8GU.D8G1.BSDS01
 UPDATE DB2 BOOTSTRAP2 = DB8GU.D8G1.BSDS02
 UPDATE DB2 LOADLIB1   = DB8G8.SDSNEXIT
 UPDATE DB2 LOADLIB2   = DB8G8.SDSNLOAD
.................
.................
 SET DB2 SSID          = D8G1
 SET PRODUCT CFG       = NULL
 SET PRODUCT VER       = NULL
*
 UPDATE ARY PLAN1      = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2      = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
```

```
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
...............
...............
 UPDATE ARY DSN PREFIX  = RETOOLS
................
................
 UPDATE LAS DSN PREFIX  = NALUR
................
................
 SET DB2 SSID           = D8G2
 UPDATE DB2 ZPARMS      = DSNZPAG2
 UPDATE DB2 BOOTSTRAP1  = DB8GU.D8G2.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8GU.D8G2.BSDS02
 UPDATE DB2 LOADLIB1    = DB8G8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8G8.SDSNLOAD
.................
.................
 SET DB2 SSID           = D8G2
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
.................
.................
 UPDATE ARY DSN PREFIX  = RETOOLS
.................
.................
 UPDATE LAS DSN PREFIX  = NALUR
.................
.................
 SET DB2 SSID           = D8F1
 UPDATE DB2 ZPARMS      = DSNZPAF1
 UPDATE DB2 BOOTSTRAP1  = DB8FU.D8F1.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8FU.D8F1.BSDS02
 UPDATE DB2 LOADLIB1    = DB8F8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8F8.SDSNLOAD
..................
..................
 SET DB2 SSID           = D8F1
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
...................
...................
```

```
 UPDATE ARY DSN PREFIX  = RETOOLS
...................
....................
 UPDATE LAS DSN PREFIX  = NALUR
....................
....................
 SET DB2 SSID           = D8F2
 UPDATE DB2 ZPARMS      = DSNZPAF2
 UPDATE DB2 BOOTSTRAP1  = DB8FU.D8F2.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8FU.D8F2.BSDS02
 UPDATE DB2 LOADLIB1    = DB8F8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8F8.SDSNLOAD
.....................
.....................
 SET DB2 SSID           = D8F2
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
......................
......................
 UPDATE ARY DSN PREFIX  = RETOOLS
......................
......................
 UPDATE LAS DSN PREFIX  = NALUR
...................... ........................
```

RE Agent 53 on LPAR SC59 has a separate IBM Recovery Expert for z/OS product control file that identifies subsystems DB8A. The partial contents of this file are shown in Example 2-3, and the complete file is shown in Example E-5 on page 356.

Example 2-3   Partial contents of Recovery Expert for z/OS Agent 53 product control file on LPAR SC59

```
//ARYSJA53 JOB (999,POK),'CONWAY',CLASS=A,MSGCLASS=H,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
......................
......................
 SET DB2 SSID           = DB8A
 UPDATE DB2 ZPARMS      = DSNZPARM
 UPDATE DB2 BOOTSTRAP1  = DB8AU.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8AU.BSDS02
 UPDATE DB2 LOADLIB1    = DB8A8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8A8.SDSNLOAD
......................
......................
 SET DB2 SSID           = DB8A
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
```

```
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
......................
......................
 UPDATE ARY DSN PREFIX  = NALUR
......................
.......................
 UPDATE LAS DSN PREFIX  = NALUR
.......................
.......................
```

– RE Agent 59 on each of the LPARs references Recovery Expert for z/OS Server 59 using an agent configuration file that identifies the host name (wtsc59.itso.ibm.com) and port number 9876 (default). The partial contents of this file are shown in Example 2-4, and the complete file is shown in Example E-6 on page 361.

Example 2-4 Partial contents of Recovery Expert for z/OS Agent 59 agent configuration file on all LPARs

```
<agent-config>
<server-address>wtsc59.itso.ibm.com</server-address>
<server-port>9876</server-port>
........................
</agent-config>
```

RE Agent 59 on LPARs SC53 and SC67 shares the *same* IBM Recovery Expert for z/OS product control file used by Recovery Expert for z/OS Agent 53. The partial contents of this file are shown in Example 2-5, and the complete file is shown in Example E-7 on page 365.

Example 2-5 Partial contents of the shared Recovery Expert for z/OS Agent 59 product control file on LPARs SC53 and SC67

```
//ARYSJ002 JOB (999,POK),'CONWAY',CLASS=A,MSGCLASS=H,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
/*JOBPARM L=999,SYSAFF=SC53
....................
....................
 SET DB2 SSID           = D8G1
 UPDATE DB2 ZPARMS      = DSNZPAG1
 UPDATE DB2 BOOTSTRAP1  = DB8GU.D8G1.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8GU.D8G1.BSDS02
 UPDATE DB2 LOADLIB1    = DB8G8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8G8.SDSNLOAD
.....................
.....................
 SET DB2 SSID           = D8G1
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
.....................
.....................
```

```
 UPDATE ARY DSN PREFIX  = RETOOLS
.....................
.....................
 UPDATE LAS DSN PREFIX  = NALUR
.....................
.....................
 SET DB2 SSID           = D8G2
 UPDATE DB2 ZPARMS      = DSNZPAG2
 UPDATE DB2 BOOTSTRAP1  = DB8GU.D8G2.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8GU.D8G2.BSDS02
 UPDATE DB2 LOADLIB1    = DB8G8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8G8.SDSNLOAD
......................
......................
 SET DB2 SSID           = D8G2
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
......................
......................
 UPDATE ARY DSN PREFIX  = RETOOLS
......................
......................
 UPDATE LAS DSN PREFIX  = NALUR
......................
......................
 SET DB2 SSID           = D8F1
 UPDATE DB2 ZPARMS      = DSNZPAF1
 UPDATE DB2 BOOTSTRAP1  = DB8FU.D8F1.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8FU.D8F1.BSDS02
 UPDATE DB2 LOADLIB1    = DB8F8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8F8.SDSNLOAD
.......................
.......................
 SET DB2 SSID           = D8F1
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
.......................
.......................
 UPDATE ARY DSN PREFIX  = RETOOLS
.......................
.......................
```

```
 UPDATE LAS DSN PREFIX  = NALUR
.......................
.......................
 SET DB2 SSID           = D8F2
 UPDATE DB2 ZPARMS      = DSNZPAF2
 UPDATE DB2 BOOTSTRAP1  = DB8FU.D8F2.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8FU.D8F2.BSDS02
 UPDATE DB2 LOADLIB1    = DB8F8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8F8.SDSNLOAD
.....................
.....................
 SET DB2 SSID           = D8F2
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
.......................
.......................
 UPDATE ARY DSN PREFIX  = RETOOLS
.......................
.......................
 UPDATE LAS DSN PREFIX  = NALUR
.......................
.......................
```

RE Agent 59 on LPAR SC59 has a separate IBM Recovery Expert for z/OS product control file that only identifies subsystem DB8W. The partial contents of this file are shown in Example 2-6, and the complete file is shown in Example E-8 on page 382.

Example 2-6 Partial contents of Recovery Expert for z/OS Agent 59 product control file on LPAR SC59

```
//ARYSJA59 JOB (999,POK),'CONWAY',CLASS=A,MSGCLASS=H,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
...................
...................
 SET DB2 SSID           = DB8W
 UPDATE DB2 ZPARMS      = DSNZPARM
 UPDATE DB2 BOOTSTRAP1  = DB8WU.DB2.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8WU.DB2.BSDS02
 UPDATE DB2 LOADLIB1    = DB8W8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8W8.SDSNLOAD
....................
....................
 SET DB2 SSID           = DB8W
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
```

```
.....................
.....................
 UPDATE ARY DSN PREFIX  = NALUR
.....................
.....................
 UPDATE LAS DSN PREFIX  = NALUR
......................
```

This configuration supports the following administration environment:

– Database administrators (DBAs) logging on to the Recovery Expert for z/OS Server 53 can manage DB2 subsystems D8F1, D8F2, D8G1, D8G2 and DB8A.

– DBAs logging on to Recovery Expert for z/OS Server 59 can manage DB2 subsystems D8F1, D8F2, D8G1, D8G2 and DB8W.

These configurations demonstrate the following tasks:

– How to configure IBM DB2 Recovery Expert for z/OS in a data sharing environment and a non-data sharing environment

– How to have different DB2 subsystems (DB8A and DB8W) on the *same* LPAR managed by two different IBM DB2 Recovery Expert for z/OS servers: Recovery Expert for z/OS Server 53 (which manages DB8A) and Recovery Expert for z/OS Server 59 (which manages DB8W)

  This effect is achieved by having the Recovery Expert for z/OS Agent 53 and Recovery Expert for z/OS Agent 59 on LPAR SC59 use different product control files. This simulates an environment where a more experienced DBA is responsible for managing a critical DB2 subsystem, and a lesser experienced DBA manages less critical DB2 subsystem on the same LPAR.

– How to have different DB2 subsystems (D8G1, D8G2, D8F1 and D8F2) on the *same* sysplex managed by two different IBM DB2 Recovery Expert for z/OS servers Recovery Expert for z/OS Server 53 and Recovery Expert for z/OS Server 59

  This effect is achieved by having the agents Recovery Expert for z/OS Agent 53 and Recovery Expert for z/OS Agent 59 on LPARs SC53 and SC67 use the *same* product control files.

  Note: With such a setup, to avoid conflicts and confusion, you have to ensure that different DBAs do not try to simultaneously manage the same DB2 subsystems from different IBM DB2 Recovery Expert for z/OS servers.

Attention: From the perspective of managing the recovery of DB2 objects in a data sharing or non-data sharing environment, the location of the IBM Recovery Expert for z/OS server (whether it is on a member of a sysplex, or on an LPAR that is not part of a sysplex) does not alter the approach for recovering DB2 objects using the IBM Recovery Expert for z/OS tool.

In the following user scenarios, the choice of a particular IBM Recovery Expert for z/OS server (RE Server 53 or Recovery Expert for z/OS Server 59) as the management center for a given scenario has no particular significance other than that it was convenient to use at that point.

Note: Recovery Expert for z/OS supports the recovery of partitioned and non-partitioned table spaces, table space partitions, tables, and aggregate objects such as databases, storage groups, plans, packages, DB2 Grouper groups, and DB2 Automation Tool object profiles. It currently does not support indexes as objects of recovery in the Recovery Expert for z/OS GUI client.

2.3 Recovering objects to current

The triggering event for recovering an object or set of objects to current is typically media failure, such as an input/output (I/O) error or volume failure. Typically, when a single object such as a table space is recovered to current, its dependent objects such as indexes or other table spaces in the same referential set are not recovered because they are considered to be unaffected by the error.

Attention: Recovery Expert for z/OS assumes that all dependent objects such as other table spaces in the referential set, and all indexes[a] have to be recovered as well, and therefore generates appropriate control statements to do so. It also automatically generates CHECK DATA control statements for table spaces that are part of a referential set. If you think that this additional level of protection is not required in your organization, you can modify the generated recovery plans at your own risk.

a. Indexes are rebuilt using the REBUILD INDEX statement.

As far as possible, we have used a common timeline showing a sequence of events in all the scenarios. These events are not the most optimal of operations, and have been selected to demonstrate the recovery semantics of Recovery Expert for z/OS.

The first scenario describes the complete sequence of operations required to recover an object to current. The subsequent scenarios only detail some of the key screen captures of interest to that particular scenario. The recover to current scenarios described in this section are:

- Recovering multi-table table space
- Recovering multiple table spaces (volume list)
- Recovering one or more partitions of a partitioned table space

2.3.1 Recovering multi-table table space

This scenario describes the recovery of a multi-table table space MTBLTSTS in database MTBLTSDB. This environment is a non-data sharing environment. The following section describes the sequence of operations to recover a multi-table table space MTBLTSTS to current using the Recovery Expert for z/OS GUI client.

To launch the Recovery Expert for z/OS GUI client from your workstation, select **Start** → **IBM DB2 Recovery Expert** → **IBM DB2 Recovery Expert**, which opens the window shown in Figure 2-2.

Figure 2-2 Selecting the DB2 Recovery Expert server

Type or select the DB2 Recovery Expert server and click **Connect** to proceed to the Launchpad window shown in Figure 2-3.

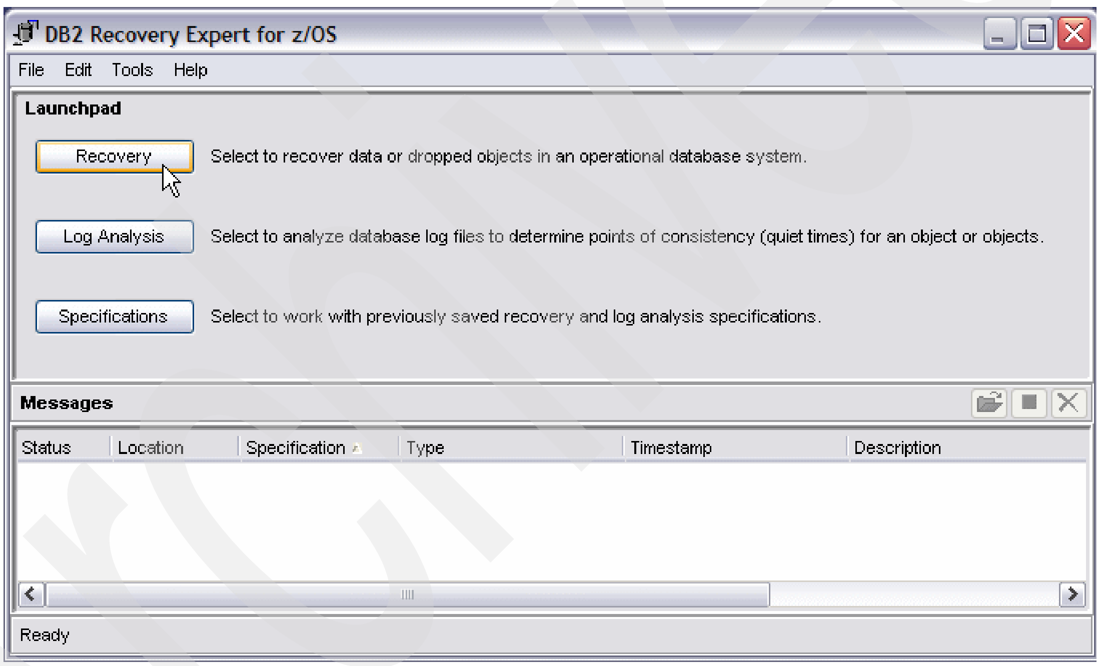

Figure 2-3 Launchpad

Click **Recovery** in Figure 2-3 to proceed to the Recovery Advisor welcome window shown in Figure 2-4.

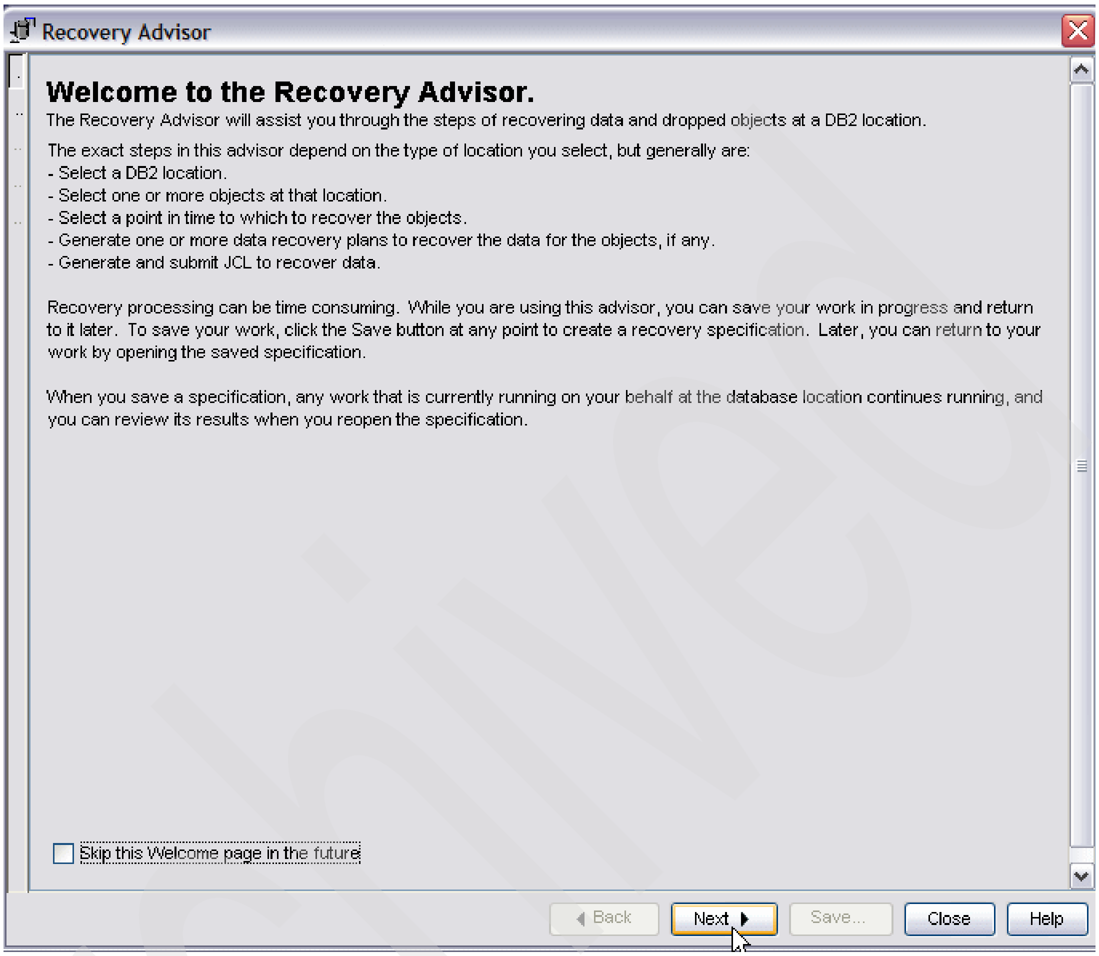

Figure 2-4 Recovery Advisor welcome window

Review the contents shown in Figure 2-4, select the check box to skip this welcome window, if required. Click **Next** to proceed to the selection of the location shown Figure 2-5.

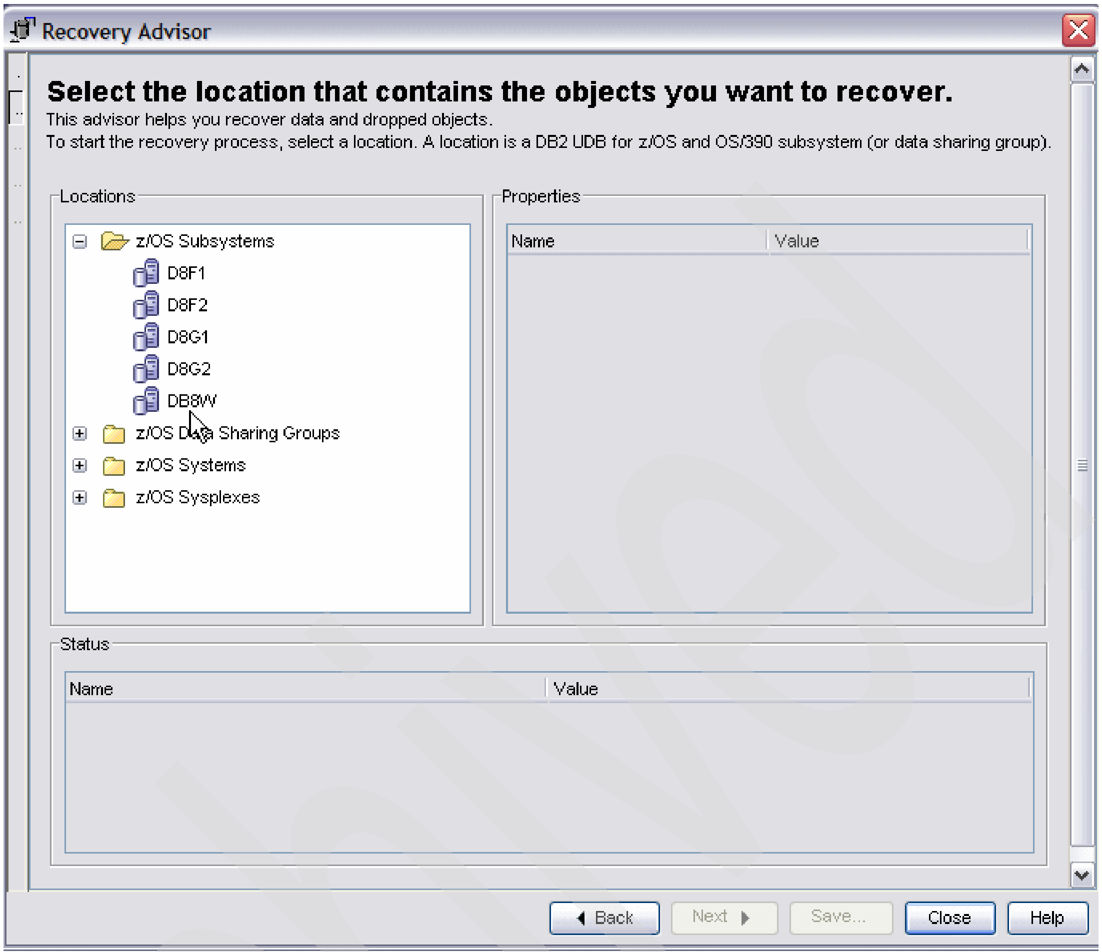

Figure 2-5 Selecting the location DB8W

Expand the **z/OS Subsystems** navigation pane and select **DB8W** in Figure 2-5. This prompts you to provide authentication information, as shown in Figure 2-6.

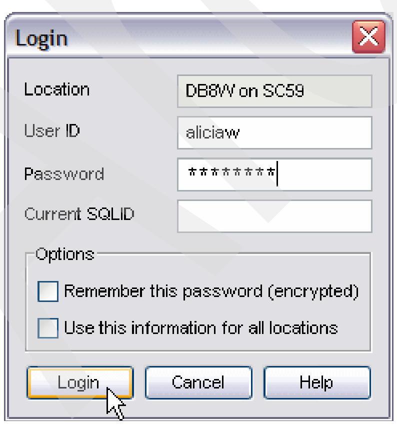

Figure 2-6 Login information for the location

Provide the User ID and password information. Select the appropriate check boxes in the Options section of Figure 2-6, and click **Login**. This opens Figure 2-7.

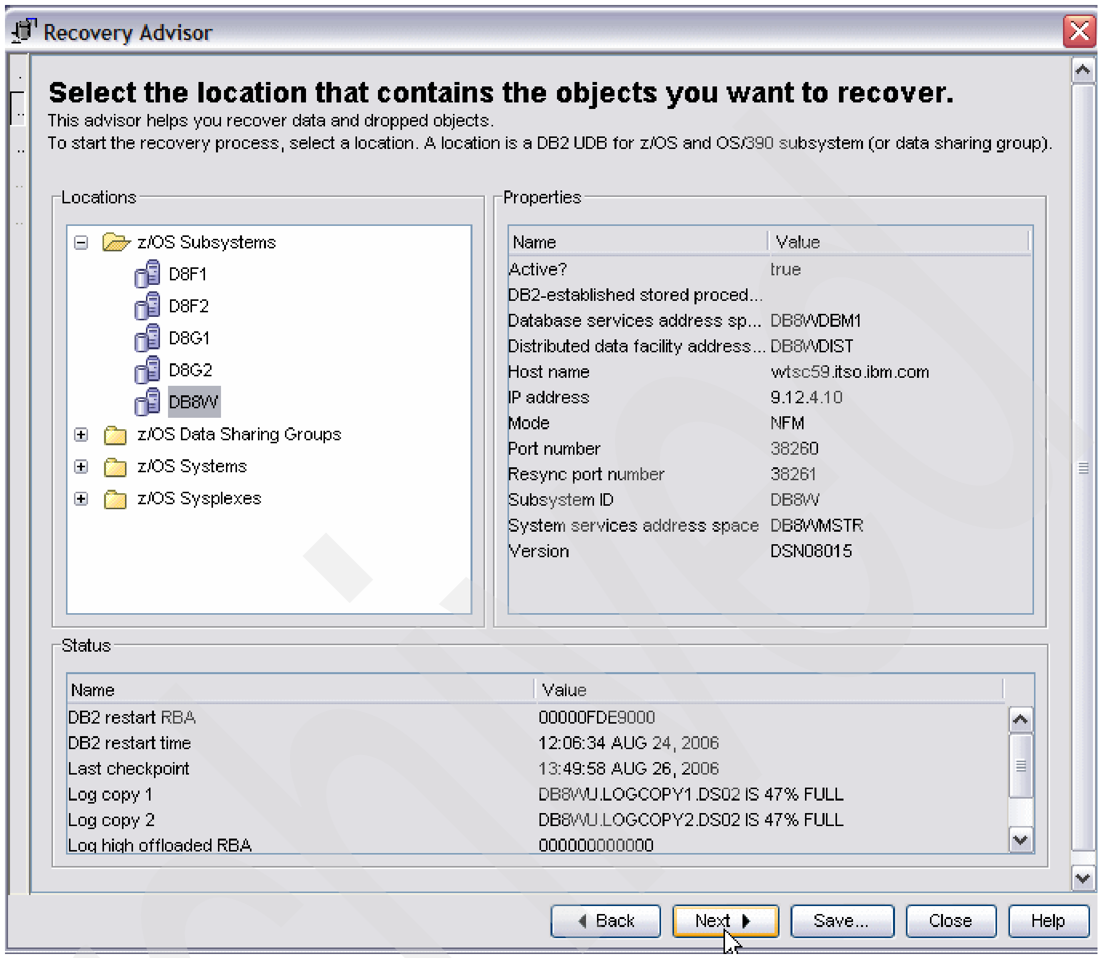

Figure 2-7 Selected location DB8W details

Figure 2-7 displays details of the selected location as shown in the Status and Properties sections. Click **Next** to select the objects to recover, as shown in Figure 2-8.

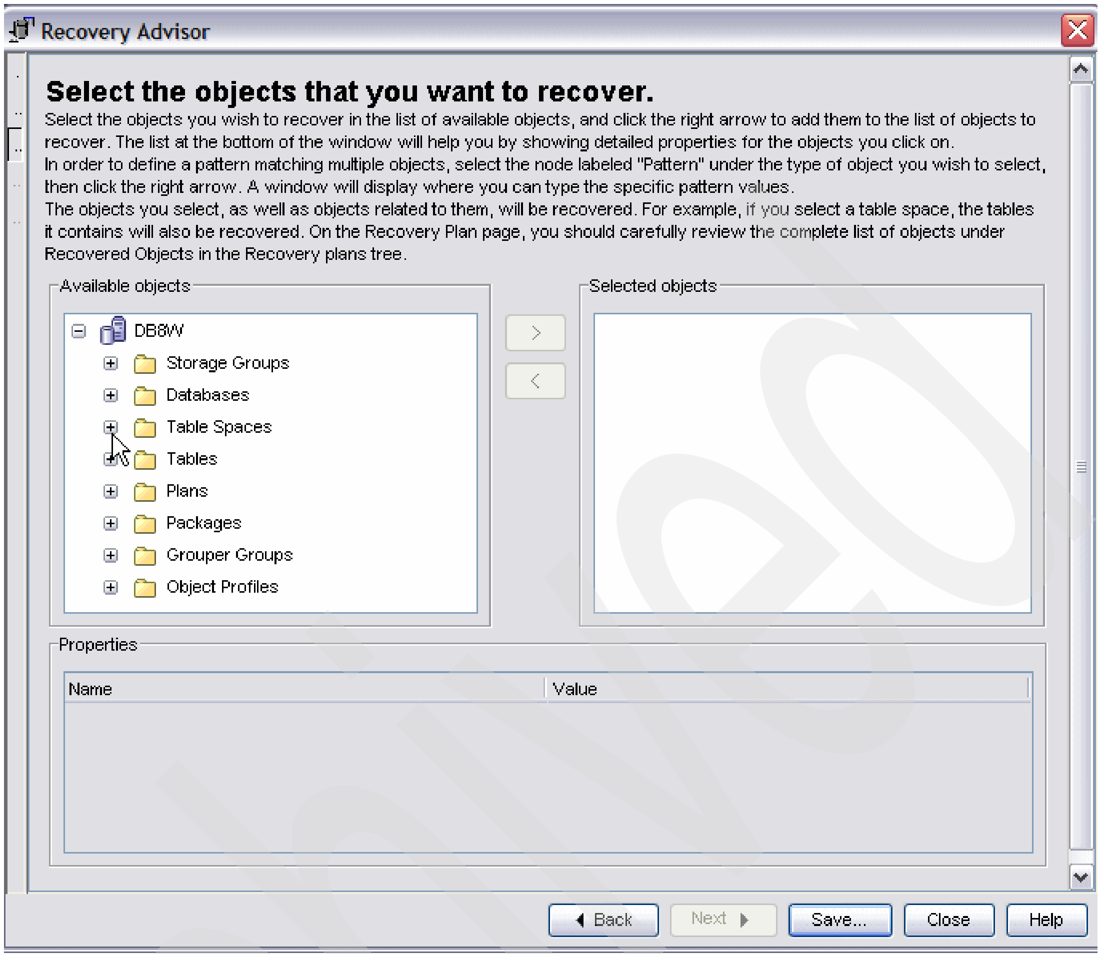

Figure 2-8 Selecting the objects to recover (1/9)

Figure 2-8 through Figure 2-16 on page 67 show the steps involved in selecting the objects to recover. We choose the Pattern option in the navigation pane to demonstrate a filtering mechanism that is available in Recovery Expert for z/OS. Use the Help button to seek clarification of the available options, for example, uppercase is required when providing filtering criteria.

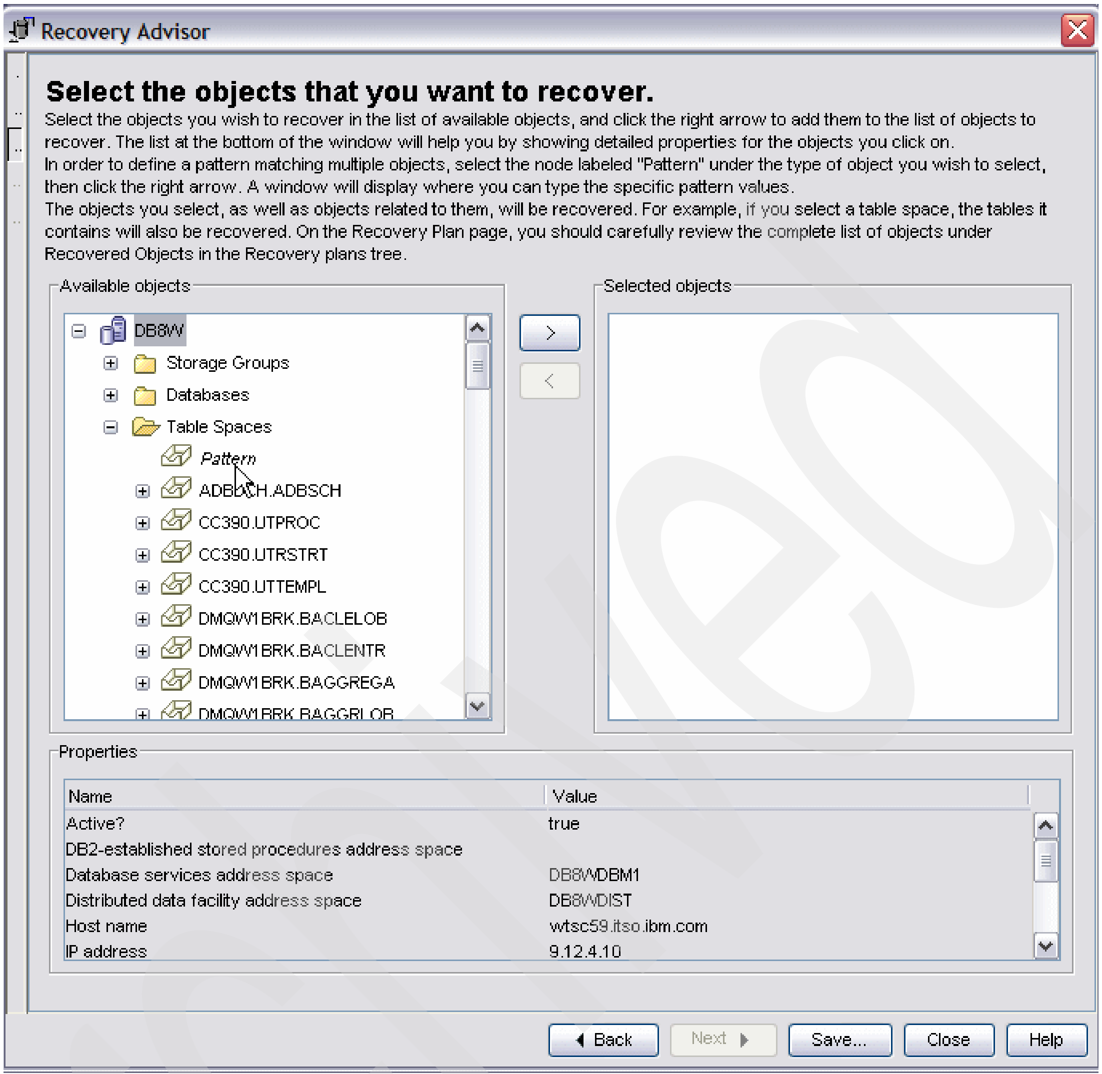

Figure 2-9 Selecting the objects to recover (2/9)

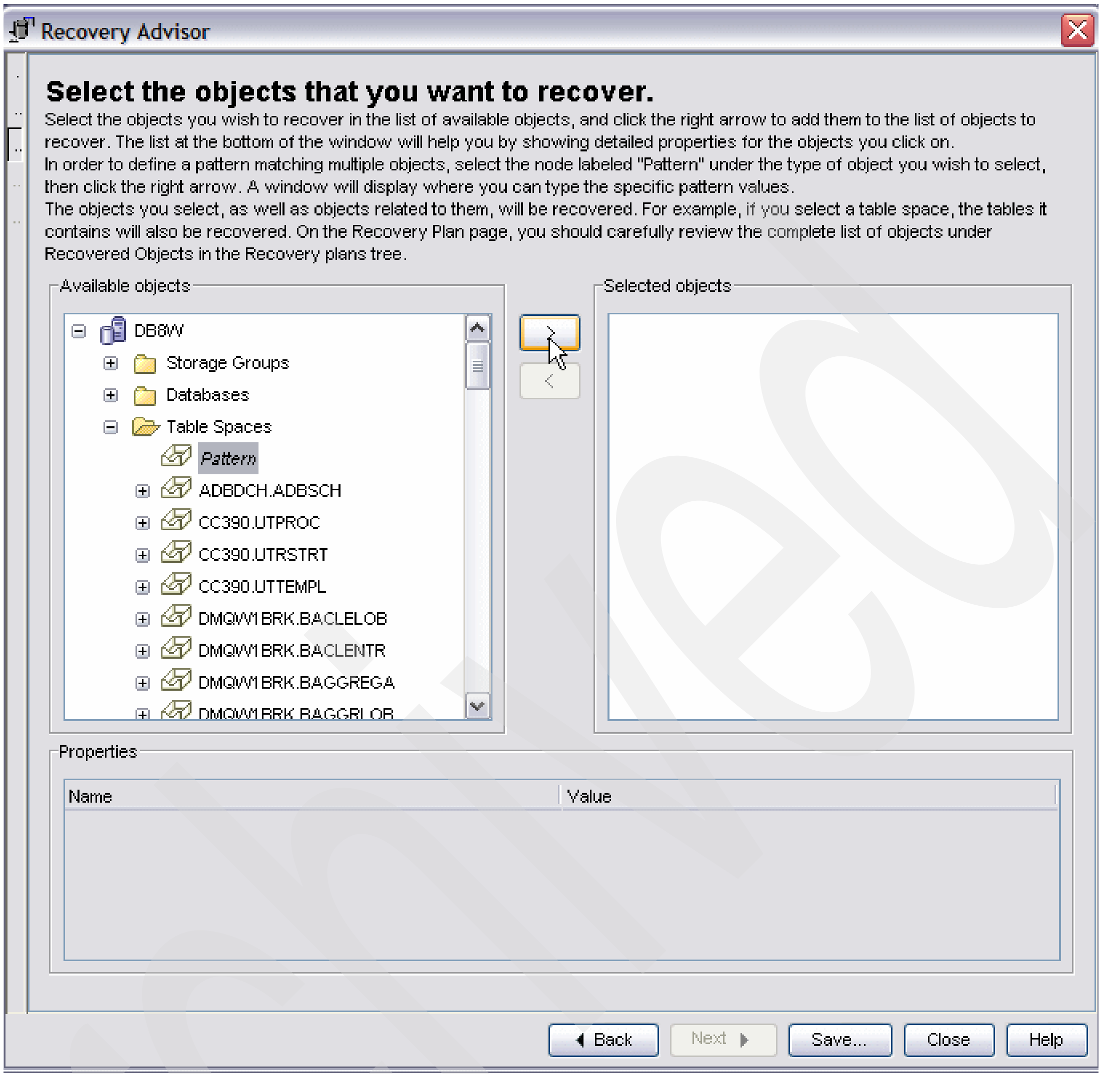

Figure 2-10   Selecting the objects to recover (3/9)

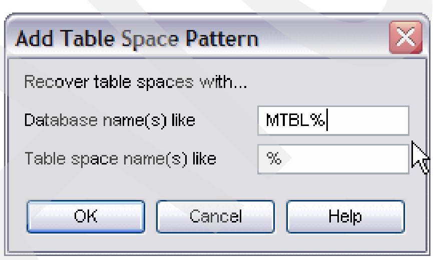

Figure 2-11   Selecting the objects to recover (4/9)

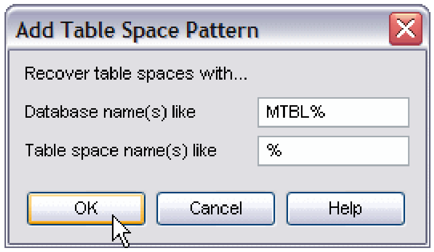

Figure 2-12 Selecting the objects to recover (5/9)

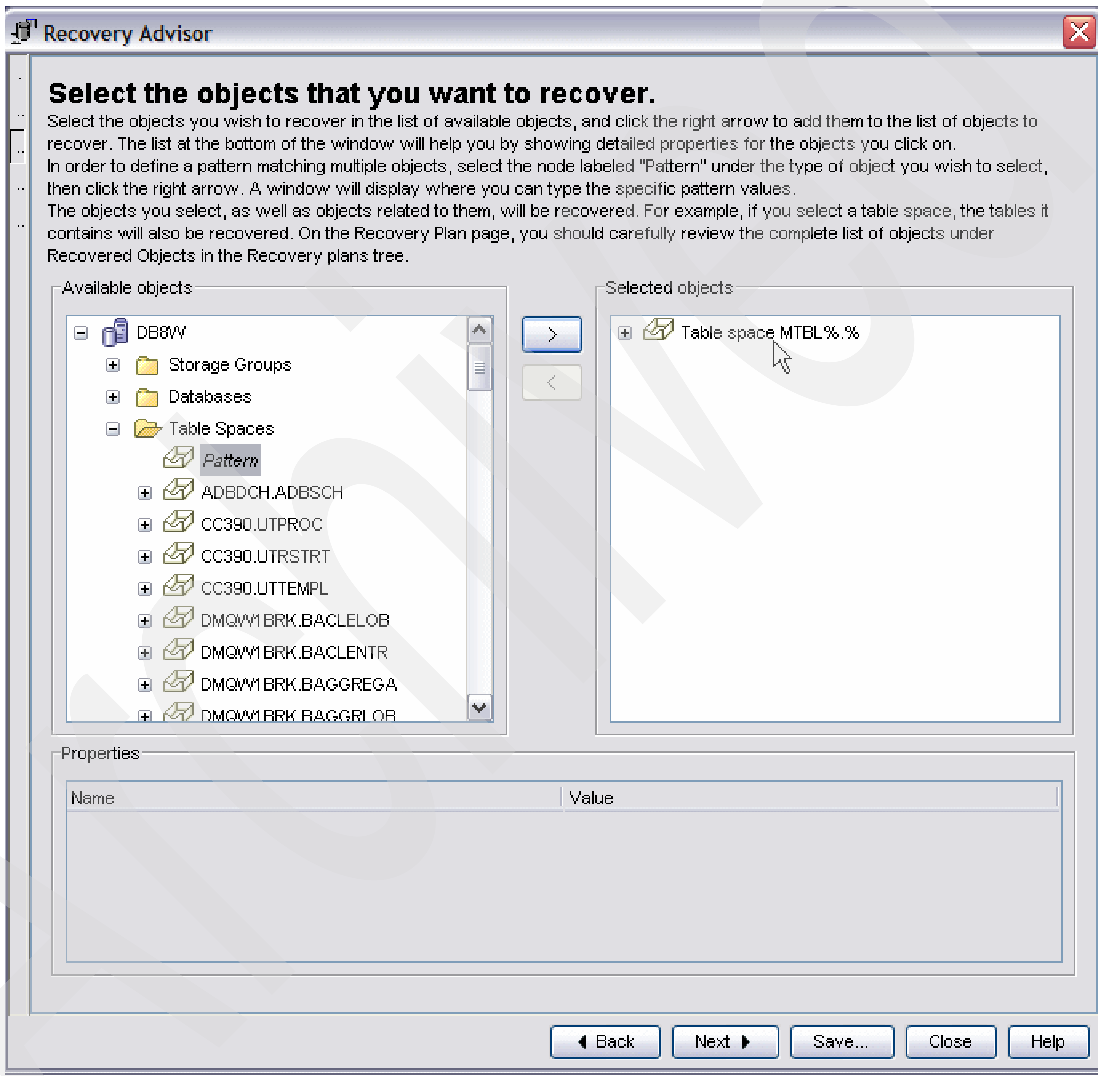

Figure 2-13 Selecting the objects to recover (6/9)

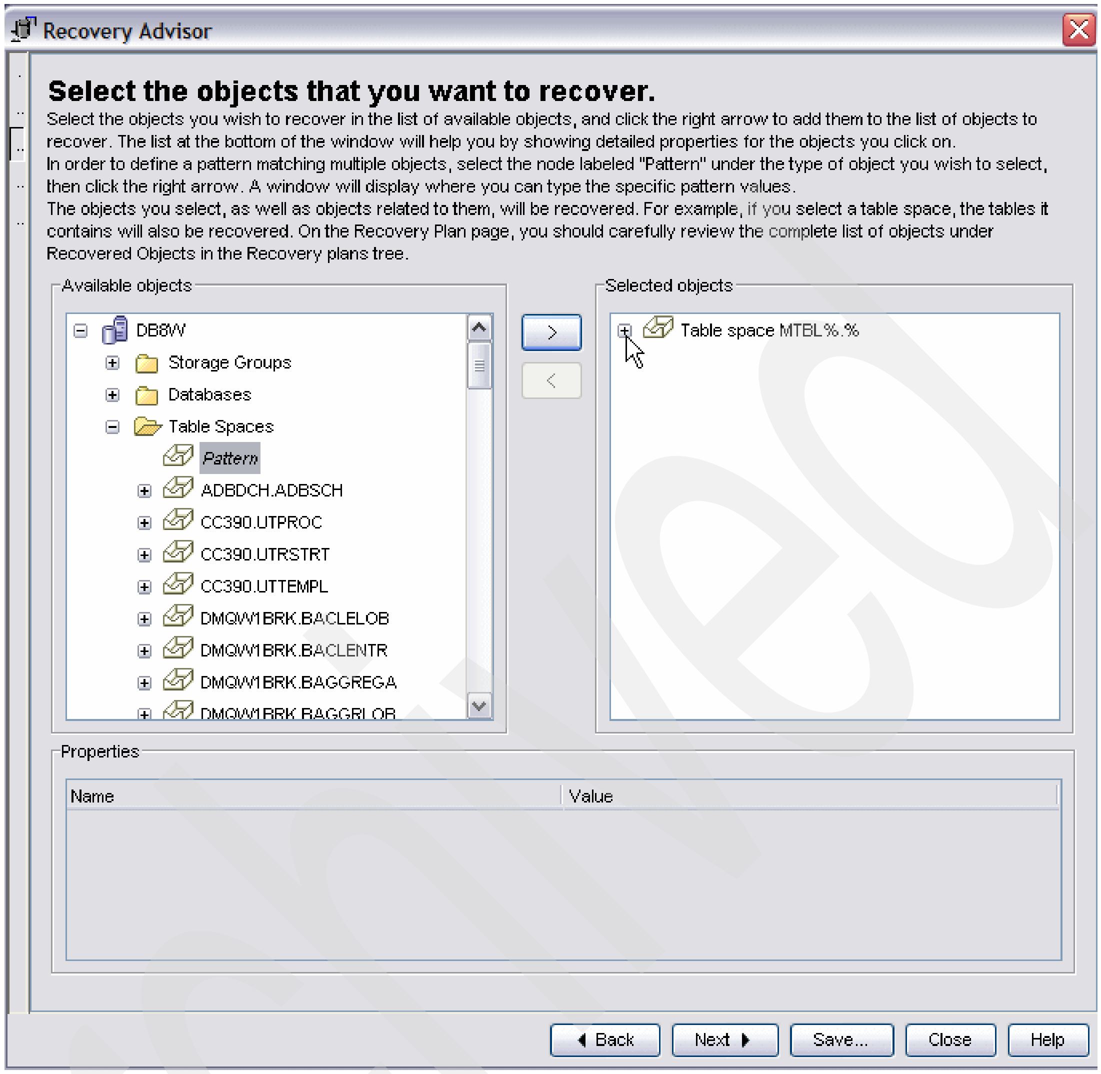

Figure 2-14 Selecting the objects to recover (7/9)

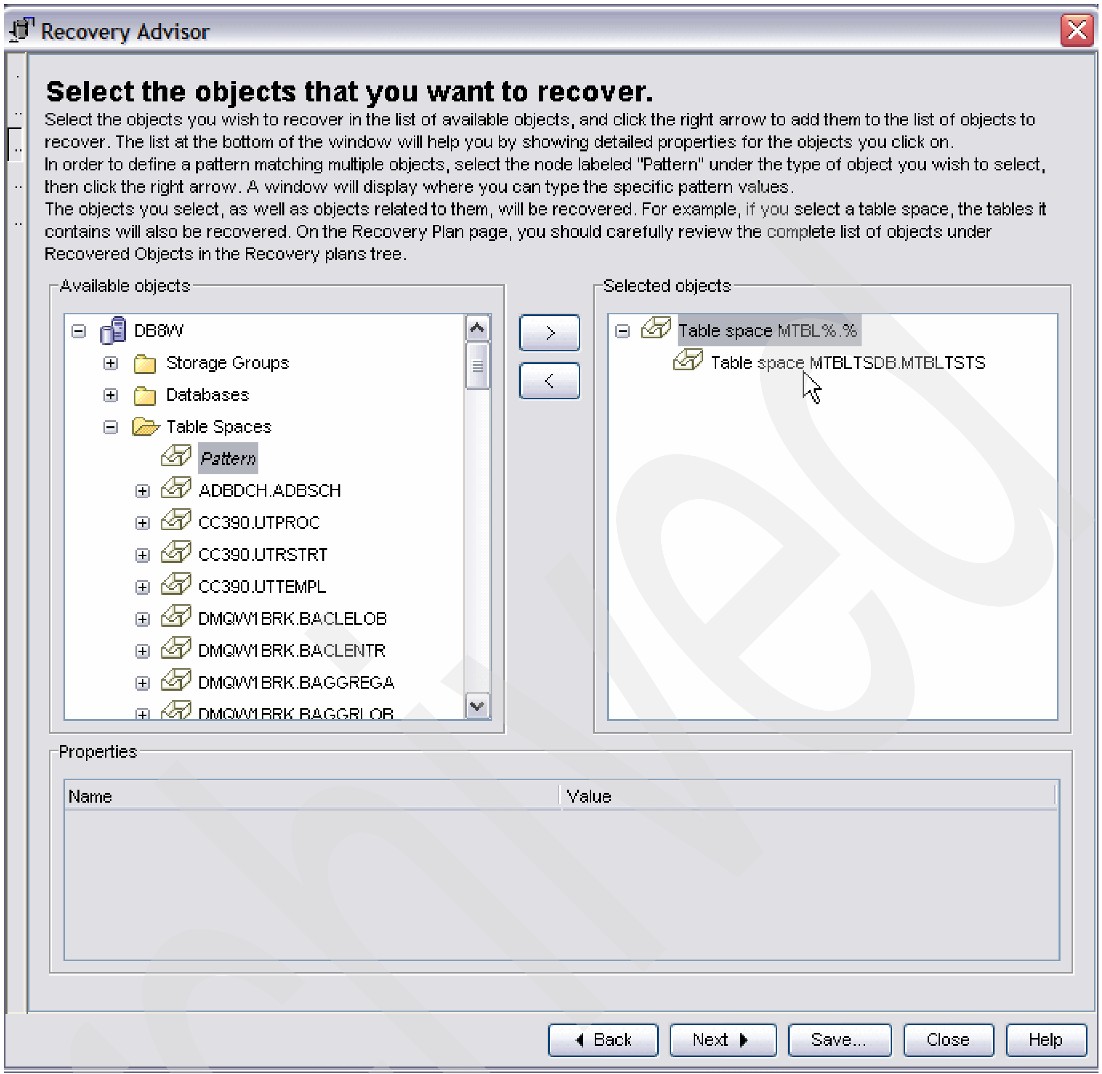

Figure 2-15 Selecting the objects to recover (8/9)

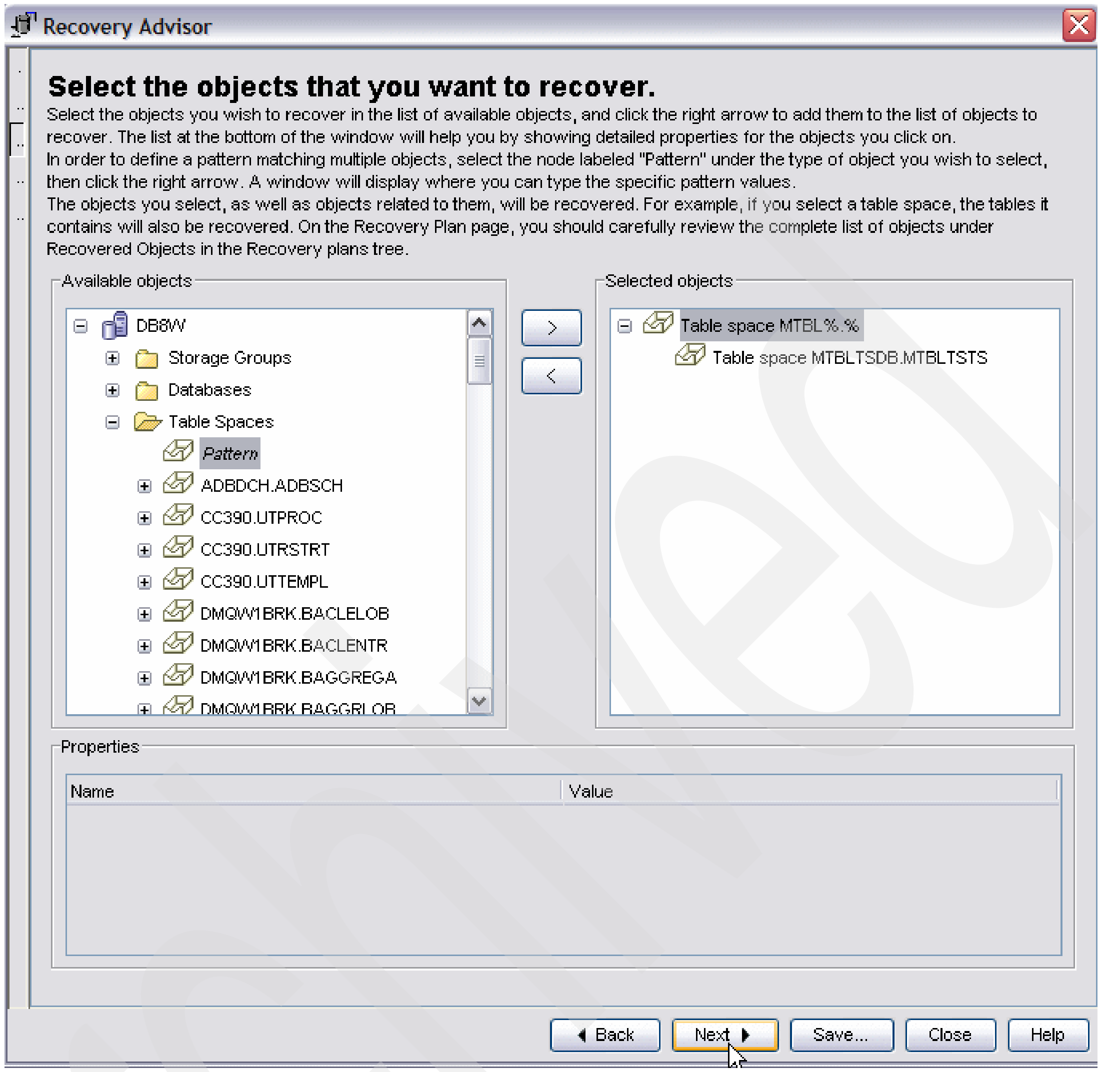

Figure 2-16 Selecting the objects to recover (9/9)

After you select the object to recover, to select the point to which it is to be recovered, click **Next** in Figure 2-16. This opens the window shown in Figure 2-17.

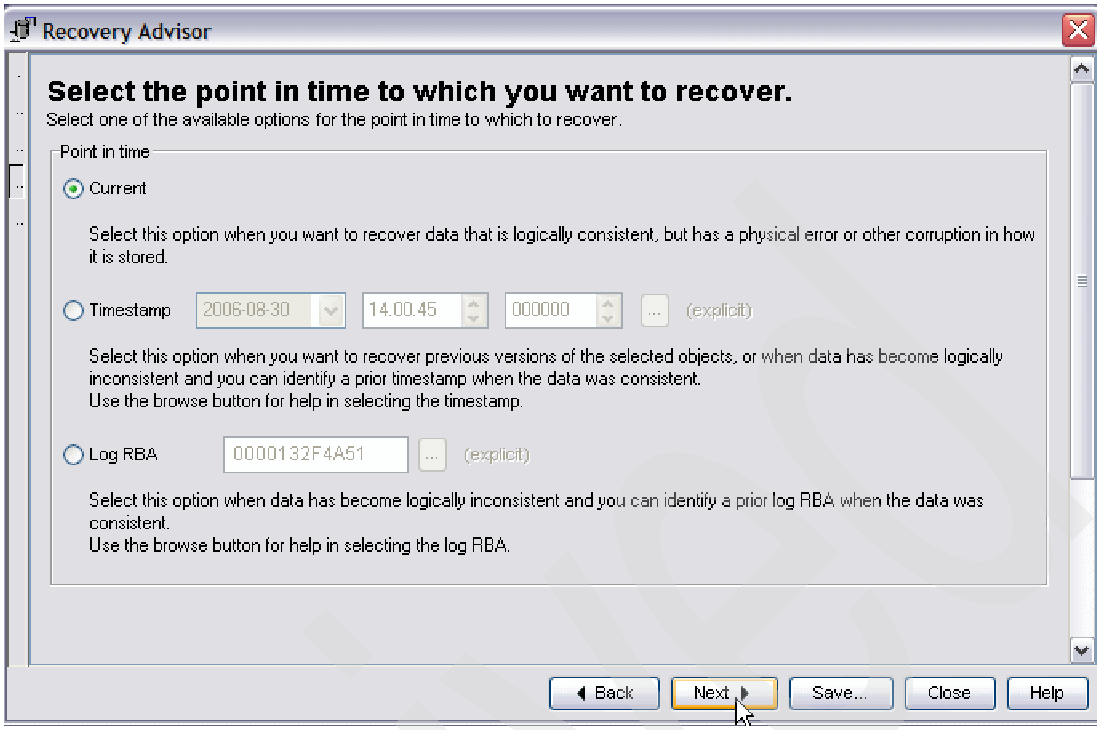

Figure 2-17 Selecting the point to recover to

Select **Current** in Figure 2-17 and click **Next** to generate potential recovery plans. This opens the window shown in Figure 2-18.

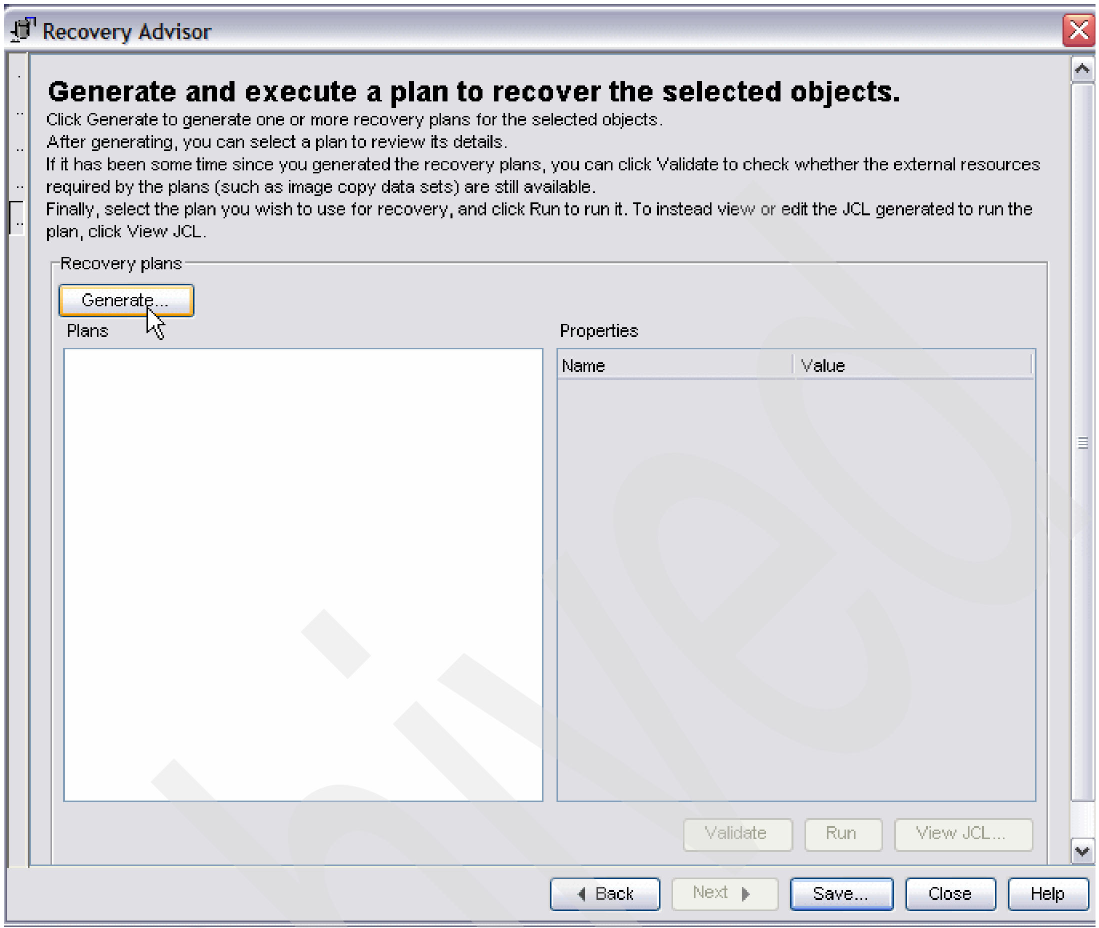

Figure 2-18 Generating recovery plans

Click **Generate** in the Recovery plans section, as shown in Figure 2-18. This opens a series of windows that provide you with options that influence the recovery plans generated, as shown in Figure 2-19 through Figure 2-22 on page 70. As advised earlier, use the Help button for considerations in choosing each option. After you choose all of the options, proceed to Figure 2-23 on page 71.

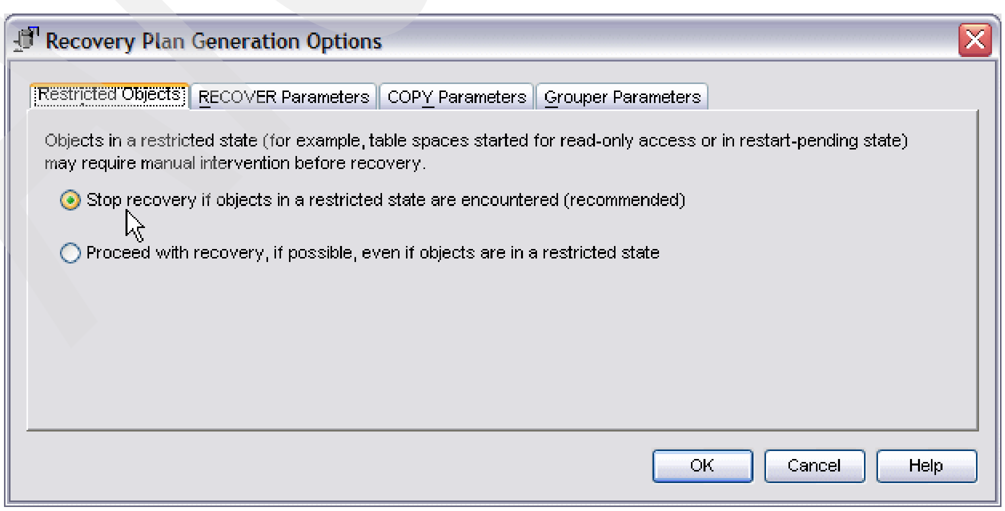

Figure 2-19 Restricted objects options

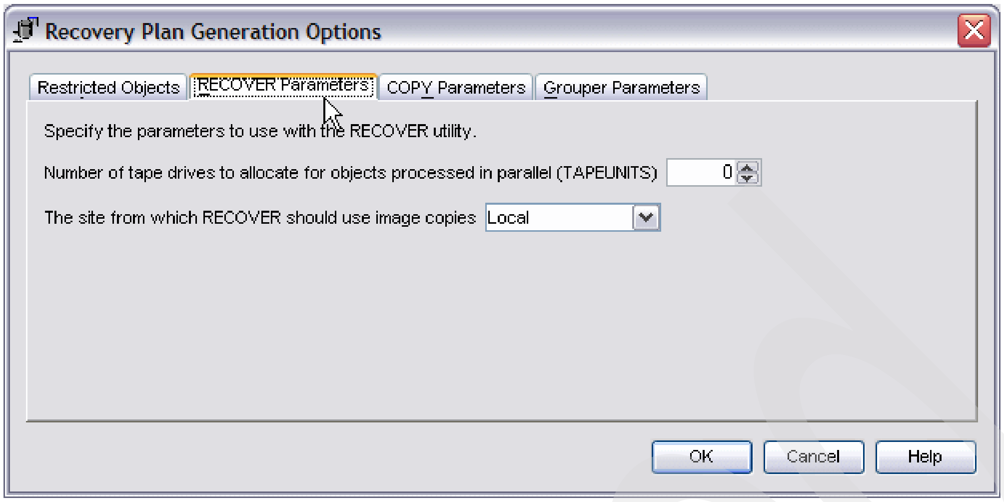

Figure 2-20 Recover parameters options

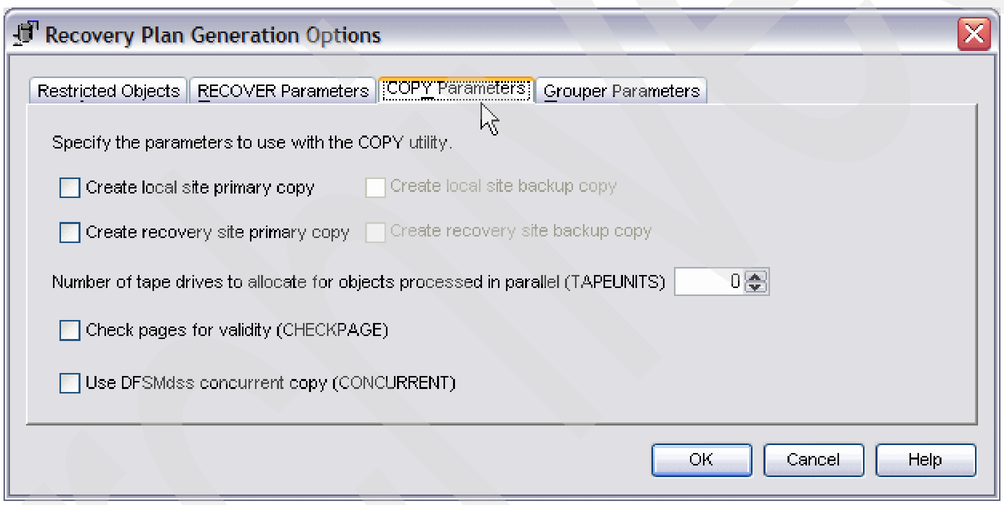

Figure 2-21 Copy parameters options

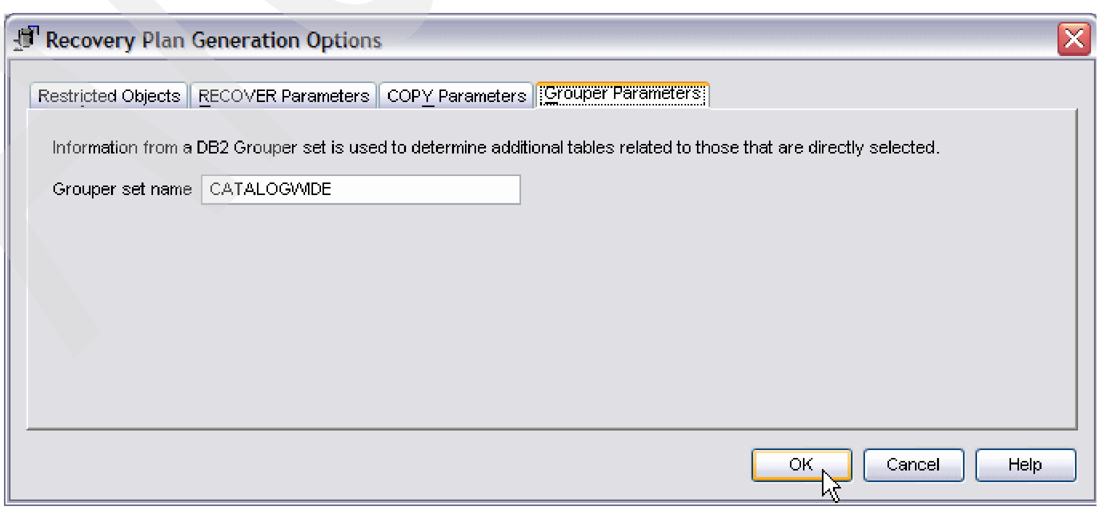

Figure 2-22 Grouper parameters options

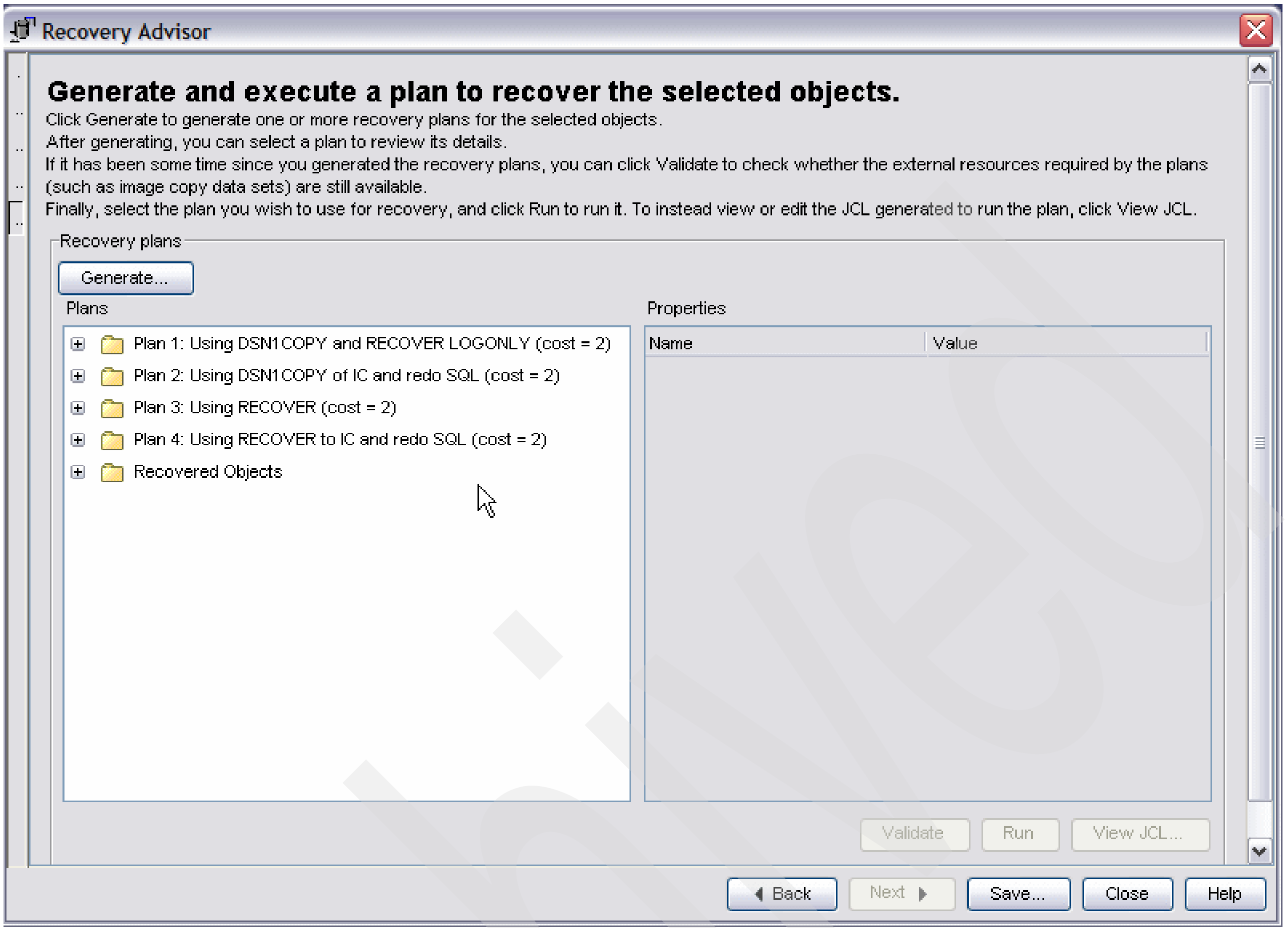

Figure 2-23 Generated recovery plans and recovered objects (1/4)

Figure 2-23 through Figure 2-26 on page 74 list the recovered objects and the potential recovery plans that you can run to achieve the recovery of the object to current. In this case, four possible recovery plans are available. These recovery plans are organized in descending sequence of efficiency[1] as determined by Recovery Expert for z/OS with Plan 1 being the most efficient and Plan 4 being the least efficient.

Figure 2-27 on page 74 shows the successful generation of recovery plans in the Messages section of the Launchpad window. Figure 2-28 on page 75 expands the steps involved in recovery Plan 3 and all the objects that are being recovered, which includes this table's dependencies such as table spaces and indexes.

[1] Efficiency is defined as speed of completing the recovery action.

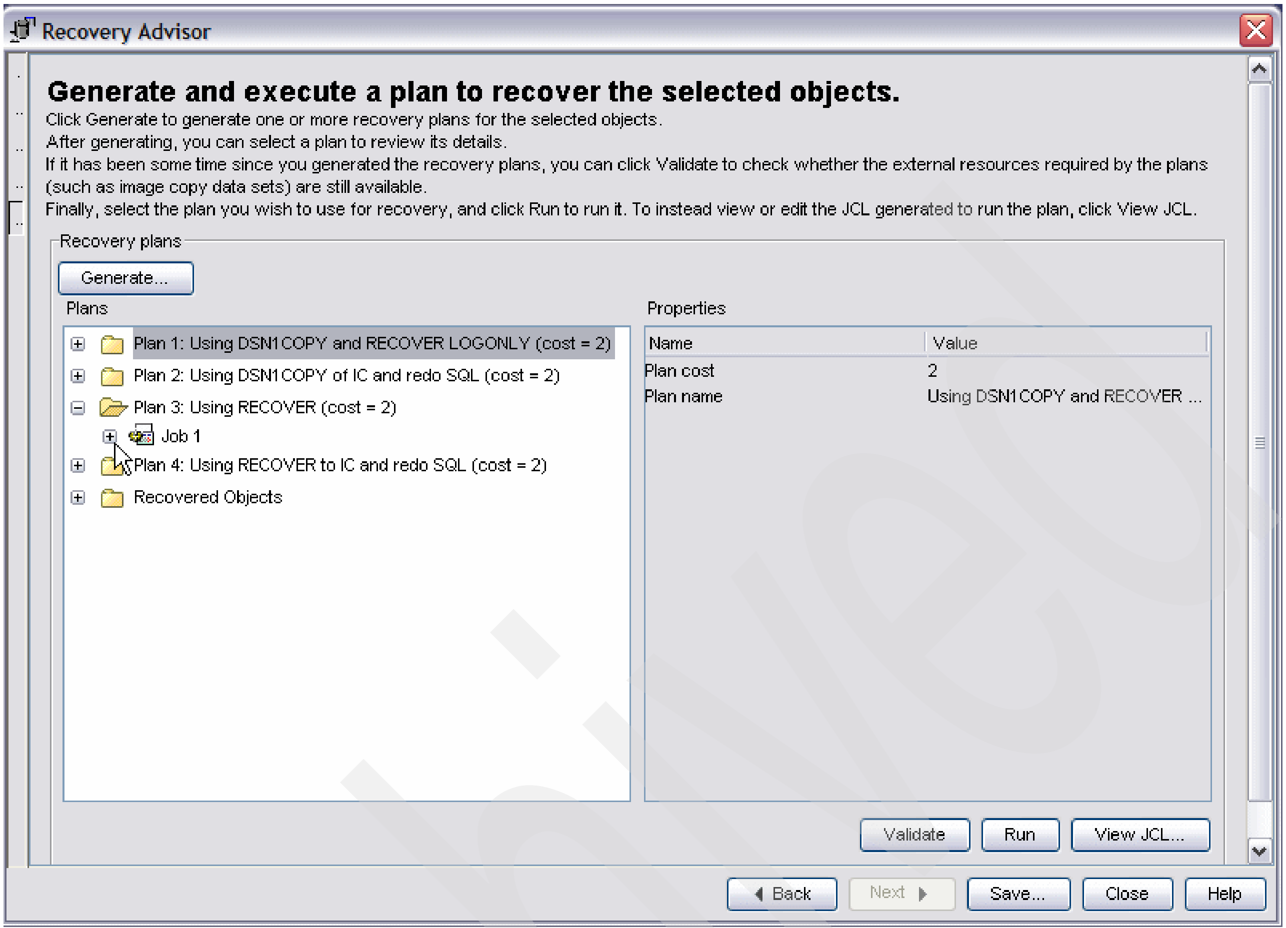

Figure 2-24 Generated recovery plans and recovered objects (2/4)

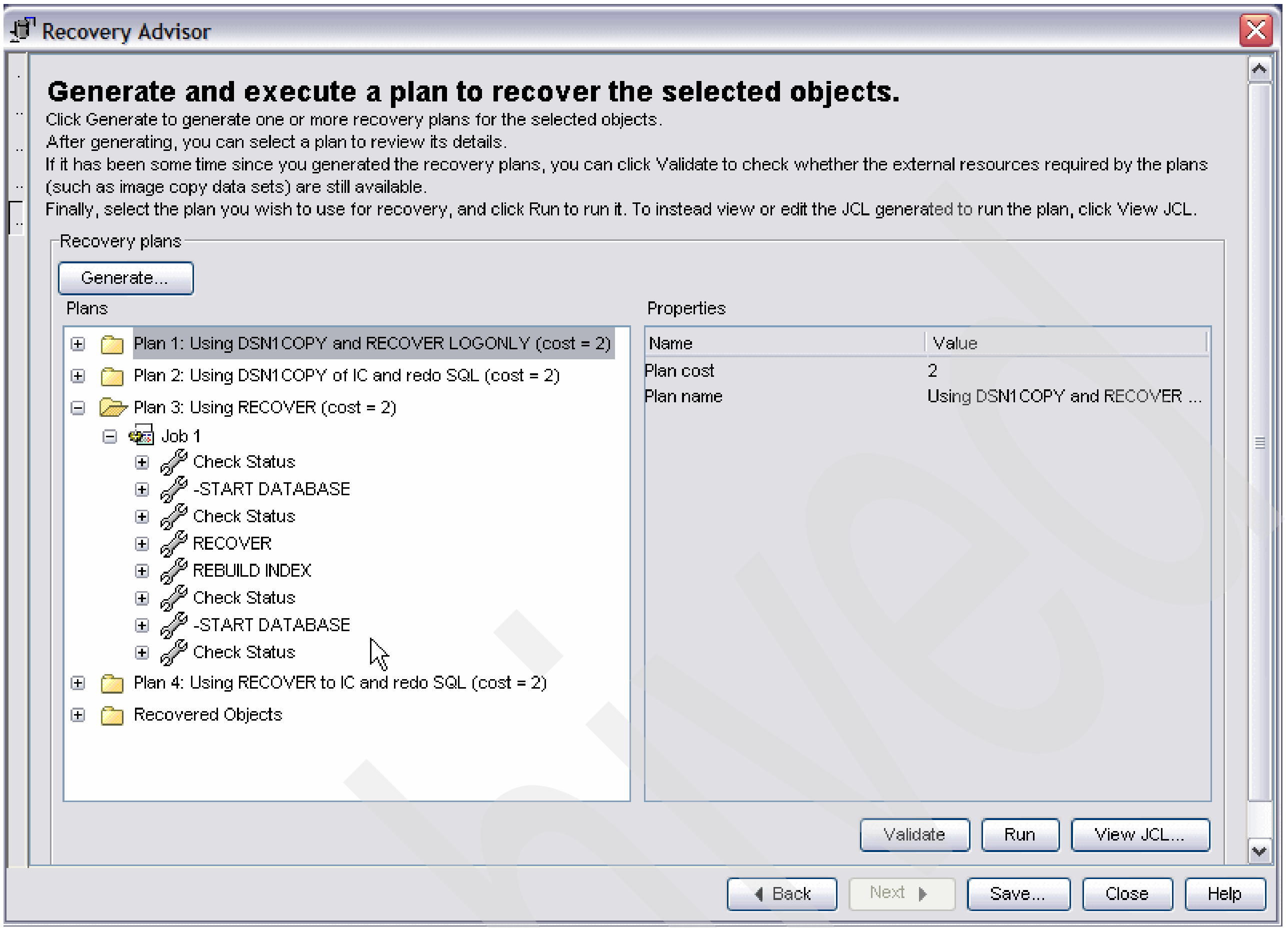

Figure 2-25   Generated recovery plans and recovered objects (3/4)

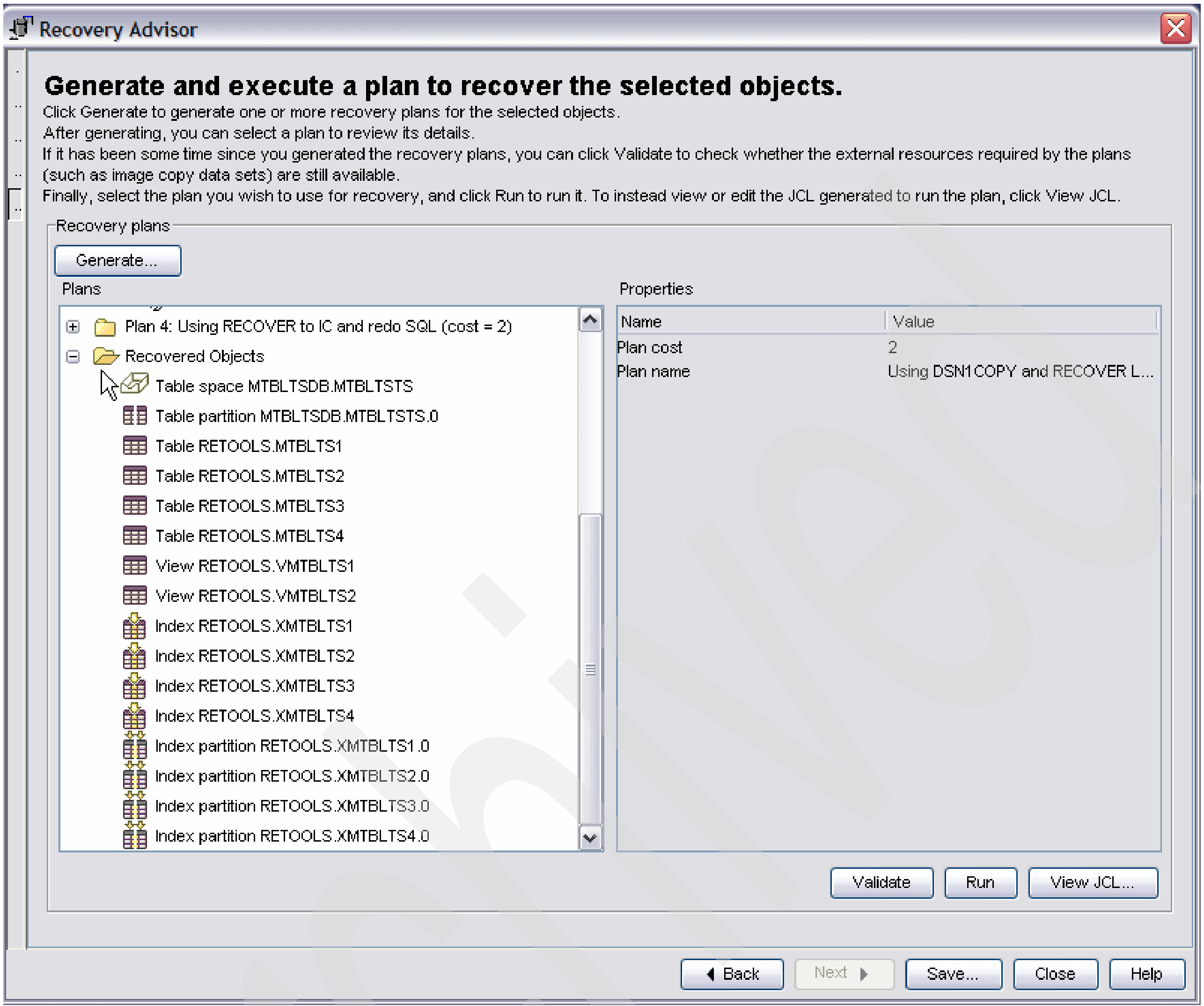

Figure 2-26 Generated recovery plans and recovered objects (4/4)

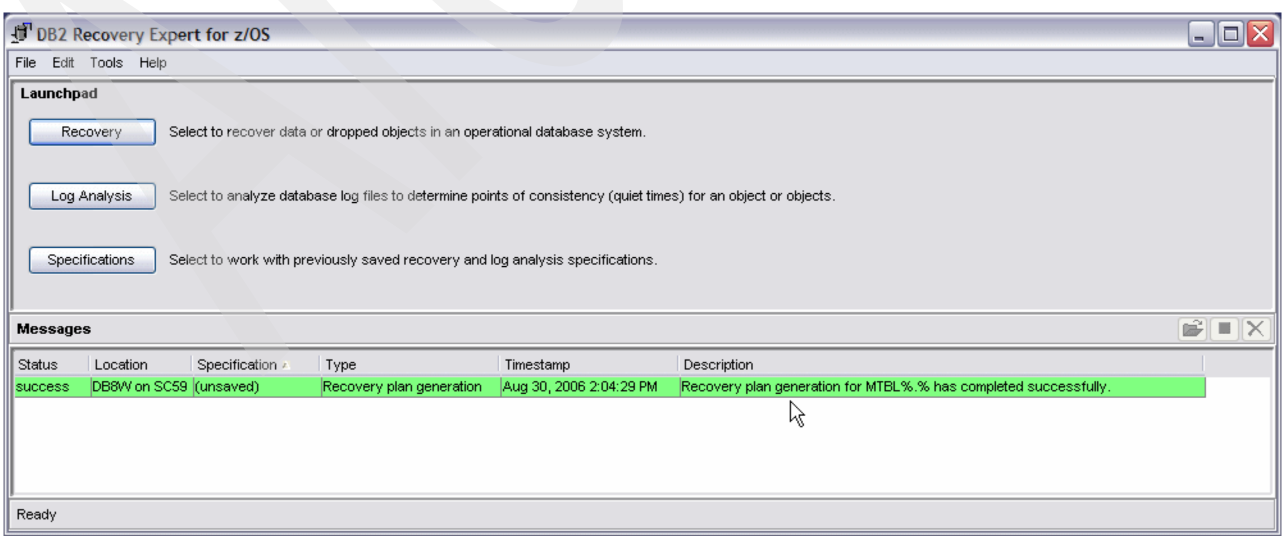

Figure 2-27 Recovery plan generation successful message in the messages section of the launchpad window

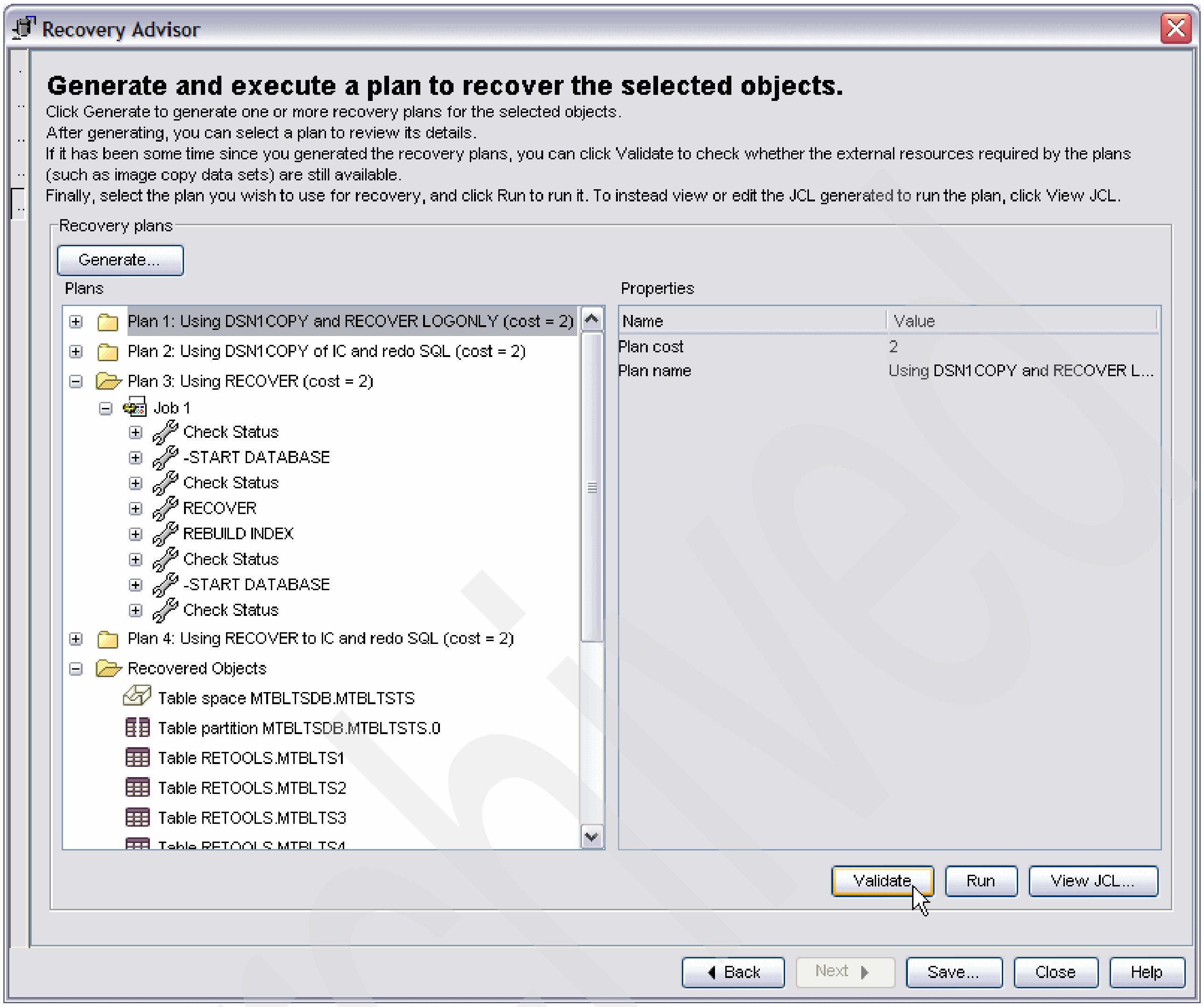

Figure 2-28 Plan 3: Steps in recovery plan and recovered objects list

Assuming that Plan 1 is the most efficient recovery plan as recommended by Recovery Expert for z/OS, highlight it and click the **Validate** button as shown in Figure 2-28. In this case, the validation is successful as indicated by Figure 2-29. Note the warning that successful validation is not a guarantee of successful execution.

Figure 2-29 Validation successful message

Click **OK** in Figure 2-29. Note the successful validation message in the Messages section of the Launchpad window, as shown in Figure 2-30.

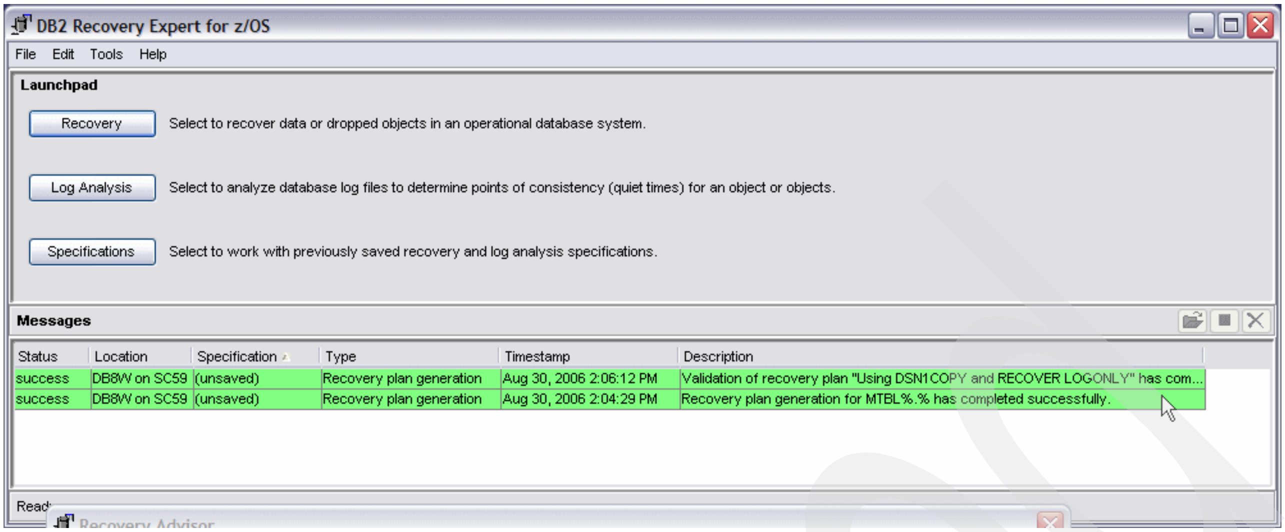

Figure 2-30 Successful validation message in the messages section of the launchpad window

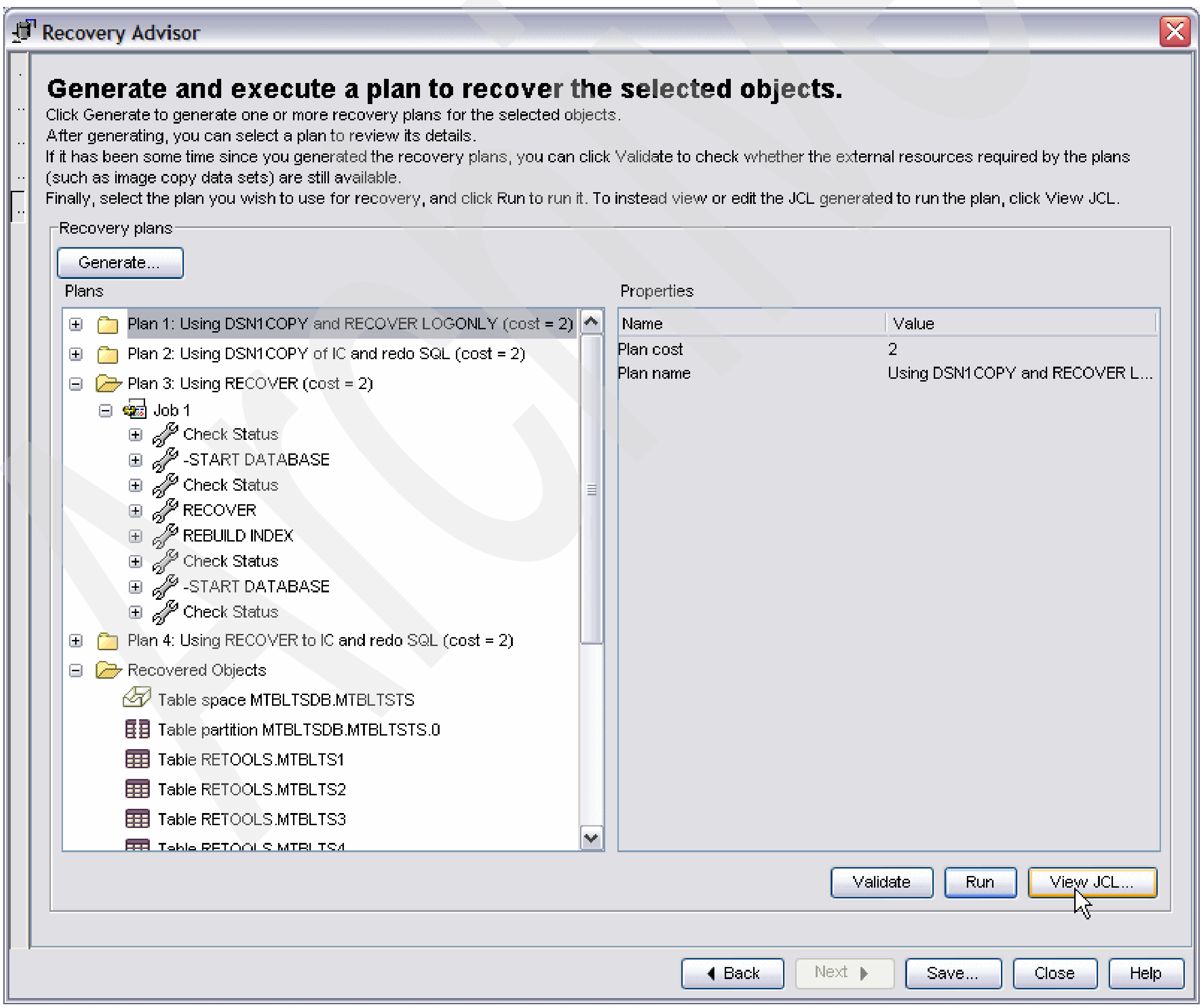

Figure 2-31 Viewing JCL (1/3)

After successful validation, click the **View JCL** button in Figure 2-31 to view the generated job control language (JCL) as shown in Figure 2-32[2] and Figure 2-33 on page 78. The partial contents of the generated JCL are also shown in Example 2-7 on page 83. Modify the JCL as appropriate after a careful review of the generated JCL, and click **Run** to run the generated JCL.

Figure 2-34 on page 79 shows the successful execution message in the Messages pane of the Launchpad window.

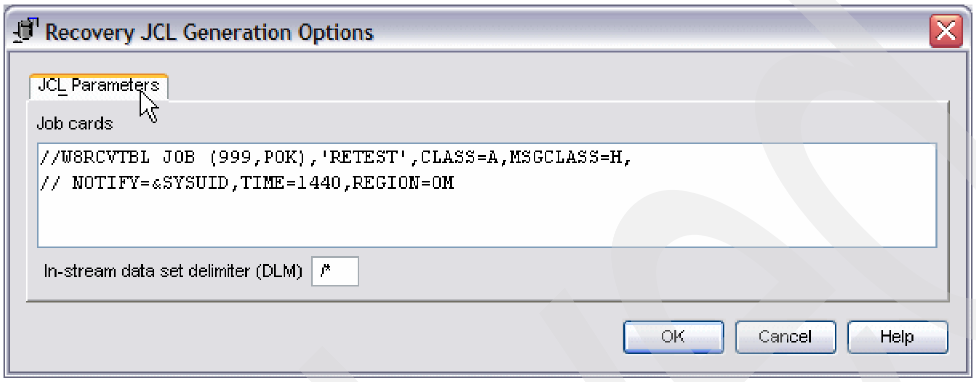

Figure 2-32 Viewing JCL (2/3)

2 You can modify the job card on this window and click **OK** to proceed to the full JCL shown in Figure 2-33 on page 78.

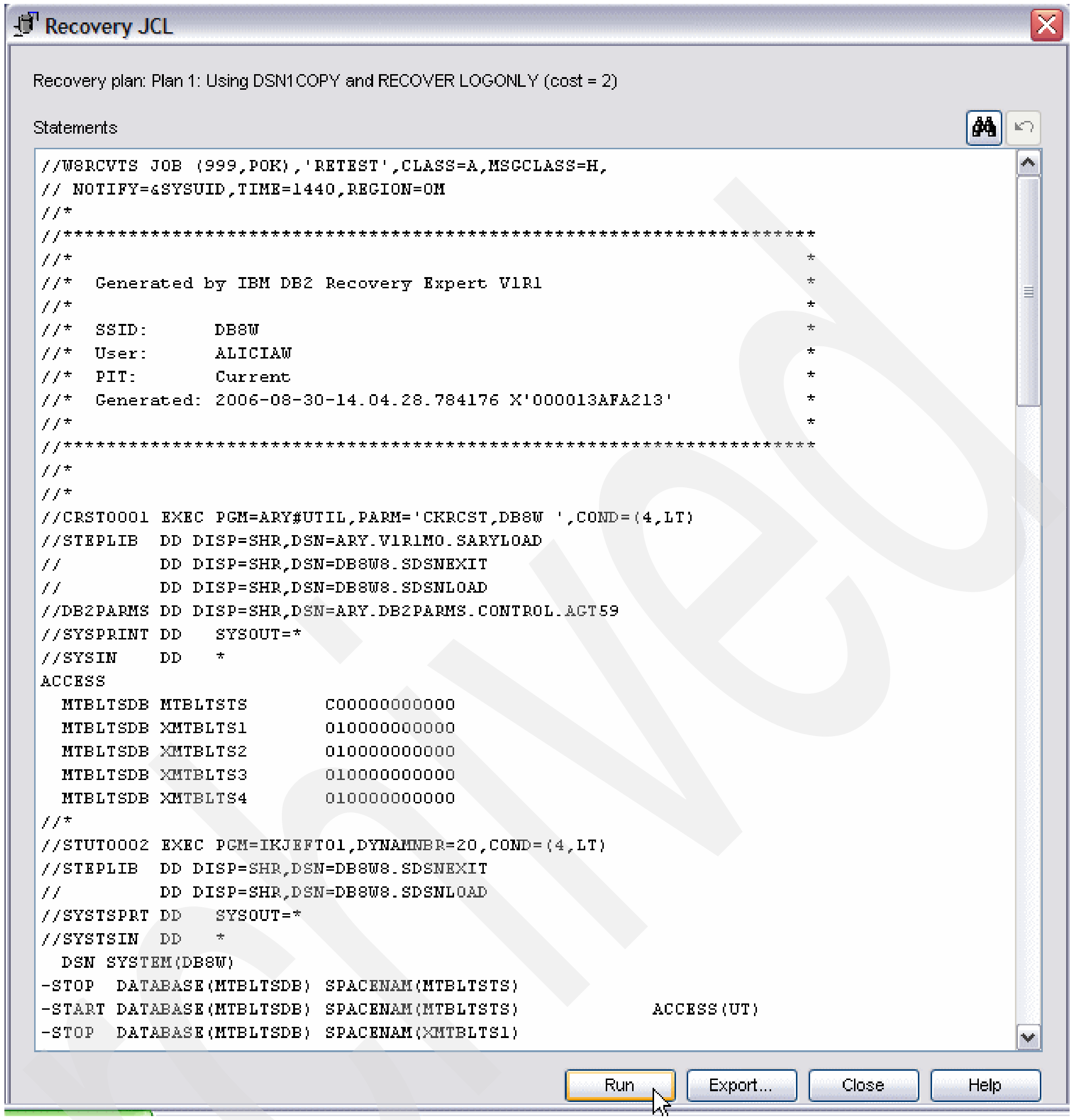

Figure 2-33 Viewing JCL (3/3)

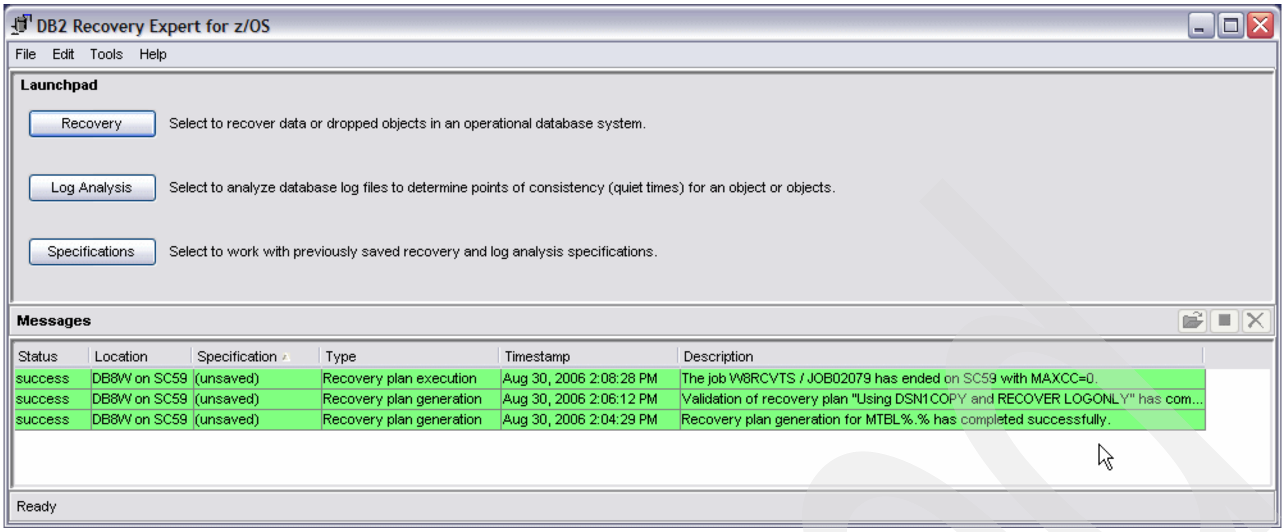

Figure 2-34   Plan 1: Successful execution of the generated JCL for in the launchpad window

You can view the recovery job results in the Recovery Expert for z/OS GUI client (or directly in z/OS using System Display and Search Facility (SDSF); the partial contents of this is shown in Example 2-8 on page 85) as shown in Figure 2-35 through Figure 2-37 on page 82. You can then disconnect from the Recovery Expert for z/OS GUI client using the Exit option from the File menu of the Launchpad window, as shown in Figure 2-38 on page 82.

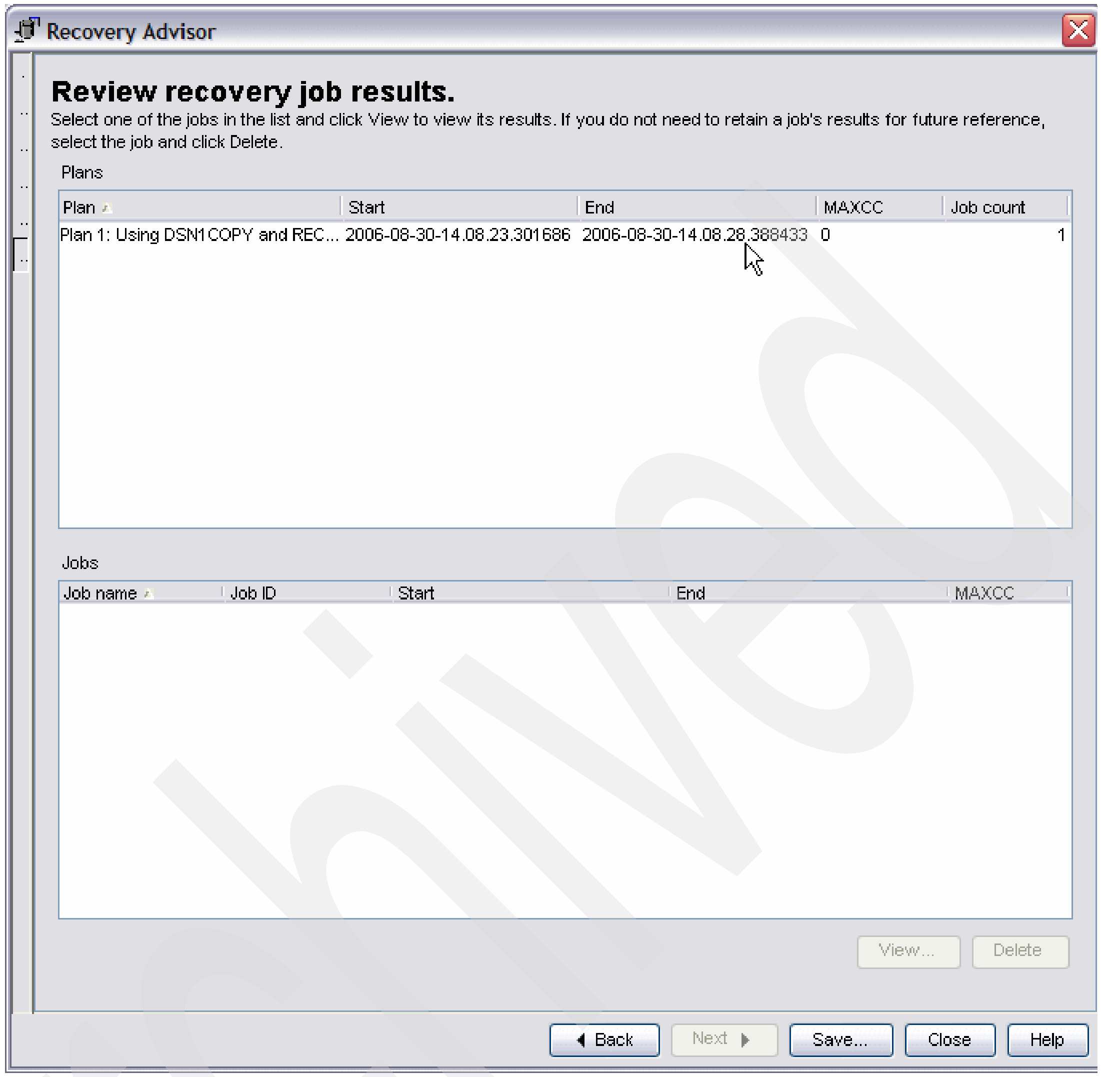

Figure 2-35 Recovery job results of Plan 1 (1/3)

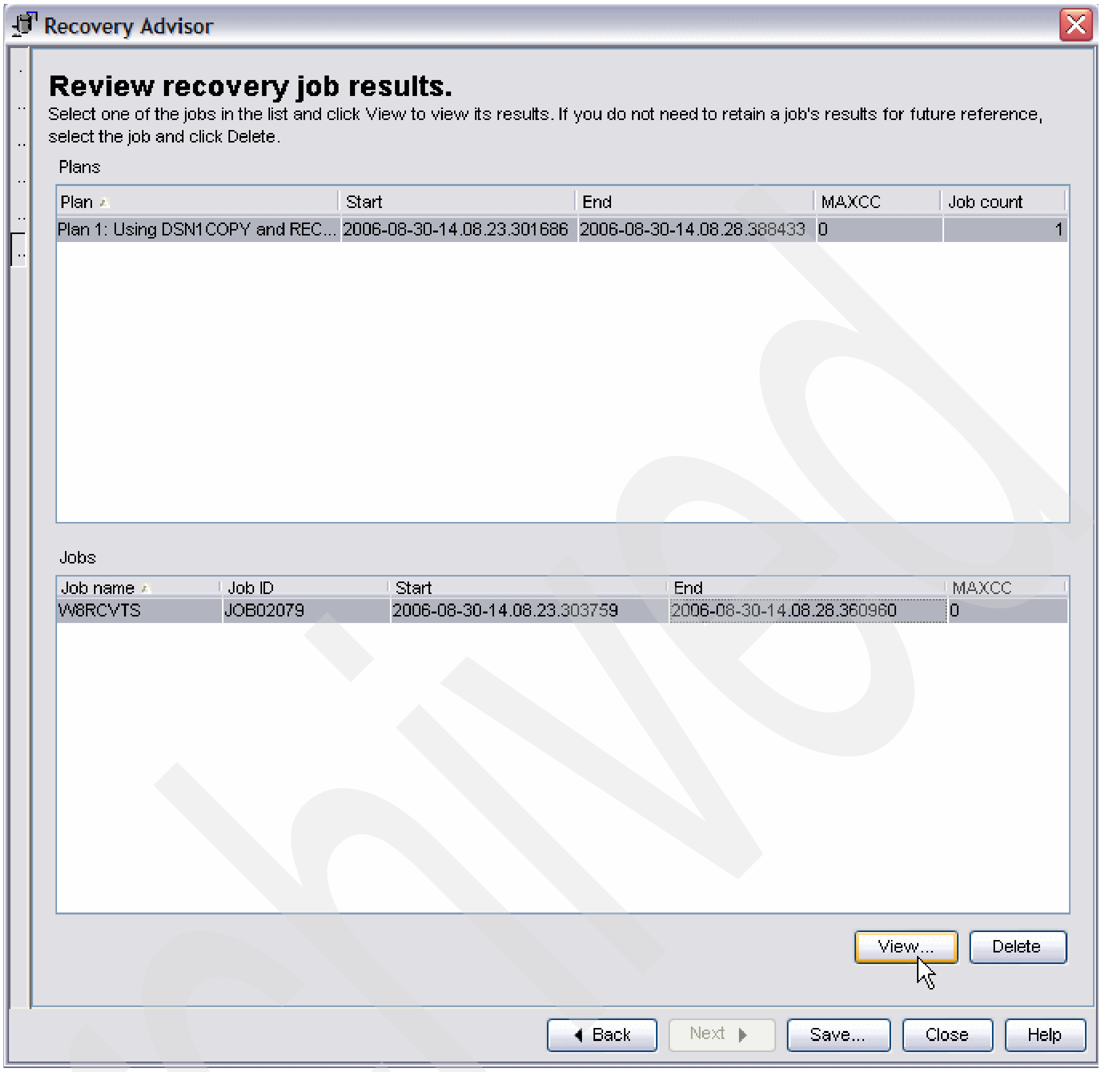

Figure 2-36 Recovery job results of Plan 1 (2/3)

The job ID shown in Figure 2-36 is the z/OS assigned job number.

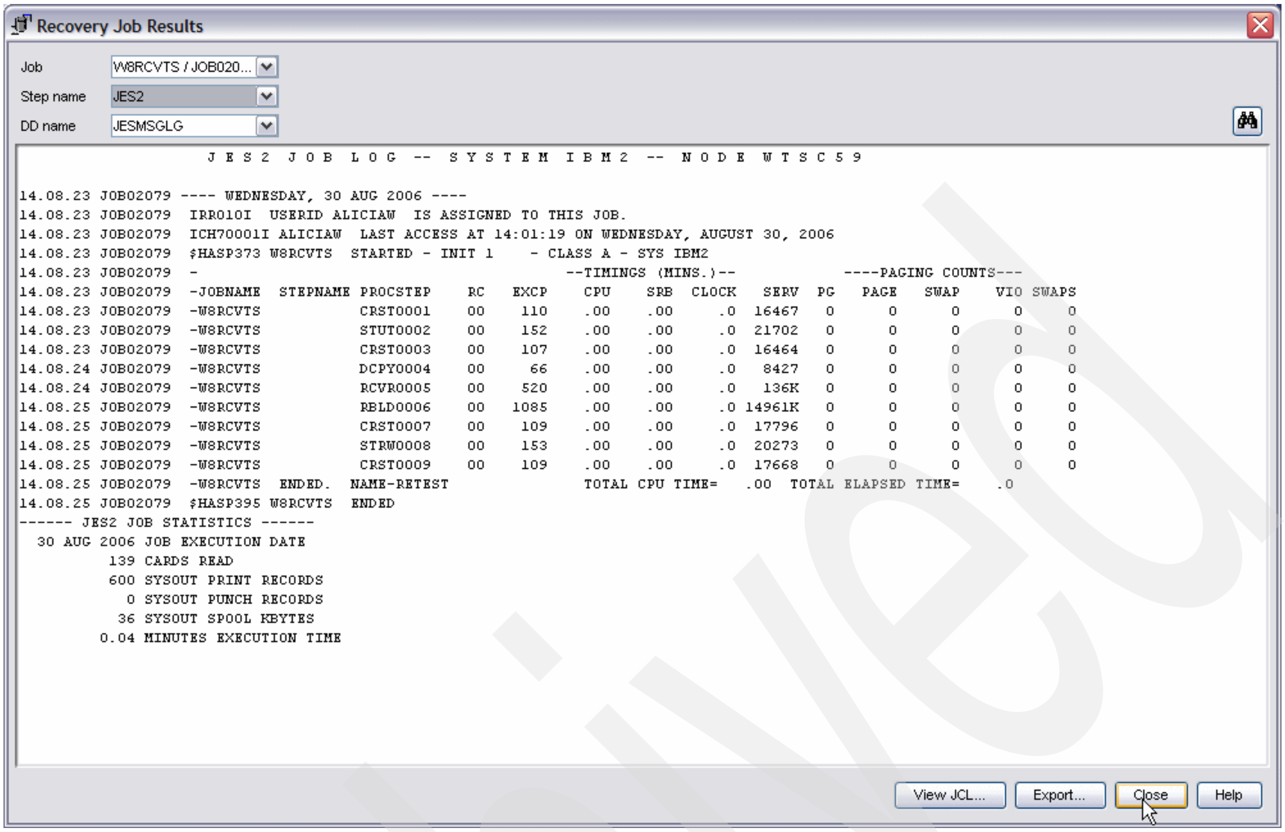

Figure 2-37 Recovery job results of Plan 1 (3/3)

In Figure 2-37, you can view all held print output data sets for all the steps in the job by selecting the appropriate step name and DD name.

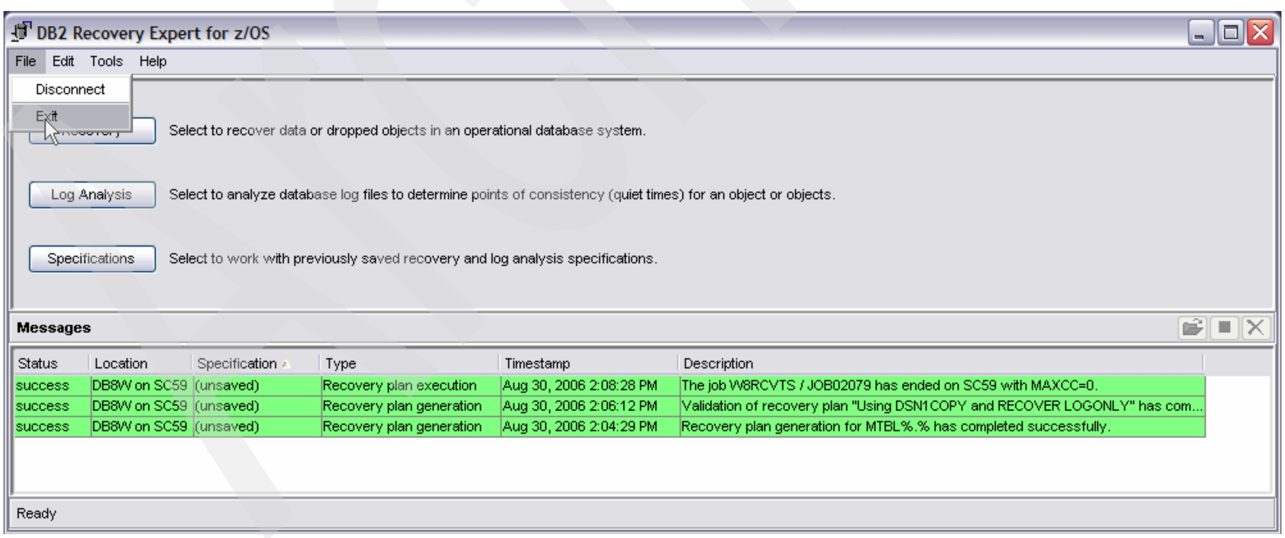

Figure 2-38 Exiting from the Recovery Expert for z/OS GUI client

Example 2-7 Generated JCL for Plan 1

```
//W8RCVTS JOB (999,POK),'RETEST',CLASS=A,MSGCLASS=H,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
//*
//*********************************************************************
//*                                                                   *
//*  Generated by IBM DB2 Recovery Expert V1R1                        *
//*                                                                   *
//*  SSID:      DB8W                                                  *
//*  User:      ALICIAW                                               *
//*  PIT:       Current                                               *
//*  Generated: 2006-08-30-14.04.28.784176 X'000013AFA213'            *
//*                                                                   *
//*********************************************************************
//*
//*
//CRST0001 EXEC PGM=ARY#UTIL,PARM='CKRCST,DB8W ',COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
ACCESS
  MTBLTSDB MTBLTSTS       C00000000000
  MTBLTSDB XMTBLTS1       010000000000
  MTBLTSDB XMTBLTS2       010000000000
  MTBLTSDB XMTBLTS3       010000000000
  MTBLTSDB XMTBLTS4       010000000000
//*
//STUT0002 EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
  DSN SYSTEM(DB8W)
-STOP  DATABASE(MTBLTSDB) SPACENAM(MTBLTSTS)
-START DATABASE(MTBLTSDB) SPACENAM(MTBLTSTS)            ACCESS(UT)
-STOP  DATABASE(MTBLTSDB) SPACENAM(XMTBLTS1)
-START DATABASE(MTBLTSDB) SPACENAM(XMTBLTS1)            ACCESS(UT)
-STOP  DATABASE(MTBLTSDB) SPACENAM(XMTBLTS2)
-START DATABASE(MTBLTSDB) SPACENAM(XMTBLTS2)            ACCESS(UT)
-STOP  DATABASE(MTBLTSDB) SPACENAM(XMTBLTS3)
-START DATABASE(MTBLTSDB) SPACENAM(XMTBLTS3)            ACCESS(UT)
-STOP  DATABASE(MTBLTSDB) SPACENAM(XMTBLTS4)
-START DATABASE(MTBLTSDB) SPACENAM(XMTBLTS4)            ACCESS(UT)
  END
//*
//CRST0003 EXEC PGM=ARY#UTIL,PARM='CKRCST,DB8W ',COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
```

```
ACCESS(UT)
  MTBLTSDB MTBLTSTS       C00000000000
  MTBLTSDB XMTBLTS1       010000000000
  MTBLTSDB XMTBLTS2       010000000000
  MTBLTSDB XMTBLTS3       010000000000
  MTBLTSDB XMTBLTS4       010000000000
//*
//DCPY0004 EXEC PGM=DSN1COPY,COND=(4,LT),
// PARM='CHECK,PAGESIZE(4K),FULLCOPY'
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSUT1   DD DISP=SHR,DSN=REIC.DB8W.MTBLTSDB.MTBLTSTS.FULL.G0002V00
//SYSUT2   DD DISP=SHR,DSN=DB8WUS.DSNDBC.MTBLTSDB.MTBLTSTS.I0001.A001
//SYSPRINT DD   SYSOUT=*
//*
//RCVR0005 EXEC PGM=DSNUTILB,PARM=(DB8W),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *

  RECOVER
    TABLESPACE MTBLTSDB.MTBLTSTS
      LOGONLY
      LOCALSITE

/*
//*
//RBLD0006 EXEC PGM=DSNUTILB,PARM=(DB8W),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//UTPRINT  DD   SYSOUT=*
//SYSOUT   DD   UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN    DD   *

  REBUILD INDEX
    ( "RETOOLS"."XMTBLTS1"
    , "RETOOLS"."XMTBLTS2"
    , "RETOOLS"."XMTBLTS3"
    , "RETOOLS"."XMTBLTS4" )
      SCOPE ALL

/*
//*
//CRST0007 EXEC PGM=ARY#UTIL,PARM='CKRCST,DB8W ',COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
ACCESS(UT)
  MTBLTSDB MTBLTSTS       000000000000
  MTBLTSDB XMTBLTS1       000000000000
```

```
  MTBLTSDB XMTBLTS2       000000000000
  MTBLTSDB XMTBLTS3       000000000000
  MTBLTSDB XMTBLTS4       000000000000
//*
//STRW0008 EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
  DSN SYSTEM(DB8W)
-START DATABASE(MTBLTSDB) SPACENAM(MTBLTSTS)            ACCESS(RW)
-START DATABASE(MTBLTSDB) SPACENAM(XMTBLTS1)            ACCESS(RW)
-START DATABASE(MTBLTSDB) SPACENAM(XMTBLTS2)            ACCESS(RW)
-START DATABASE(MTBLTSDB) SPACENAM(XMTBLTS3)            ACCESS(RW)
-START DATABASE(MTBLTSDB) SPACENAM(XMTBLTS4)            ACCESS(RW)
  END
//*
//CRST0009 EXEC PGM=ARY#UTIL,PARM='CKRCST,DB8W ',COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
ACCESS(RW)
  MTBLTSDB MTBLTSTS       000000000000
  MTBLTSDB XMTBLTS1       000000000000
  MTBLTSDB XMTBLTS2       000000000000
  MTBLTSDB XMTBLTS3       000000000000
  MTBLTSDB XMTBLTS4       000000000000
```

Example 2-7 highlights the operations performed on the objects involved in the recovery process such as putting the table spaces in utility status, restoring the latest full image copy recorded in SYSIBM.SYSCOPY using DSN1COPY, performing a RECOVER ...LOGONLY on the table space, rebuilding all the indexes, and then restarting the table spaces.

This option recovers all the tables in the multi-table table space and rebuilds all the indexes of all the tables in the table space. You might consider looking at the other recovery plans that are available if you consider them to be more efficient in your particular scenario.

Example 2-8 shows the partial contents of the executed JCL output.

Example 2-8 Partial contents of the executed JCL output

```
1                    J E S 2  J O B  L O G  --  S Y S T E M  I B M 2  --  N O D E  W T S C 5 9
0
 14.08.23 JOB02079 ---- WEDNESDAY, 30 AUG 2006 ----
 14.08.23 JOB02079  IRR010I  USERID ALICIAW  IS ASSIGNED TO THIS JOB.
 14.08.23 JOB02079  ICH70001I ALICIAW  LAST ACCESS AT 14:01:19 ON WEDNESDAY, AUGUST 30, 2006
 14.08.23 JOB02079  $HASP373 W8RCVTS  STARTED - INIT 1    - CLASS A - SYS IBM2
 14.08.23 JOB02079  -                                       --TIMINGS (MINS.)--
...............
...............
0------ JES2 JOB STATISTICS ------
-  30 AUG 2006 JOB EXECUTION DATE
-          139 CARDS READ
```

```
-          600 SYSOUT PRINT RECORDS
-            0 SYSOUT PUNCH RECORDS
-           36 SYSOUT SPOOL KBYTES
-         0.04 MINUTES EXECUTION TIME
...............
...............
*MSG* -DISPLAY UTILITY(ALICIAW.W8RCVTS)
 *MSG* DSNU112I  -DB8W DSNUGDIS - NO AUTHORIZED UTILITY FOUND FOR UTILID =
 *MSG* ALICIAW.W8RCVTS
 *MSG* DSN9022I  -DB8W DSNUGCCC '-DISPLAY UTILITY' NORMAL COMPLETION

 00001 ACCESS
 00002   MTBLTSDB MTBLTSTS       C00000000000                                            RW
 00003   MTBLTSDB XMTBLTS1       010000000000                                            RW
 00004   MTBLTSDB XMTBLTS2       010000000000                                            RW
 00005   MTBLTSDB XMTBLTS3       010000000000                                            RW
 00006   MTBLTSDB XMTBLTS4       010000000000                                            RW

 *** End of Report ***
1READY
   DSN SYSTEM(DB8W)
 DSN
 -STOP  DATABASE(MTBLTSDB) SPACENAM(MTBLTSTS)
 DSN9022I  -DB8W DSNTDDIS 'STOP DATABASE' NORMAL COMPLETION
 DSN
 -START DATABASE(MTBLTSDB) SPACENAM(MTBLTSTS)            ACCESS(UT)
 DSN9022I  -DB8W DSNTDDIS 'START DATABASE' NORMAL COMPLETION
 DSN
 -STOP  DATABASE(MTBLTSDB) SPACENAM(XMTBLTS1)
 DSN9022I  -DB8W DSNTDDIS 'STOP DATABASE' NORMAL COMPLETION
 DSN
 -START DATABASE(MTBLTSDB) SPACENAM(XMTBLTS1)            ACCESS(UT)
 DSN9022I  -DB8W DSNTDDIS 'START DATABASE' NORMAL COMPLETION
 DSN
 -STOP  DATABASE(MTBLTSDB) SPACENAM(XMTBLTS2)
 DSN9022I  -DB8W DSNTDDIS 'STOP DATABASE' NORMAL COMPLETION
 DSN
 -START DATABASE(MTBLTSDB) SPACENAM(XMTBLTS2)            ACCESS(UT)
 DSN9022I  -DB8W DSNTDDIS 'START DATABASE' NORMAL COMPLETION
 DSN
 -STOP  DATABASE(MTBLTSDB) SPACENAM(XMTBLTS3)
 DSN9022I  -DB8W DSNTDDIS 'STOP DATABASE' NORMAL COMPLETION
 DSN
 -START DATABASE(MTBLTSDB) SPACENAM(XMTBLTS3)            ACCESS(UT)
 DSN9022I  -DB8W DSNTDDIS 'START DATABASE' NORMAL COMPLETION
 DSN
 -STOP  DATABASE(MTBLTSDB) SPACENAM(XMTBLTS4)
 DSN9022I  -DB8W DSNTDDIS 'STOP DATABASE' NORMAL COMPLETION
 DSN
 -START DATABASE(MTBLTSDB) SPACENAM(XMTBLTS4)            ACCESS(UT)
 DSN9022I  -DB8W DSNTDDIS 'START DATABASE' NORMAL COMPLETION
 DSN
   END
 READY
 END
```

```
.................
.................
1DSN1999I START OF DSN1COPY FOR JOB W8RCVTS  DCPY0004
 DSN1989I DSN1COPY IS PROCESSED WITH THE FOLLOWING OPTIONS:
             CHECK/NO PRINT/ 4K/FULLCOPY    /NON-SEGMENT/NUMPARTS=   0/NO OBIDXLAT/NO VALUE/NO
RESET/     /   /        /
           DSSIZE=   /PIECESIZ=    /       /
 DSN1998I INPUT  DSNAME = REIC.DB8W.MTBLTSDB.MTBLTSTS.FULL.G0002V00   , SEQ
 DSN1997I OUTPUT DSNAME = DB8WUS.DSNDBC.MTBLTSDB.MTBLTSTS.I0001.A001  , VSAM
0DSN1994I DSN1COPY COMPLETED SUCCESSFULLY,  00000006 PAGES PROCESSED

1DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = ALICIAW.W8RCVTS
 DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
0DSNU050I    DSNUGUTC -  RECOVER TABLESPACE MTBLTSDB.MTBLTSTS LOGONLY LOCALSITE
 DSNU532I  -DB8W DSNUCALA - RECOVER TABLESPACE MTBLTSDB.MTBLTSTS   START
 DSNU549I  -DB8W DSNUCALA - RECOVER TABLESPACE MTBLTSDB.MTBLTSTS
                       USES ONLY DB2 LOGS STARTING FROM LOGPOINT=X'000013A86EDD'.
 DSNU513I  -DB8W DSNUCALA - RECOVER UTILITY LOG APPLY RANGE IS RBA 000013ABDECE LRSN
000013ABDECE TO RBA 000013AE798A LRSN 000013AE798A
 DSNU1510I   DSNUCBDR - LOG APPLY PHASE COMPLETE, ELAPSED TIME = 00:00:00
 DSNU500I    DSNUCBDR - RECOVERY COMPLETE, ELAPSED TIME=00:00:00
 DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
1DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = ALICIAW.W8RCVTS
 DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
0DSNU050I    DSNUGUTC -  REBUILD INDEX("RETOOLS"."XMTBLTS1", "RETOOLS"."XMTBLTS2",
"RETOOLS"."XMTBLTS3",
 "RETOOLS"."XMTBLTS4") SCOPE ALL
 DSNU395I    DSNUCRIB - INDEXES WILL BE BUILT IN PARALLEL, NUMBER OF TASKS = 9
 DSNU555I  -DB8W DSNUCRUL - UNLOAD PHASE STATISTICS - NUMBER OF RECORDS PROCESSED=60
 DSNU705I    DSNUCRIB - UNLOAD PHASE COMPLETE - ELAPSED TIME=00:00:00
 DSNU394I  -DB8W DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=14 FOR INDEX
RETOOLS.XMTBLTS1
 DSNU394I  -DB8W DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=14 FOR INDEX
RETOOLS.XMTBLTS4
 DSNU394I  -DB8W DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=16 FOR INDEX
RETOOLS.XMTBLTS3
 DSNU394I  -DB8W DSNURBXC - SORTBLD PHASE STATISTICS - NUMBER OF KEYS=16 FOR INDEX
RETOOLS.XMTBLTS2
 DSNU391I    DSNUCRIB - SORTBLD PHASE STATISTICS. NUMBER OF INDEXES = 4
 DSNU392I    DSNUCRIB - SORTBLD PHASE COMPLETE, ELAPSED TIME = 00:00:01
 DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
.............
.............
DSN SYSTEM(DB8W)
 DSN
 -START DATABASE(MTBLTSDB) SPACENAM(MTBLTSTS)            ACCESS(RW)
 DSN9022I  -DB8W DSNTDDIS 'START DATABASE' NORMAL COMPLETION
 DSN
 -START DATABASE(MTBLTSDB) SPACENAM(XMTBLTS1)            ACCESS(RW)
 DSN9022I  -DB8W DSNTDDIS 'START DATABASE' NORMAL COMPLETION
 DSN
 -START DATABASE(MTBLTSDB) SPACENAM(XMTBLTS2)            ACCESS(RW)
 DSN9022I  -DB8W DSNTDDIS 'START DATABASE' NORMAL COMPLETION
 DSN
 -START DATABASE(MTBLTSDB) SPACENAM(XMTBLTS3)            ACCESS(RW)
```

```
DSN9022I  -DB8W DSNTDDIS 'START DATABASE' NORMAL COMPLETION
DSN
-START DATABASE(MTBLTSDB) SPACENAM(XMTBLTS4)          ACCESS(RW)
DSN9022I  -DB8W DSNTDDIS 'START DATABASE' NORMAL COMPLETION
DSN
  END
READY
END
......................
```

2.3.2 Recovering multiple table spaces (volume list)

This scenario describes the recovery of multiple table spaces in different databases to current, which simulates a volume failure. This environment is a non-data sharing environment.

The following windows only represent the ones that are of particular interest (in our opinion) to this scenario. The windows leading up to these windows and those that follow these windows are almost identical to the windows described in 2.3.1, "Recovering multi-table table space" on page 56. We show the partial contents of the generated JCL for a given recovery plan, but not the execution job results.

Figure 2-39 on page 89 shows the list of selected table spaces for recovery by navigating through the table spaces in the navigation pane. Figure 2-40 on page 90 shows the recovery point for these objects (to current). Figure 2-41 on page 91 shows the four recovery plans that are generated for the requested recovery action, and the expanded steps included in the recovery plan, Plan 4. Because Plan 4 contains a log analysis step, a warning is issued to this effect, as shown in Figure 2-42 on page 92. A portion of the generated JCL for Plan 4 is shown in Example 2-9 on page 93.

The selected recovery plan, Plan 4, has steps to restore the various table spaces to the latest image copies using the RECOVER statement followed by a rebuilding of all indexes. It is then followed by the generation of redo Structured Query Language (SQL) from the DB2 log using log analysis for table spaces on the same volume that encountered a direct access storage device (DASD) failure. The log analysis is bounded by the end of the log (relative byte address (RBA) FFFFFFFFFFFF) and the starting RBA (000014DDD625) corresponding to the earliest timestamp of all the image copies used in this recovery action.

However, the table space on volume VOLDB4DB is recovered using the RECOVER TABLESPACE statement followed by a rebuilding of all the indexes associated with this table. The reason this table space is treated differently is because there is no full image copy for this table space, and it is recovered to current using only the DB2 logs.

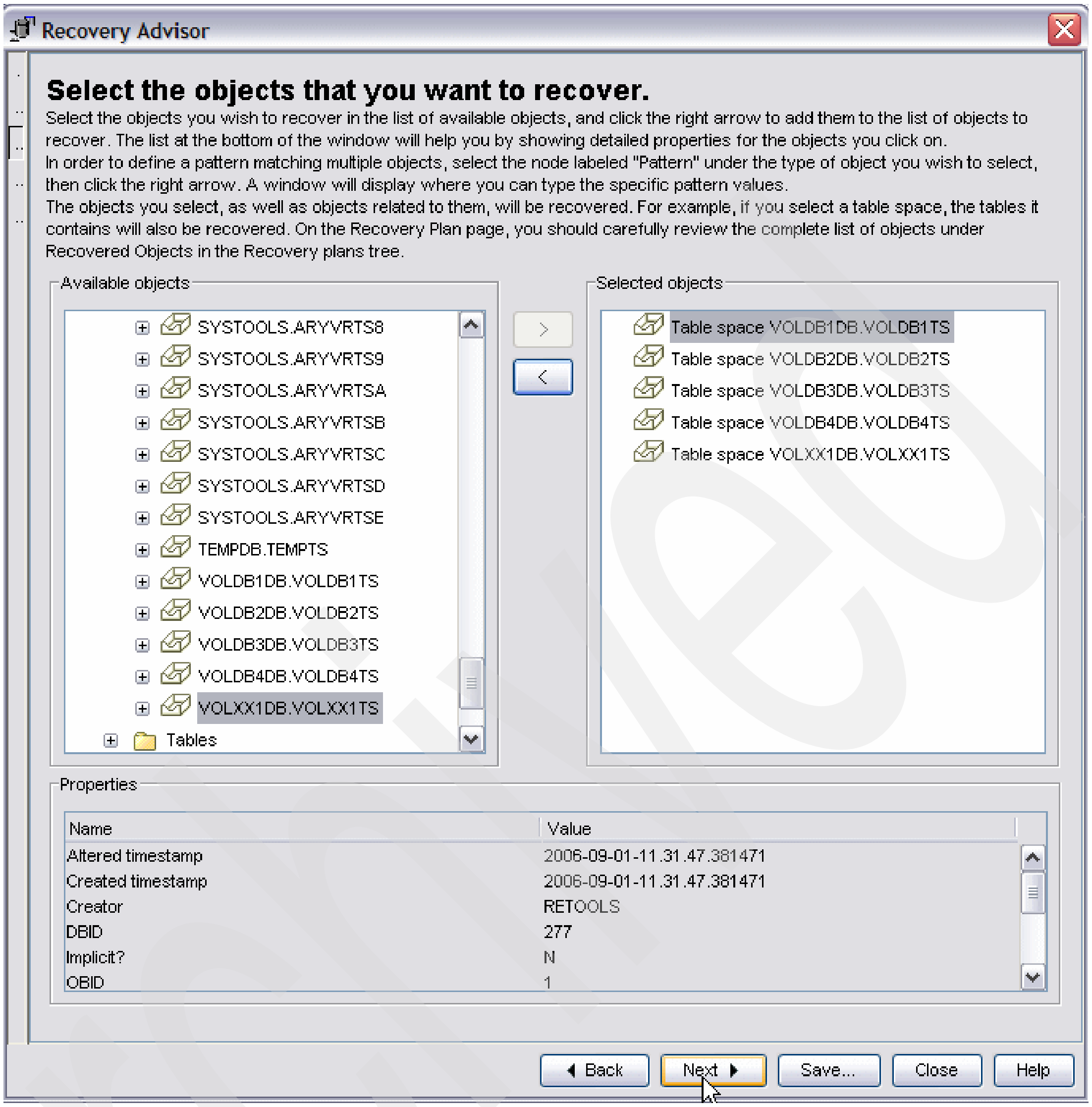

Figure 2-39 Selecting the objects to recover

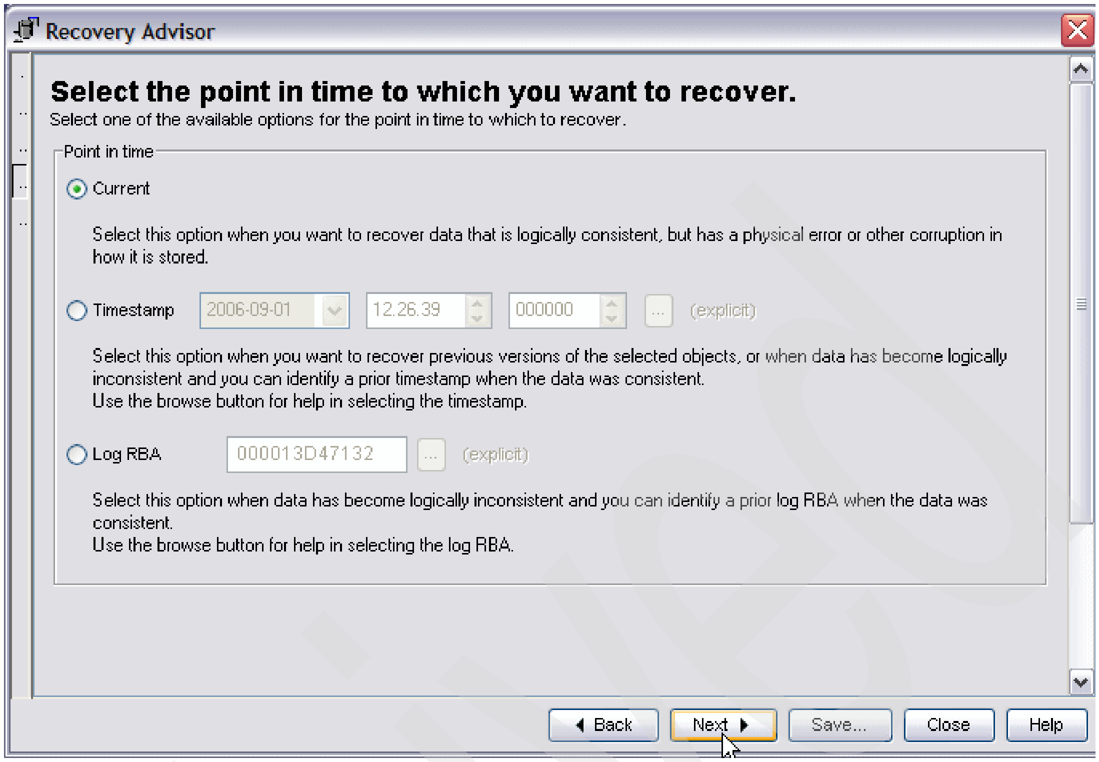

Figure 2-40 Selecting the point to recover to

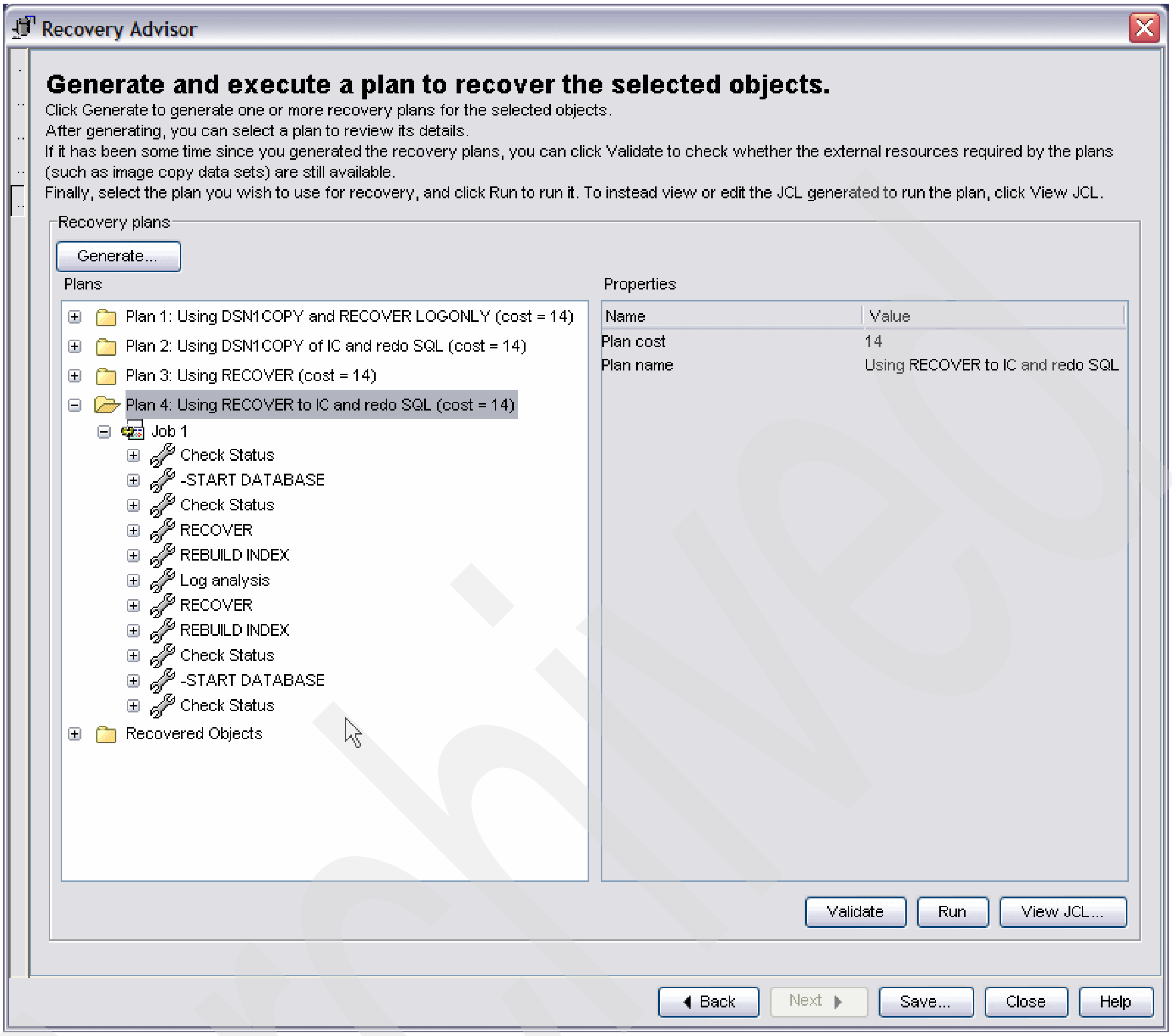

Figure 2-41   Generated recovery plans

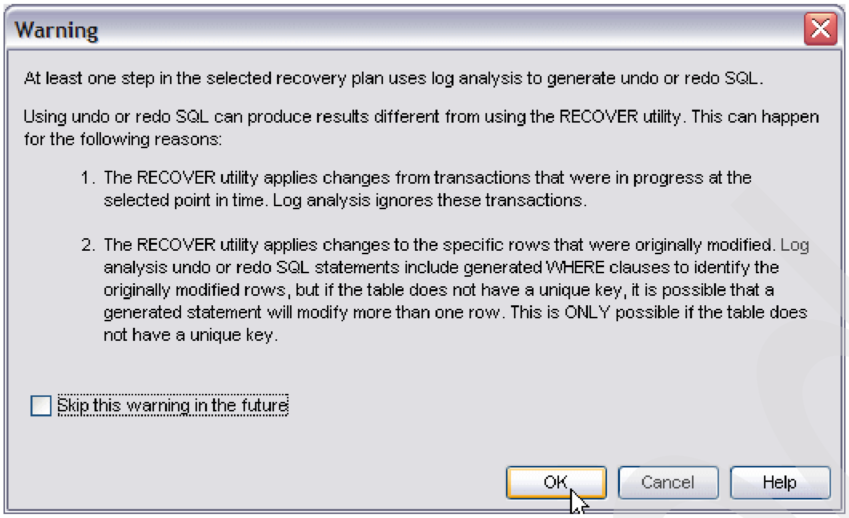

Figure 2-42 Warning message regarding log analysis step in the selected recovery plan

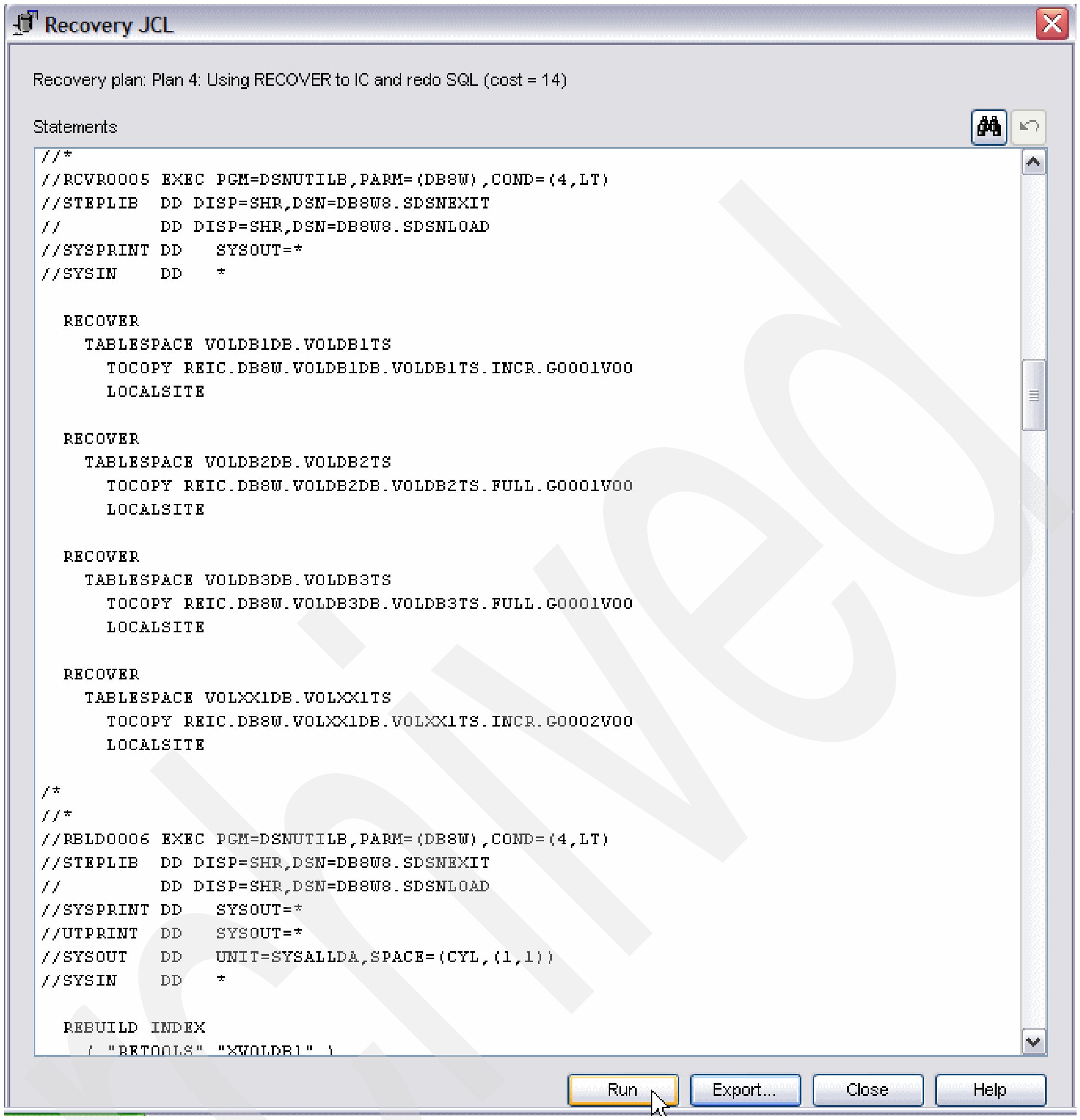

Figure 2-43 A section of generated JCL for Plan 4

Example 2-9 Partial contents of the generated JCL for Plan 4

```
//W8MLTPLE JOB (999,POK),'RETEST',CLASS=A,MSGCLASS=H,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
............
............
//DFLT0004 EXEC PGM=ARYLDFLT,PARM=(DB8W),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//SYSPRINT DD   SYSOUT=*
//FILTERS  DD DISP=SHR,DSN=NALUR.ARYLAS.FILTERS.R0000695
//*
//RCVR0005 EXEC PGM=DSNUTILB,PARM=(DB8W),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
```

```
  RECOVER
    TABLESPACE VOLDB1DB.VOLDB1TS
      TOCOPY REIC.DB8W.VOLDB1DB.VOLDB1TS.INCR.G0001V00
      LOCALSITE

  RECOVER
    TABLESPACE VOLDB2DB.VOLDB2TS
      TOCOPY REIC.DB8W.VOLDB2DB.VOLDB2TS.FULL.G0001V00
      LOCALSITE

  RECOVER
    TABLESPACE VOLDB3DB.VOLDB3TS
      TOCOPY REIC.DB8W.VOLDB3DB.VOLDB3TS.FULL.G0001V00
      LOCALSITE

  RECOVER
    TABLESPACE VOLXX1DB.VOLXX1TS
      TOCOPY REIC.DB8W.VOLXX1DB.VOLXX1TS.INCR.G0002V00
      LOCALSITE
/*
//*
//RBLD0006 EXEC PGM=DSNUTILB,PARM=(DB8W),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//UTPRINT  DD   SYSOUT=*
//SYSOUT   DD   UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN    DD   *

  REBUILD INDEX
    ( "RETOOLS"."XVOLDB1" )
      SCOPE ALL

  REBUILD INDEX
    ( "RETOOLS"."XVOLDB2" )
      SCOPE ALL

  REBUILD INDEX
    ( "RETOOLS"."XVOLDB3" )
      SCOPE ALL

  REBUILD INDEX
    ( "RETOOLS"."XVOLXX11"
    , "RETOOLS"."XVOLXX12"
    , "RETOOLS"."XVOLXX13"
    , "RETOOLS"."XVOLXX14" )
      SCOPE ALL

/*
..................
..................
//*********************************************************************
//* STEP 2: READ THE DB2 LOG TO GENERATE THE GENERAL REPORT            *
//*********************************************************************
//LASR0002 EXEC PGM=ARYGEN1,REGION=0M,COND=(4,LT)
```

```
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//MODEFILE DD   DSN=NALUR.ARYLAS.MODE.R0000695,
// DISP=OLD
//SYSOUT   DD   SYSOUT=*
//CFILES   DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//GENRPT   DD   SYSOUT=*
//EXTREP   DD   SYSOUT=*
//SUMREPT  DD   SYSOUT=*
//XDREPT   DD   SYSOUT=*
//QTRPT    DD   SYSOUT=*
//WARNINGS DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//DATAIN   DD   *
SSID          =DB8W
START RBA     =000014DDD625
END RBA       =FFFFFFFFFFFF
/*
.................
.................
//LASR0006 EXEC PGM=ARYDTL4,REGION=0M,COND=(4,LT)
//*******************************************************************
//*  STEP 6: THIS STEP DOES THE FOLLOWING:
//*         -SQL GENERATION
//*******************************************************************
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//MODEFILE DD   DSN=NALUR.ARYLAS.MODE.R0000695,
// DISP=OLD
//SQLOUT   DD   DSN=NALUR.ARY06244.S123453.M578218.EXECSQL,
//         DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE),
//         DCB=(LRECL=80,BLKSIZE=23440,RECFM=FB)
//SYSOUT   DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*                                              01870000
//SYSUDUMP DD   SYSOUT=*                                              01880000
//WARNINGS DD   SYSOUT=*                                              01890000
//CFILES   DD   SYSOUT=*
//*                                                                   01920000
//*
.........................
.........................
//LGNL0007 EXEC PGM=ARY#UTIL,PARM='EXECSQL,DB8W ',
//            COND=((4,LT),(0,NE,LASR0002))
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
```

```
CONFIG DEFAULT
CONTINUE ON WARNING
//SYSIN    DD   DSN=NALUR.ARY06244.S123453.M578218.EXECSQL,
//             DISP=(OLD,KEEP)
//*
..................
..................
//*
//RCVR0008 EXEC PGM=DSNUTILB,PARM=(DB8W),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *

  RECOVER
    TABLESPACE VOLDB4DB.VOLDB4TS
      LOCALSITE

//RBLD0009 EXEC PGM=DSNUTILB,PARM=(DB8W),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//UTPRINT  DD   SYSOUT=*
//SYSOUT   DD   UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN    DD   *

  REBUILD INDEX
    ( "RETOOLS"."XVOLDB4" )
      SCOPE ALL

/*
..................
..................
```

2.3.3 Recovering one or more partitions of a partitioned table space

This scenario describes the recovery of two partitions (1 and 3) of a four-partitioned table space PARTTS in database PARTDB to current, which simulates an I/O error (less likely) or volume (more likely because multiple partitions are involved). In the timeline shown in Figure 2-44 on page 97, we assume that at least one full image copy exists for the entire partitioned table space (less likely in a real-world scenario, which tends to copy each partition in parallel) or each of the table space partitions to be recovered to current. This environment is a data sharing environment.

The following windows only represent the ones of particular interest (in our opinion) to this scenario. The windows leading up to these windows and those that follow these windows are almost identical to the windows described in 2.3.1, “Recovering multi-table table space” on page 56. We show the partial contents of the generated JCL for a given recovery plan, but not the execution job results.

Figure 2-45 on page 97 shows the list of partitions 1 and 3 selected for recovery by navigating through the table spaces in the navigation pane. Figure 2-46 on page 98 shows the recovery point for these objects (to current). Figure 2-47 on page 98 shows the option to create an image copy after recovery. Figure 2-48 on page 99 shows that four recovery plans were generated for the requested recovery action, and the expanded steps included in the recovery plan, Plan 4. Because Plan 4 contains a log analysis step, a warning is issued to this effect, as shown in Figure 2-49 on page 99. A portion of the generated JCL for Plan 4 is shown in Example 2-10 on page 100.

The selected recovery plan, Plan 4, has steps to restore the various table spaces to the latest partition level image copies using the RECOVER statement followed by a rebuilding of all indexes, followed by the generation of redo SQL from the DB2 log using log analysis. The start log record sequence number (LRSN) for log analysis corresponds to that of the image copy selected (LRSN BF642E978754) and the end LRSN (FFFFFFFFFFFF) corresponds to the end of the log. The final steps are to apply the generated SQL and take image copies of the partitions, as specified in Figure 2-47 on page 98.

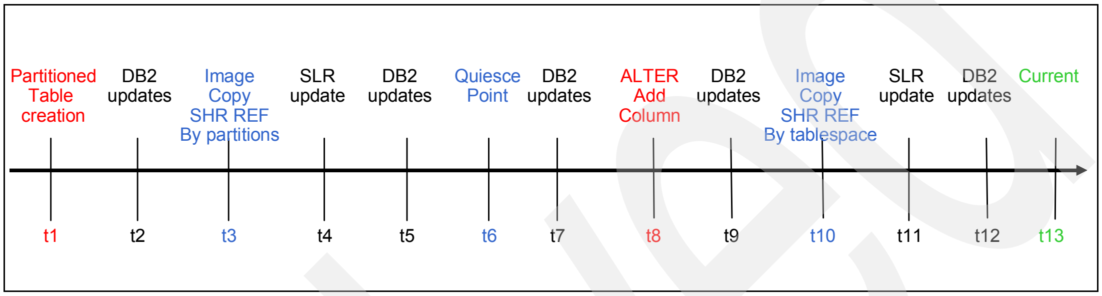

Figure 2-44 Recovering two partitions of a partitioned table space scenario timeline

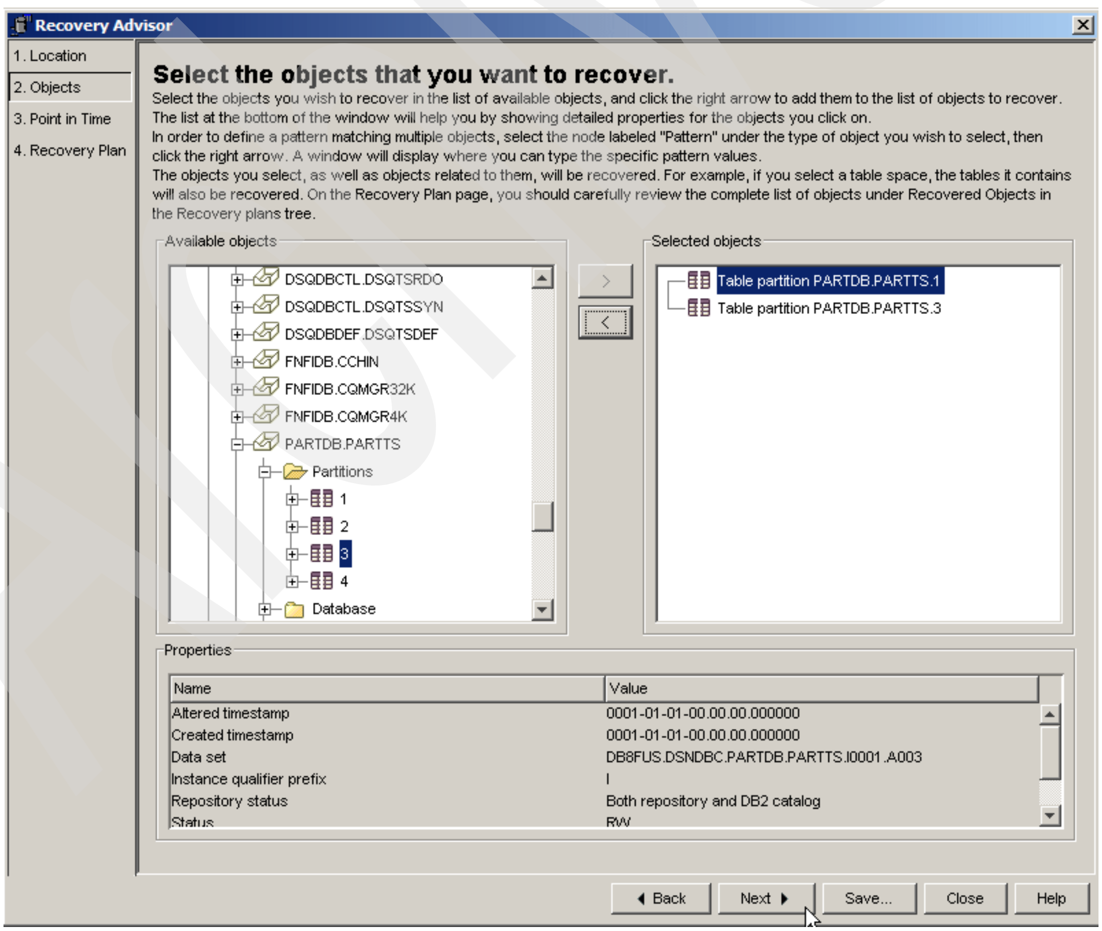

Figure 2-45 Selecting partitions 1 and 3 of PARTTS to recover

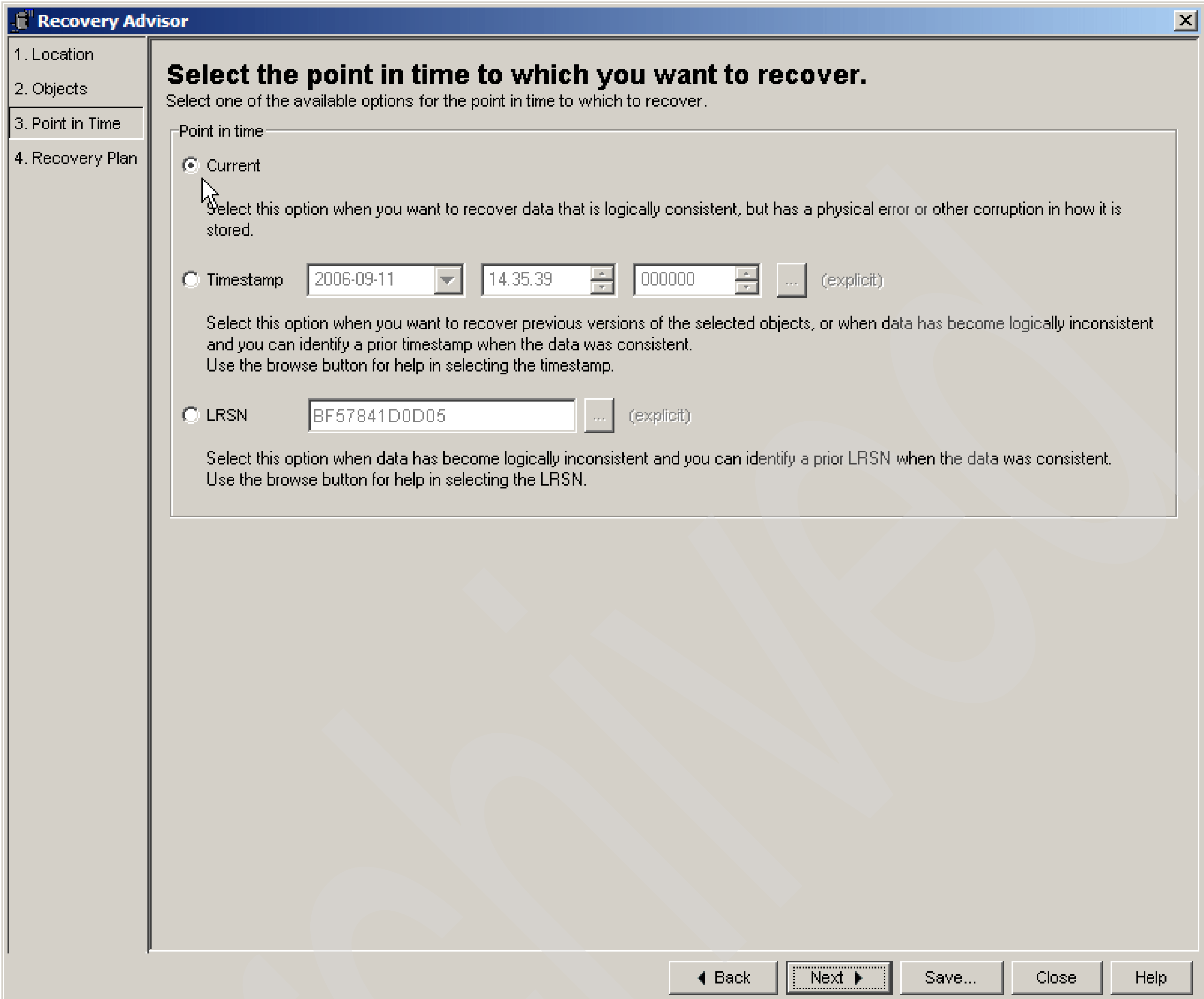

Figure 2-46 Recovering partitions 1 and 3 to current

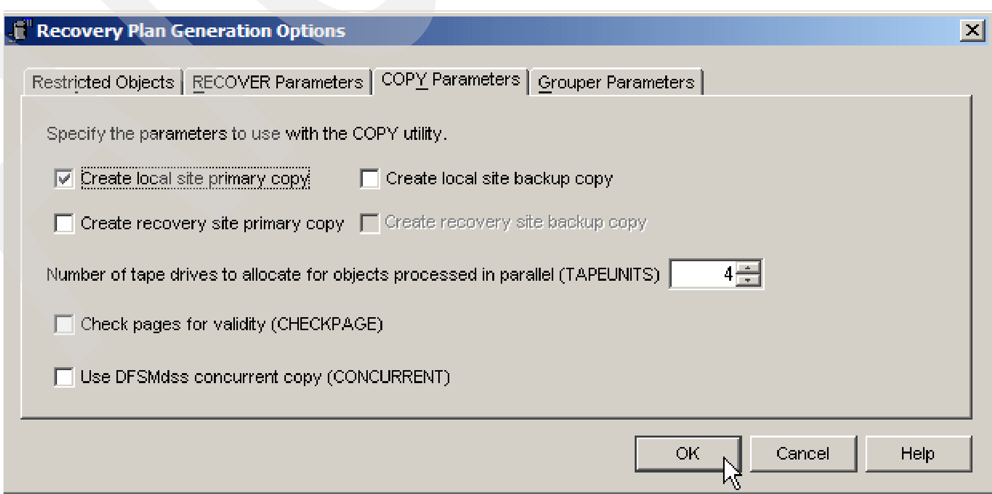

Figure 2-47 Selecting creation of image copies of partitions

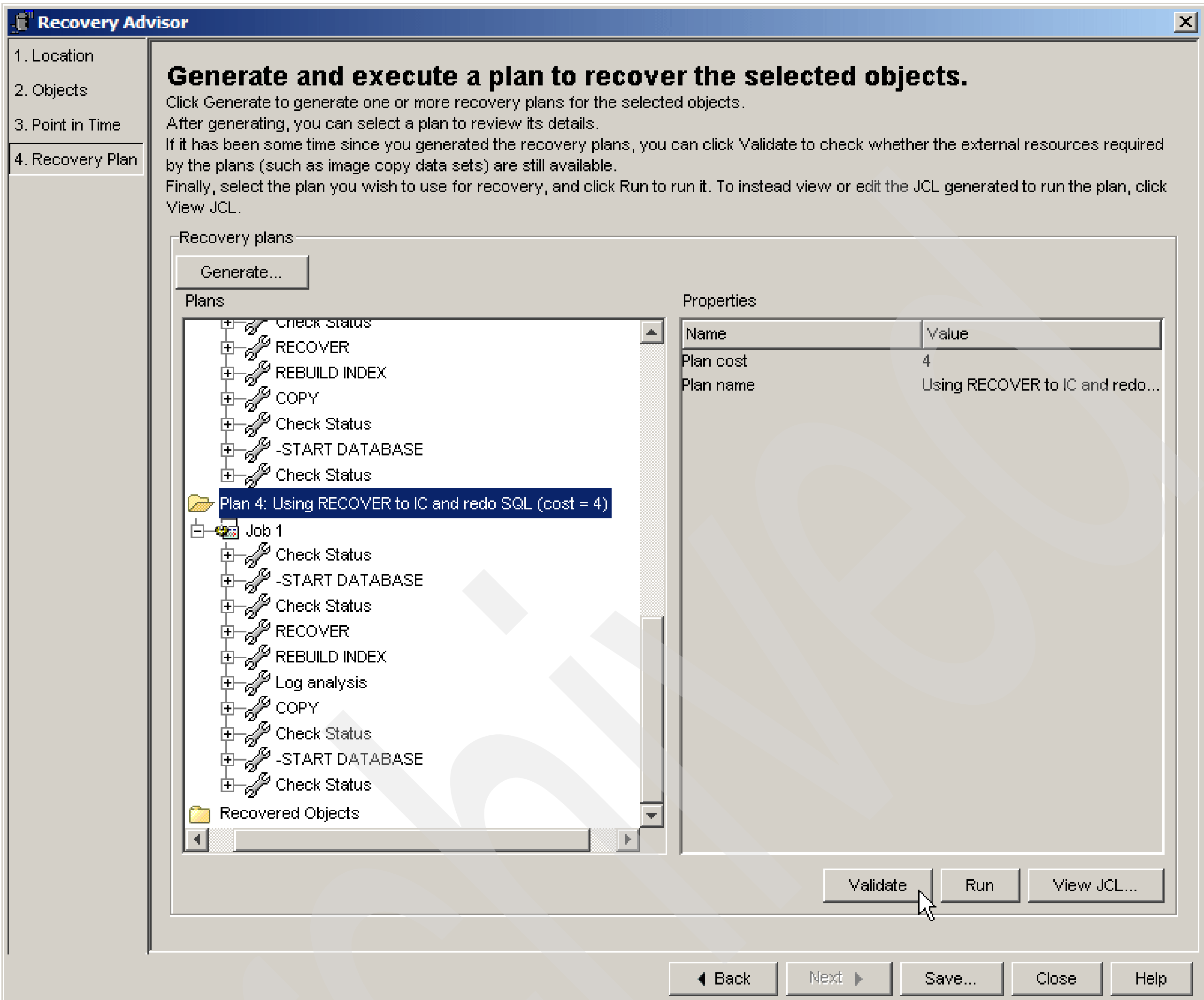

Figure 2-48 Generated recovery plans

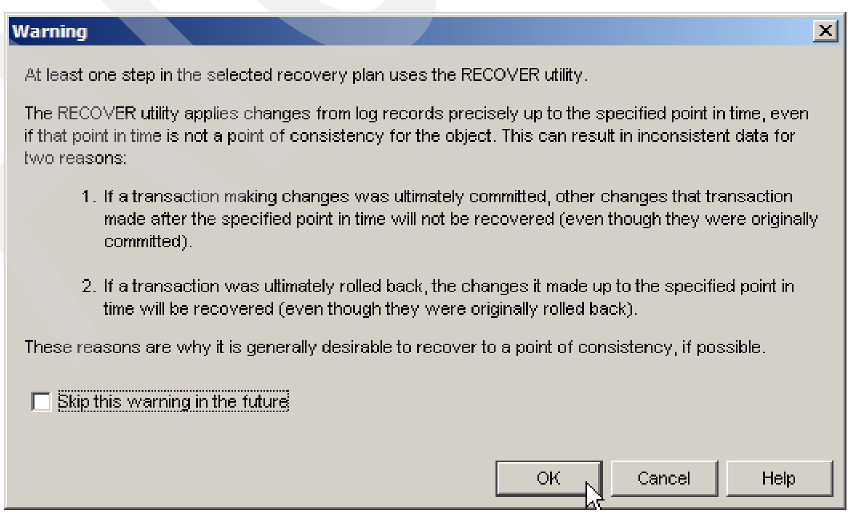

Figure 2-49 Warning message about log analysis being started for the selected recovery plan

Example 2-10 Partial contents of the generated JCL for Plan 1

```
//F8PRECPT JOB ,'DB2 RECOVERY EXPERT',MSGCLASS=H,
//            REGION=0M,NOTIFY=&SYSUID
//*
..................
..................
//RCVR0005 EXEC PGM=DSNUTILB,PARM=(D8F1),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *

  RECOVER
    TABLESPACE PARTDB.PARTTS     DSNUM 1
      TOCOPY REIC.D8FG.PARTDB.PARTTS.FULL.P1.G0003V00
      LOCALSITE

  RECOVER
    TABLESPACE PARTDB.PARTTS     DSNUM 3
      TOCOPY REIC.D8FG.PARTDB.PARTTS.FULL.P3.G0003V00
      LOCALSITE

/*
//*
//RBLD0006 EXEC PGM=DSNUTILB,PARM=(D8F1),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//UTPRINT  DD   SYSOUT=*
//SYSOUT   DD   UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN    DD   *

  REBUILD INDEX
    ( "RETOOLS"."XPART1"
    , "RETOOLS"."XPART2" )
      SCOPE ALL

/*
.....................
.....................
//*******************************************************************
//* STEP 2: READ THE DB2 LOG TO GENERATE THE GENERAL REPORT         *
//*******************************************************************
//LASR0002 EXEC PGM=ARYGEN1,REGION=0M,COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//MODEFILE DD   DSN=RETOOLS.ARYLAS.MODE.R0000164,
// DISP=OLD
//SYSOUT   DD   SYSOUT=*
//CFILES   DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//GENRPT   DD   SYSOUT=*
//EXTREP   DD   SYSOUT=*
```

```
//SUMREPT  DD   SYSOUT=*
//XDREPT   DD   SYSOUT=*
//QTRPT    DD   SYSOUT=*
//WARNINGS DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//DATAIN   DD   *
SSID           =D8F1
START LRSN     =BF642E978754
END LRSN       =FFFFFFFFFFFF
/*
//FILTERS  DD DISP=SHR,
// DSN=RETOOLS.ARYLAS.FILTERS.R0000164
//*********************************
//* PARAMETERS FOR JOB SHOWN BELOW *
//*********************************
//* SSID           =D8F1
//* START LRSN     =BF642E978754
//* END LRSN       =FFFFFFFFFFFF
//* /*
//* //FILTERS  DD DISP=SHR,
//* // DSN=RETOOLS.ARYLAS.FILTERS.R0000164
//*
................
................
//LASR0006 EXEC PGM=ARYDTL4,REGION=0M,COND=(4,LT)
//*****************************************************************
//*  STEP 6: THIS STEP DOES THE FOLLOWING:
//*         -SQL GENERATION
//*****************************************************************
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//MODEFILE DD   DSN=RETOOLS.ARYLAS.MODE.R0000164,
// DISP=OLD
//SQLOUT   DD   DSN=RETOOLS.ARY06254.S153918.M494001.EXECSQL,
//         DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE),
//         DCB=(LRECL=80,BLKSIZE=23440,RECFM=FB)
//SYSOUT   DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*                                          01870000
//SYSUDUMP DD   SYSOUT=*                                          01880000
//WARNINGS DD   SYSOUT=*                                          01890000
//CFILES   DD   SYSOUT=*
//*                                                               01920000
//*
.................
.................
//LGNL0007 EXEC PGM=ARY#UTIL,PARM='EXECSQL,D8F1 ',
//             COND=((4,LT),(0,NE,LASR0002))
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//SYSPRINT DD   SYSOUT=*
```

```
//CONTROLS DD   *
CONFIG DEFAULT
CONTINUE ON WARNING
//SYSIN    DD   DSN=RETOOLS.ARY06254.S153918.M494001.EXECSQL,
//             DISP=(OLD,KEEP)
//*
...............
...............
//COPY0008 EXEC PGM=DSNUTILB,PARM=(D8F1),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//CPY00010 DD   DSN=RETOOLS.ARY06254.S153919.M245783.COPY,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(NEW,CATLG,DELETE)
//CPY00020 DD   DSN=RETOOLS.ARY06254.S153919.M250344.COPY,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(NEW,CATLG,DELETE)
//SYSIN    DD   *

  COPY
    TABLESPACE PARTDB.PARTTS     DSNUM 1
      FULL YES
      COPYDDN(CPY00010)
    TABLESPACE PARTDB.PARTTS     DSNUM 3
      FULL YES
      COPYDDN(CPY00020)
................
................
```

2.4 Recovering objects to a point-in-time

The triggering event for recovering an object or set of objects to a point-in-time is typically application errors or operator errors, and sometimes a site failure that requires disaster recovery[3]. When a single object such as a table space is recovered to a PIT, its dependent objects such as indexes or other table spaces in the same referential set *must* also be recovered because they are affected by the recovery action.

Attention: Recovery Expert for z/OS automatically identifies all table spaces in the referential set when you identify just a single member of the referential set as the target of recovery, and all dependent objects such as indexes. It then generates appropriate control statements to recover all these objects to the PIT. It also automatically generates CHECK DATA control statements for table spaces that are part of a referential set as appropriate.

As far as possible, we used a common timeline showing a sequence of events in all the scenarios. These events are not the most optimal of operations, but we have selected them to demonstrate the recovery semantics of Recovery Expert for z/OS.

[3] Recovery Expert for z/OS does not support disaster recovery.

The recover to point-in-time scenarios described in this section are:

- Recovering multi-table table space to before current Data Definition Language (DDL) version
- Recovering multi-table table space to RBA at current DDL version
- Recovering a partition to image copy (IC) before current DDL version
- Recovering a partition to LRSN before current DDL version
- Recovering a partitioned table space to IC at current DDL version
- Recovering a database to a timestamp at current DDL version
- Recovering a table space in referential set to IC at current DDL version
- Recovering a table space in referential set to quiet time (QT) at current DDL version

2.4.1 Recovering multi-table table space to IC before current DDL version

This scenario describes the recovery of a multi-table table space to an image copy that was taken before a DDL change that occurred on one of the tables in the multi-table table space, that is, before the current DDL version of the object. The timeline shown in Figure 2-50 on page 104 is assumed for this scenario. This environment is a non-data sharing environment.

The following windows only represent the ones of particular interest (in our opinion) to this scenario. The windows leading up to these windows and those that follow these windows are almost identical to the windows described in 2.3.1, “Recovering multi-table table space” on page 56. We show the partial contents of the generated JCL for a given recovery plan, but not the execution job results.

Figure 2-51 on page 104 shows the multi-table table space MTBTS6TS selected for recovery by using the pattern mechanism. Figure 2-53 on page 105 shows the available point-of-consistency entries[4] in SYSIBM.SYSCOPY (recovery history events) from which we selected an image copy entry (share level reference) corresponding to time t3 in Figure 2-50 on page 104; not all window interactions are shown here. Figure 2-54 on page 106 shows this image copy selection translated to a timestamp corresponding to the image copy entry that is selected in Figure 2-53 on page 105. Figure 2-55 on page 107 shows a single recovery plan being generated for the requested recovery action, the recovered objects, and the expanded steps included in the recovery plan, Plan 1. A portion of the generated JCL for Plan 1 is shown in Example 2-11 on page 107.

The selected recovery plan, Plan 1, has multiple steps that include the dropping of all the tables in the multi-table space, followed by a DDL recreation of these tables (from information stored in the schema level repository (SLR)) and its dependencies (such as indexes and views), according to the rules described in “Algorithm for restoring object dependencies and data” on page 22. This is followed by a restore of the target image copy using DSN1COPY, an object identifier (OBID) translate to reflect the OBID of the new object definitions, and a rebuilding of the indexes.

> **Attention:** As discussed in 1.3.3, “Recovery semantics” on page 16, this Recovery Expert semantics differs from that of the base DB2 for z/OS engine.

[4] The “Show events that are not point-of-consistency” check box is not selected in Figure 2-53 on page 105.

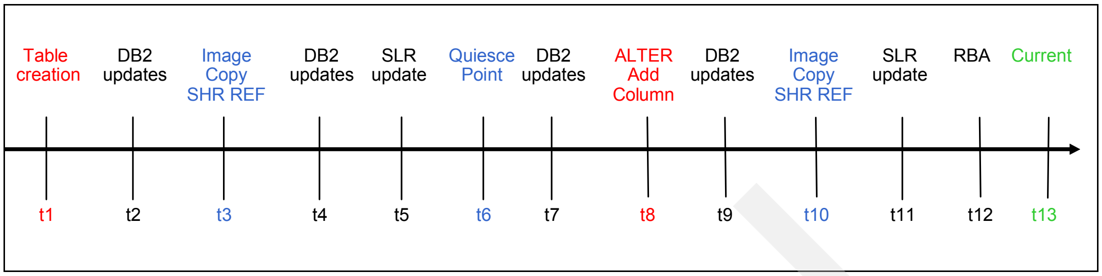

Figure 2-50 Recovering multi-table table space MTBTS6TS to PIT scenario timeline

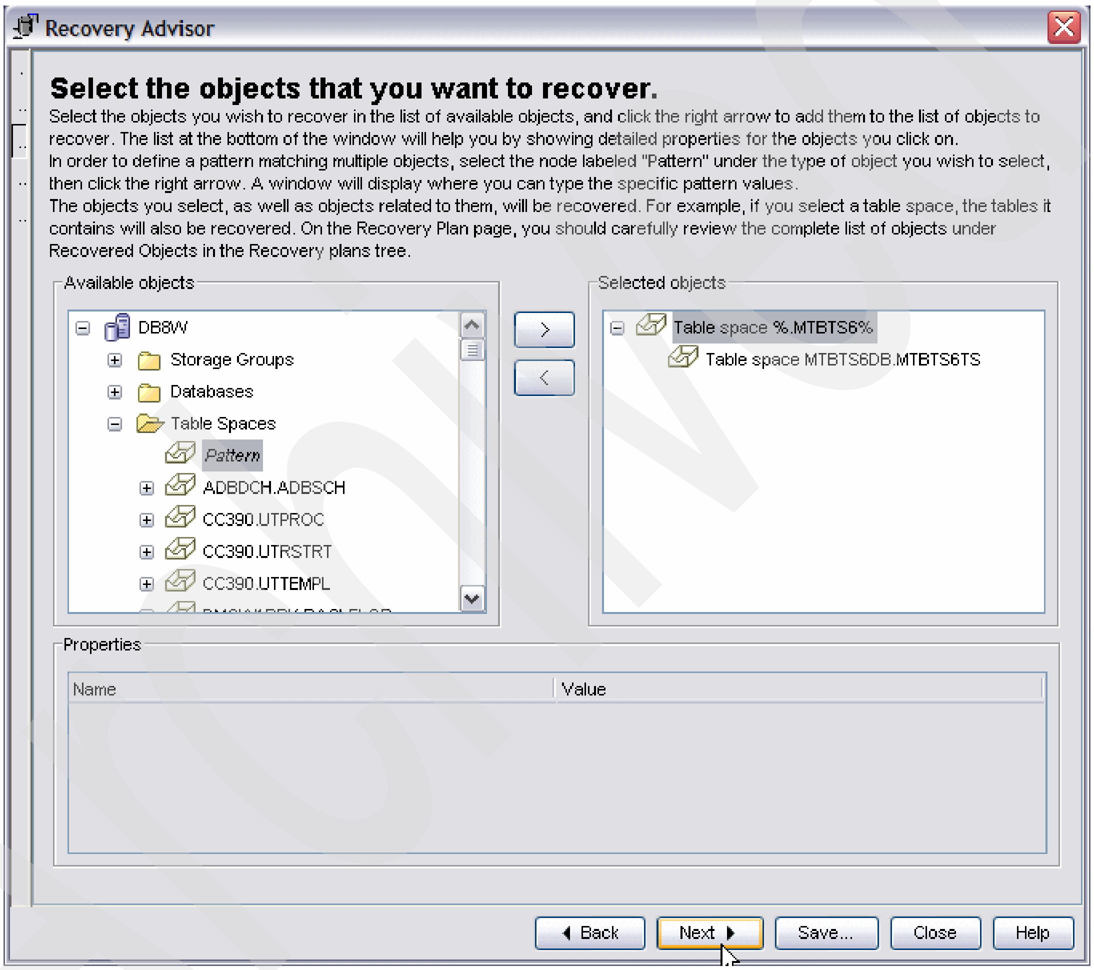

Figure 2-51 Selecting the object to recover

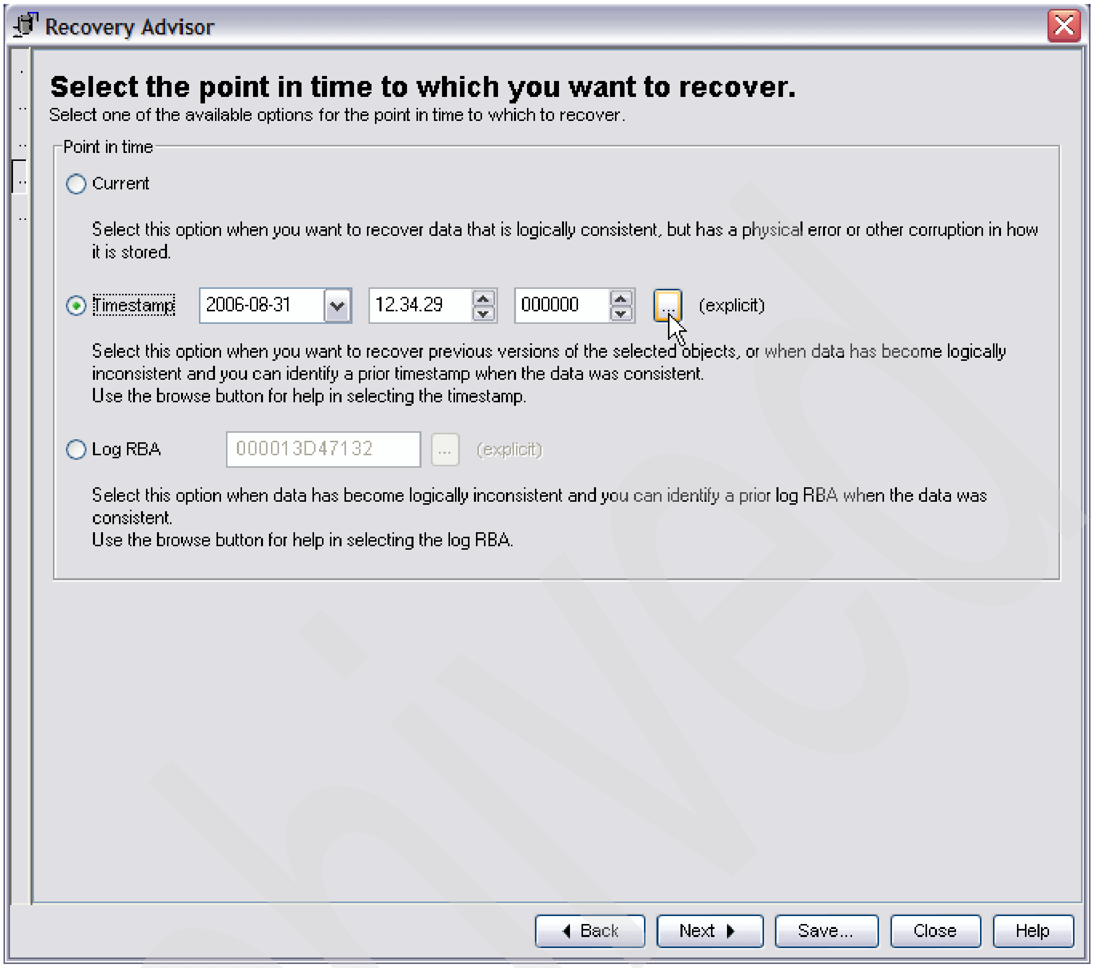

Figure 2-52 Selecting the point to recover to (1/3)

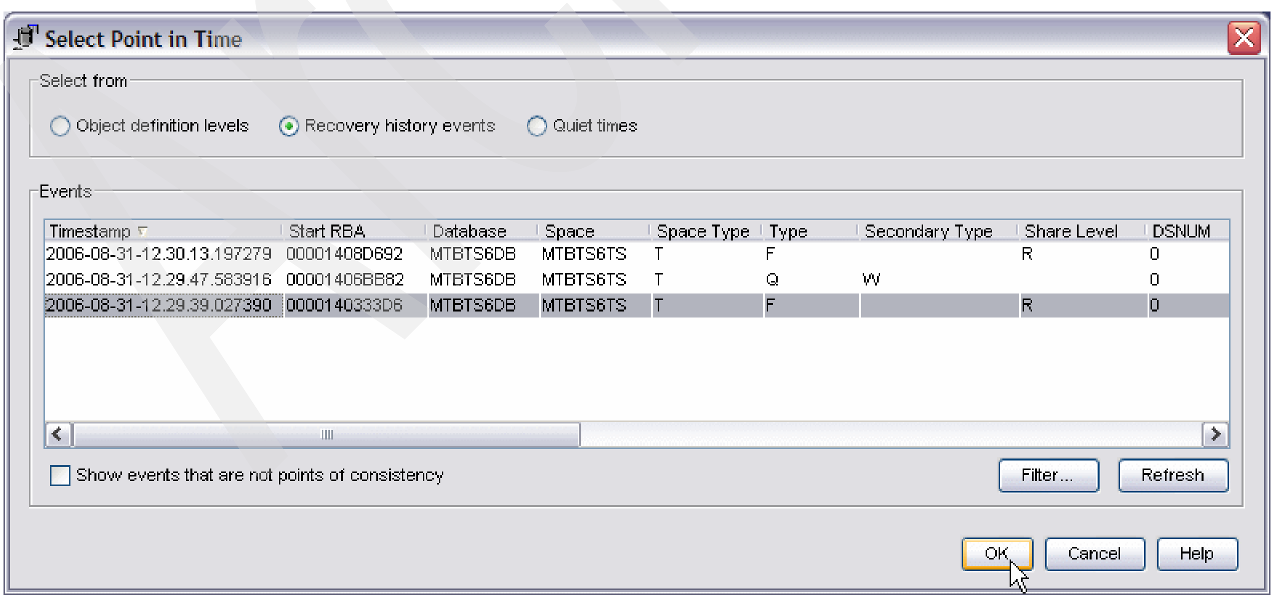

Figure 2-53 Selecting the point to recover to (2/3)

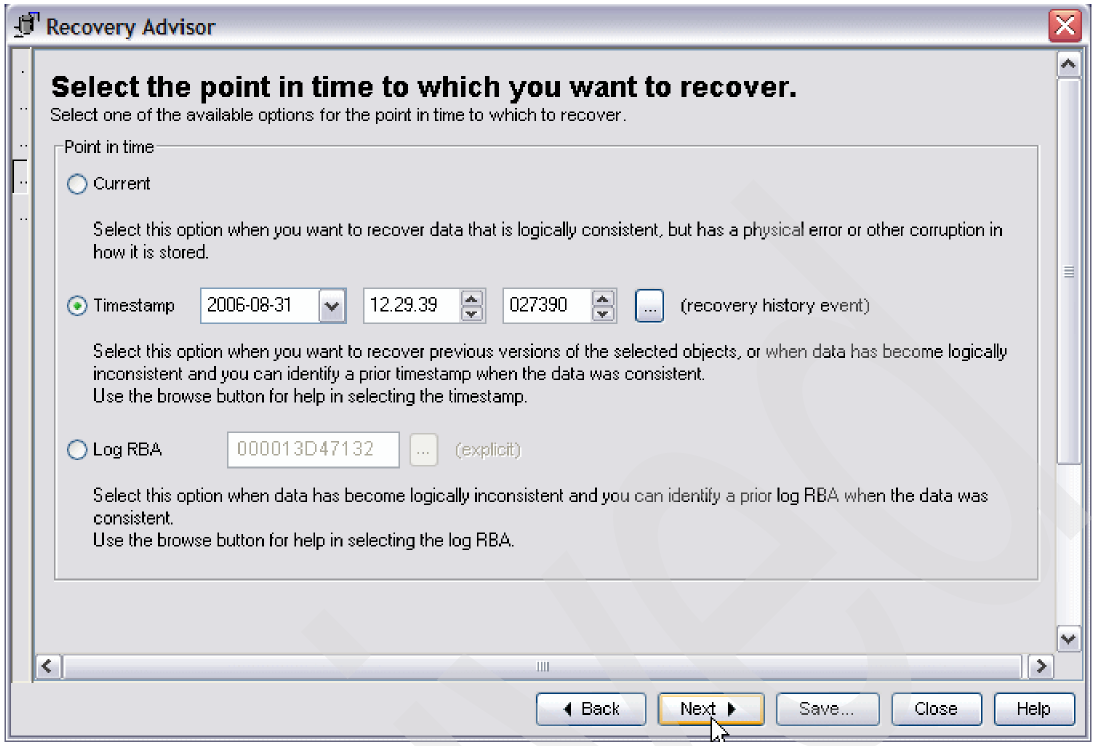

Figure 2-54 Selecting the point to recover to (3/3)

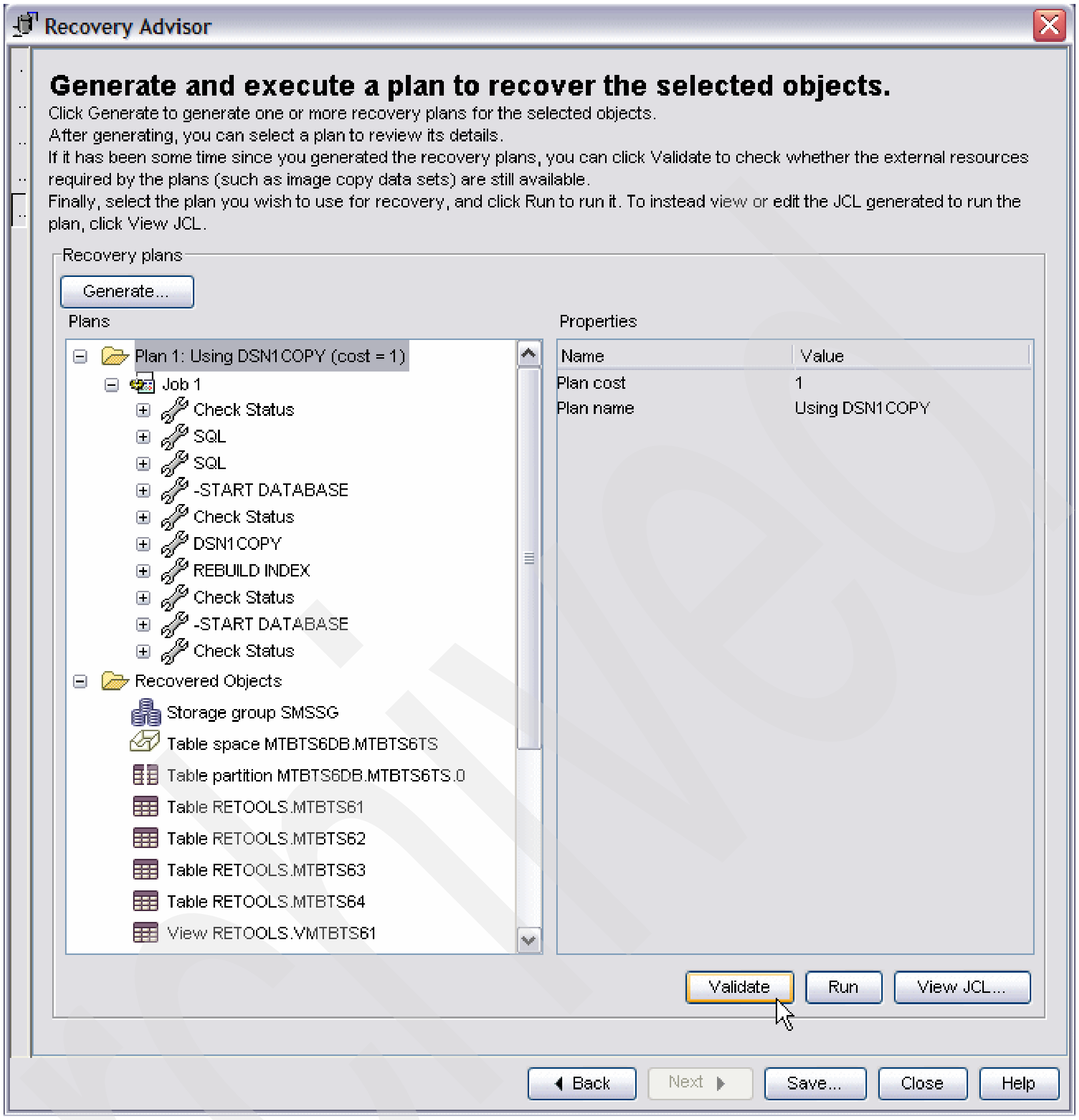

Figure 2-55 Generated recovery plans

Example 2-11 Partial contents of generated JCL for Plan 1

```
//W8RCVTS JOB (999,POK),'RETEST',CLASS=A,MSGCLASS=H,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
//*
.............
.............
//SQDR0002 EXEC PGM=ARY#UTIL,PARM='EXECSQL,DB8W ',COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
CONFIG DEFAULT
//SYSIN    DD   *
--#SET CONTINUE ON WARNING
```

```
  DROP VIEW "RETOOLS"."VMTBTS62";
  COMMIT;

  DROP TABLE "RETOOLS"."MTBTS62";
  COMMIT;

  DROP VIEW "RETOOLS"."VMTBTS61";
  COMMIT;

  DROP TABLE "RETOOLS"."MTBTS61";
  COMMIT;
/*
//*
//SQCR0003 EXEC PGM=ARY#UTIL,PARM='EXECSQL,DB8W ',COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
CONFIG DEFAULT
//SYSIN    DD   *
--#SET CONTINUE ON WARNING

  SET CURRENT SQLID = 'ALICIAW';
  CREATE TABLE "RETOOLS"."MTBTS61"
    (DEPTNO              CHAR(3) NOT NULL FOR SBCS DATA
    ,DEPTNAME            VARCHAR(36) NOT NULL FOR SBCS DATA
    ,MGRNO               CHAR(6) FOR SBCS DATA WITH DEFAULT NULL
    ,ADMRDEPT            CHAR(3) NOT NULL FOR SBCS DATA
    ,LOCATION            CHAR(16) FOR SBCS DATA WITH DEFAULT NULL
    ,PRIMARY KEY ( DEPTNO))
    IN MTBTS6DB.MTBTS6TS
    AUDIT NONE
    CCSID EBCDIC
;
  COMMIT;

  SET CURRENT SQLID = 'RETOOLS';
  SET CURRENT PATH = "SYSIBM","SYSFUN","SYSPROC","RETOOLS";
--#SET TERMINATOR #
    CREATE VIEW RETOOLS.VMTBTS61
    AS SELECT ALL DEPTNO , DEPTNAME, MGRNO , ADMRDEPT
    FROM RETOOLS.MTBTS61#
--#SET TERMINATOR ;
  COMMIT;

  SET CURRENT SQLID = 'ALICIAW';
  CREATE UNIQUE INDEX "RETOOLS"."XMTBTS61"
    ON "RETOOLS"."MTBTS61"
    (DEPTNO ASC)
    USING STOGROUP "SMSSG"
      PRIQTY 12
      SECQTY 4
      ERASE NO
```

```
    PCTFREE 10
    PIECESIZE 2 G
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
;
  COMMIT;

  SET CURRENT SQLID = 'ALICIAW';
  CREATE TABLE "RETOOLS"."MTBTS62"
    (DEPTNO              CHAR(3) NOT NULL FOR SBCS DATA
    ,DEPTNAME            VARCHAR(36) NOT NULL FOR SBCS DATA
    ,MGRNO               CHAR(6) FOR SBCS DATA WITH DEFAULT NULL
    ,ADMRDEPT            CHAR(3) NOT NULL FOR SBCS DATA
    ,LOCATION            CHAR(16) FOR SBCS DATA WITH DEFAULT NULL
    ,PRIMARY KEY ( DEPTNO))
    IN MTBTS6DB.MTBTS6TS
    AUDIT NONE
    CCSID EBCDIC
;
  COMMIT;

  SET CURRENT SQLID = 'RETOOLS';
  SET CURRENT PATH = "SYSIBM","SYSFUN","SYSPROC","RETOOLS";
--#SET TERMINATOR #
    CREATE VIEW RETOOLS.VMTBTS62
    AS SELECT ALL DEPTNO , DEPTNAME, MGRNO , ADMRDEPT
    FROM RETOOLS.MTBTS62#
--#SET TERMINATOR ;
  COMMIT;

  SET CURRENT SQLID = 'ALICIAW';
  CREATE UNIQUE INDEX "RETOOLS"."XMTBTS62"
    ON "RETOOLS"."MTBTS62"
    (DEPTNO ASC)
    USING STOGROUP "SMSSG"
      PRIQTY 12
      SECQTY 4
      ERASE NO
    PCTFREE 10
    PIECESIZE 2 G
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
;
  COMMIT;
/*
//*
...........................
...........................
//DCPX0006 EXEC PGM=ARY#UTIL,PARM='UTGENA,DB8W ',COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
```

```
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
CONFIG DEFAULT
//SYSXLAT  DD   DSN=NALUR.ARY06243.S124247.M104553.SYSXLAT,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(NEW,CATLG,DELETE)
//SYSIN    DD   *
TABLESPACE
0288,MTBTS6DB
0002,MTBTS6TS
0003,"RETOOLS"."MTBTS61"
0006,"RETOOLS"."MTBTS62"
0009,"RETOOLS"."MTBTS63"
0012,"RETOOLS"."MTBTS64"
/*
//*
//DCPY0007 EXEC PGM=DSN1COPY,COND=(4,LT),
// PARM='CHECK,PAGESIZE(4K),FULLCOPY,OBIDXLAT,RESET'
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSUT1   DD DISP=SHR,DSN=REIC.DB8W.MTBTS6DB.MTBTS6TS.FULL.G0001V00
//SYSUT2   DD DISP=SHR,DSN=DB8WUS.DSNDBC.MTBTS6DB.MTBTS6TS.I0001.A001
//SYSPRINT DD   SYSOUT=*
//SYSXLAT  DD DISP=SHR,DSN=NALUR.ARY06243.S124247.M104553.SYSXLAT
//*
//RBLD0008 EXEC PGM=DSNUTILB,PARM=(DB8W),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//UTPRINT  DD   SYSOUT=*
//SYSOUT   DD   UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN    DD   *

  REBUILD INDEX
    ( "RETOOLS"."XMTBTS61"
    , "RETOOLS"."XMTBTS62"
    , "RETOOLS"."XMTBTS63"
    , "RETOOLS"."XMTBTS64" )
      SCOPE ALL

/*
................
................
```

2.4.2 Recovering multi-table table space to RBA at current DDL version

This scenario describes the recovery of a multi-table table space to an RBA corresponding to a time after the last DDL change that occurred on one of the tables in the multi-table table space, that is, at the current DDL version of the object. The timeline shown in Figure 2-50 on page 104 is assumed for this scenario. This environment is a non-data sharing environment.

The following windows only represent the ones of particular interest (in our opinion) to this scenario. The windows leading up to these windows and those that follow these windows are almost identical to the windows described in 2.3.1, “Recovering multi-table table space” on page 56. We show the partial contents of the generated JCL for a given recovery plan, but not the execution job results.

Figure 2-56 shows the multi-table table space selected for recovery by expanding the table spaces folder in the navigation pane. Figure 2-57 on page 112 shows a user-specified RBA as the point-in-time to recover to (we assume that the user has determined this RBA to be the appropriate point to recover to using some external mechanism such as a shutdown point of the DB2 subsystem). This RBA corresponds to time t12 in Figure 2-50 on page 104. Figure 2-58 on page 113 shows five potential recovery plans being generated for the requested recovery action, and the expanded steps included in the recovery plan, Plan 2. Because the selected Plan 2 has a log analysis step, a warning message is issued, as shown in Figure 2-59 on page 113. A portion of the generated JCL for Plan 2 is shown in Example 2-12 on page 114.

The selected recovery plan, Plan 1, has multiple steps that include the restore of the multi-table table space from a full image copy followed by a rebuilding of the indexes. The next step is the generation of redo SQL from the DB2 log using log analysis for table space up to the RBA specified. The log analysis is bounded by the start RBA (00001420EAEE) corresponding to the image copy chosen by DSN1COPY to restore from, and the end RBA (00001420FFFF) corresponding to the user-specified point to recover to in Figure 2-57 on page 112. This is followed by the execution of this generated SQL to complete the recovery action requested.

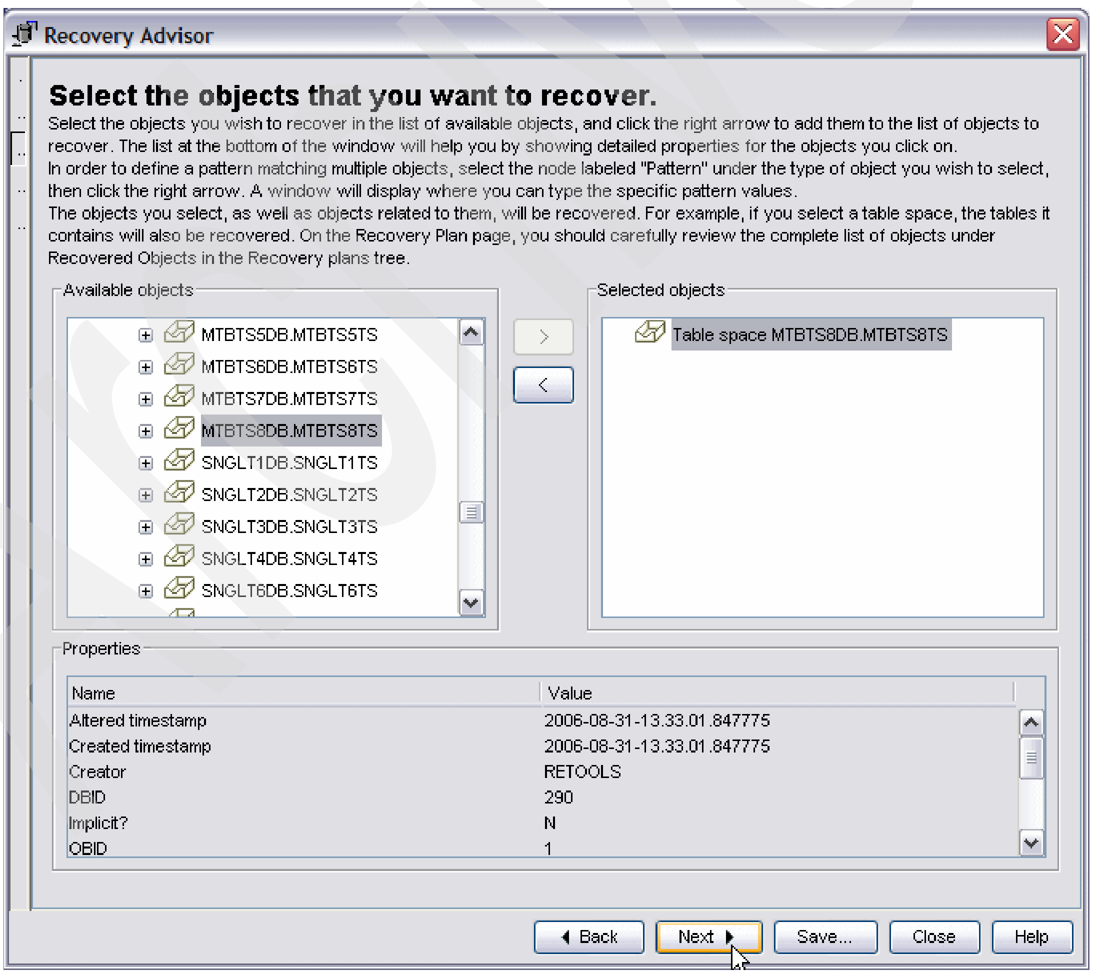

Figure 2-56 Selecting the object to recover

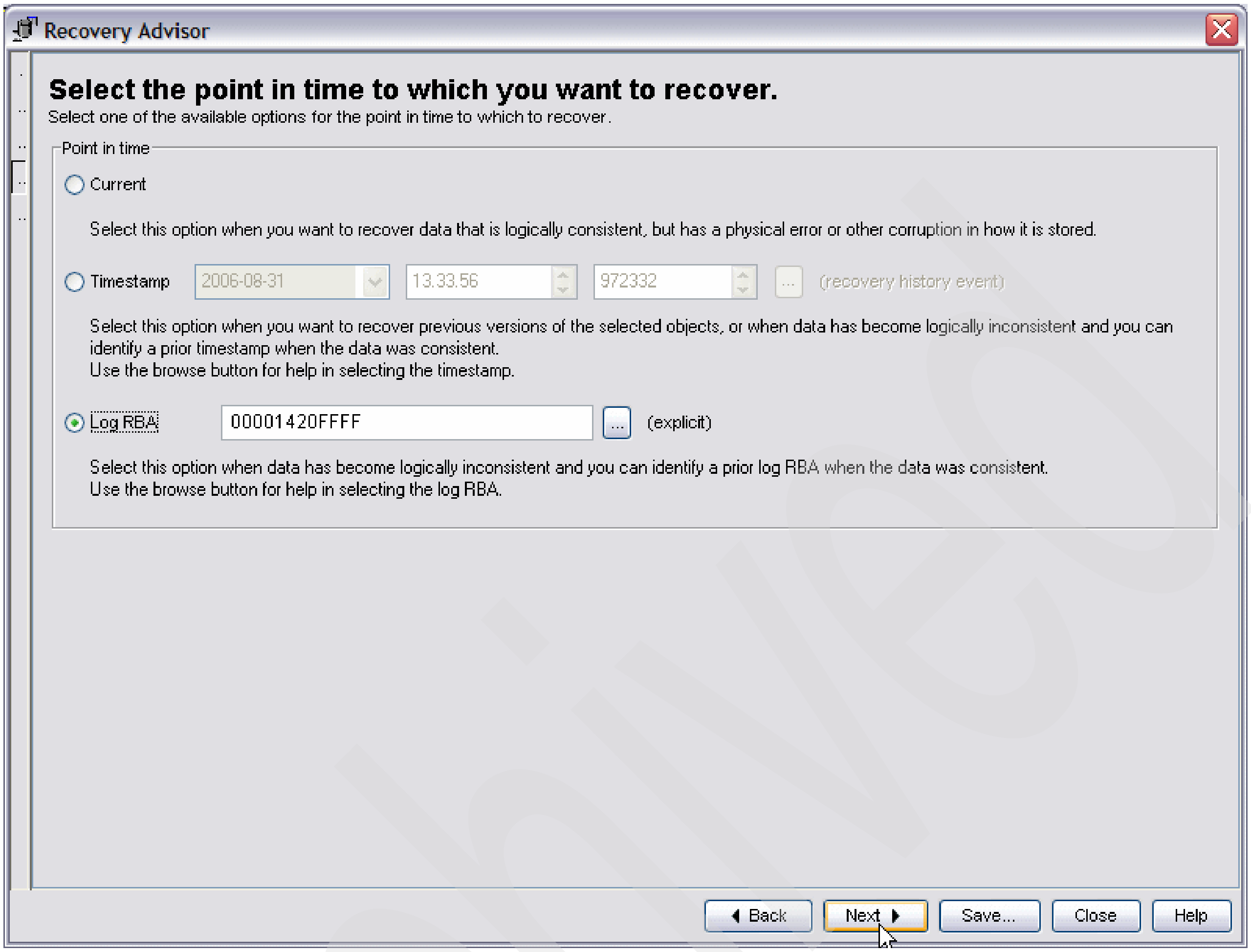

Figure 2-57 Specifying the RBA to recover to

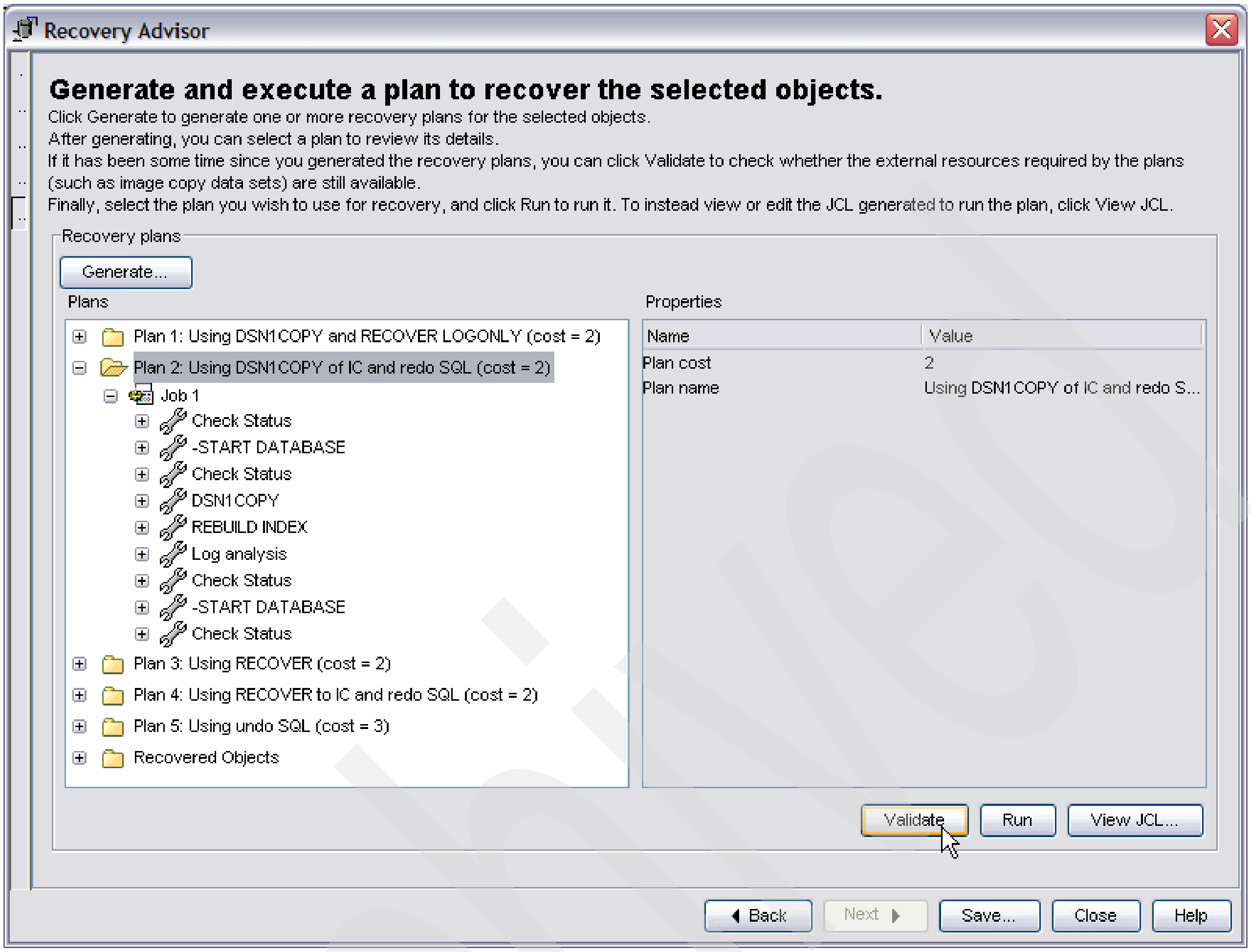

Figure 2-58 Generated recovery plans

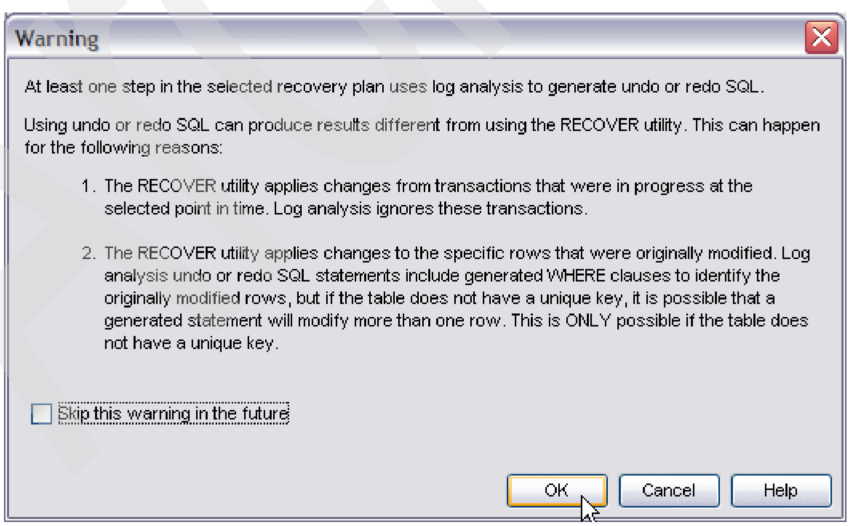

Figure 2-59 Warning message regarding log analysis being used in the selected recovery plan

Example 2-12 Partial contents of the generated JCL for Plan 2

```
//W8RBATS JOB (999,POK),'RETEST',CLASS=A,MSGCLASS=H,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
.....................
.....................
//DCPY0004 EXEC PGM=DSN1COPY,COND=(4,LT),
// PARM='CHECK,PAGESIZE(4K),FULLCOPY,RESET'
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSUT1   DD DISP=SHR,DSN=REIC.DB8W.MTBTS8DB.MTBTS8TS.FULL.G0002V00
//SYSUT2   DD DISP=SHR,DSN=DB8WUS.DSNDBC.MTBTS8DB.MTBTS8TS.I0001.A001
//SYSPRINT DD   SYSOUT=*
//*
//RBLD0005 EXEC PGM=DSNUTILB,PARM=(DB8W),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//UTPRINT  DD   SYSOUT=*
//SYSOUT   DD   UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN    DD   *

  REBUILD INDEX
    ( "RETOOLS"."XMTBTS81"
    , "RETOOLS"."XMTBTS82"
    , "RETOOLS"."XMTBTS83"
    , "RETOOLS"."XMTBTS84" )
      SCOPE ALL

....................
....................
//*******************************************************************
//* STEP 2: READ THE DB2 LOG TO GENERATE THE GENERAL REPORT          *
//*******************************************************************
//LASR0002 EXEC PGM=ARYGEN1,REGION=0M,COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//MODEFILE DD   DSN=NALUR.ARYLAS.MODE.R0000280,
// DISP=OLD
//SYSOUT   DD   SYSOUT=*
//CFILES   DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//GENRPT   DD   SYSOUT=*
//EXTREP   DD   SYSOUT=*
//SUMREPT  DD   SYSOUT=*
//XDREPT   DD   SYSOUT=*
//QTRPT    DD   SYSOUT=*
//WARNINGS DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//DATAIN   DD   *
SSID           =DB8W
START RBA      =00001420EAEE
END RBA        =00001420FFFF
```

```
/*
//FILTERS  DD *
GENERAL FILTERS...........
SHOW UPDATES   =Y
SHOW INSERTS   =X
SHOW DELETES   =Y
SHOW ROLLBACKS =N
CATALOG DATA   =N
SHOW UNCOMMITS =N
OBJECT FILTERS-BY NAME....
DBNAME     =MTBTS8DB
TSNAME     =MTBTS8TS
PARTITION  =0000
SRBA       =00001420EAEE
ERBA       =00001420FFFF
INC/EXC    =I
/*
//*********************************
//* PARAMETERS FOR JOB SHOWN BELOW *
//*********************************
//* SSID           =DB8W
//* START RBA      =00001420EAEE
//* END RBA        =00001420FFFF
//* /*
//* //FILTERS  DD *
//* GENERAL FILTERS...........
//* SHOW UPDATES   =Y
//* SHOW INSERTS   =X
//* SHOW DELETES   =Y
//* SHOW ROLLBACKS =N
//* CATALOG DATA   =N
//* SHOW UNCOMMITS =N
//* OBJECT FILTERS-BY NAME....
//* DBNAME     =MTBTS8DB
//* TSNAME     =MTBTS8TS
//* PARTITION  =0000
//* SRBA       =00001420EAEE
//* ERBA       =00001420FFFF
//* INC/EXC    =I
//*
...............
...............
//LASR0006 EXEC PGM=ARYDTL4,REGION=0M,COND=(4,LT)
//*****************************************************************
//*  STEP 6: THIS STEP DOES THE FOLLOWING:
//*         -SQL GENERATION
//*****************************************************************
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//MODEFILE DD   DSN=NALUR.ARYLAS.MODE.R0000280,
// DISP=OLD
//SQLOUT   DD   DSN=NALUR.ARY06243.S143417.M651292.EXECSQL,
//         DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE),
```

```
//         DCB=(LRECL=80,BLKSIZE=23440,RECFM=FB)
//SYSOUT   DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*                                                01870000
//SYSUDUMP DD   SYSOUT=*                                                01880000
//WARNINGS DD   SYSOUT=*                                                01890000
//CFILES   DD   SYSOUT=*
//*                                                                     01920000
............
............
//LGNL0006 EXEC PGM=ARY#UTIL,PARM='EXECSQL,DB8W ',
//             COND=((4,LT),(0,NE,LASR0002))
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
CONFIG DEFAULT
CONTINUE ON WARNING
//SYSIN    DD   DSN=NALUR.ARY06243.S143417.M651292.EXECSQL,
//             DISP=(OLD,KEEP)
//*
........................
........................
```

2.4.3 Recovering a partition to IC before current DDL version

This scenario describes the recovery of a partition to an image copy that was taken before a DDL change that occurred on the partitioned table, that is, before the current DDL version of the object. The timeline shown in Figure 2-60 on page 117 is assumed for this scenario. This environment is a data sharing environment.

The following windows only represent the ones of particular interest (in our opinion) to this scenario. The windows leading up to these windows and those that follow these windows are almost identical to the windows described in 2.3.1, “Recovering multi-table table space” on page 56. We show the partial contents of the generated JCL for a given recovery plan, but not the execution job results.

Figure 2-61 on page 117 shows partition 1 of a partitioned table space PARTTS selected for recovery in the navigation pane. Figure 2-62 on page 118 through Figure 2-64 on page 120 show the selection of an image copy (share level reference) corresponding to time t3 in Figure 2-60 on page 117 from a list of available point-of-consistency entries in SYSIBM.SYSCOPY (recovery history events); not all window interactions are shown here. Figure 2-64 on page 120 shows this image copy selection translated to a timestamp corresponding to the image copy entry that is selected in Figure 2-63 on page 119. Figure 2-65 on page 120 shows COPY parameters for recovery plan generation. Figure 2-66 on page 121 shows five recovery plans being generated for the requested recovery action, the recovered objects, and the expanded steps included in the recovery plan, Plan 1. A portion of the generated JCL for Plan 1 is shown in Example 2-13 on page 122.

The selected recovery plan, Plan 1, has multiple steps that include the RECOVER statement to the full image copy, followed by a rebuilding of the indexes that completes the recovery action requested. It is followed by an image copy of the partition as per the image copy requested in Figure 2-65 on page 120.

Attention: The semantics of recovering a partition before the current DDL version has the same semantics as that of the base DB2 for z/OS engine, as discussed in 1.3.3, “Recovery semantics” on page 16.

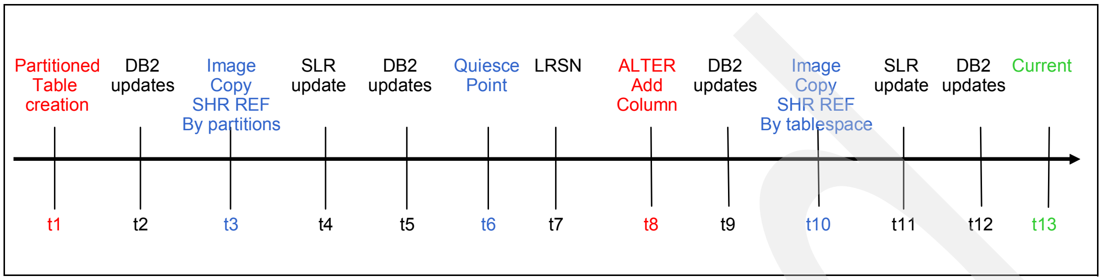

Figure 2-60 Recovering partition to an image copy scenario timeline

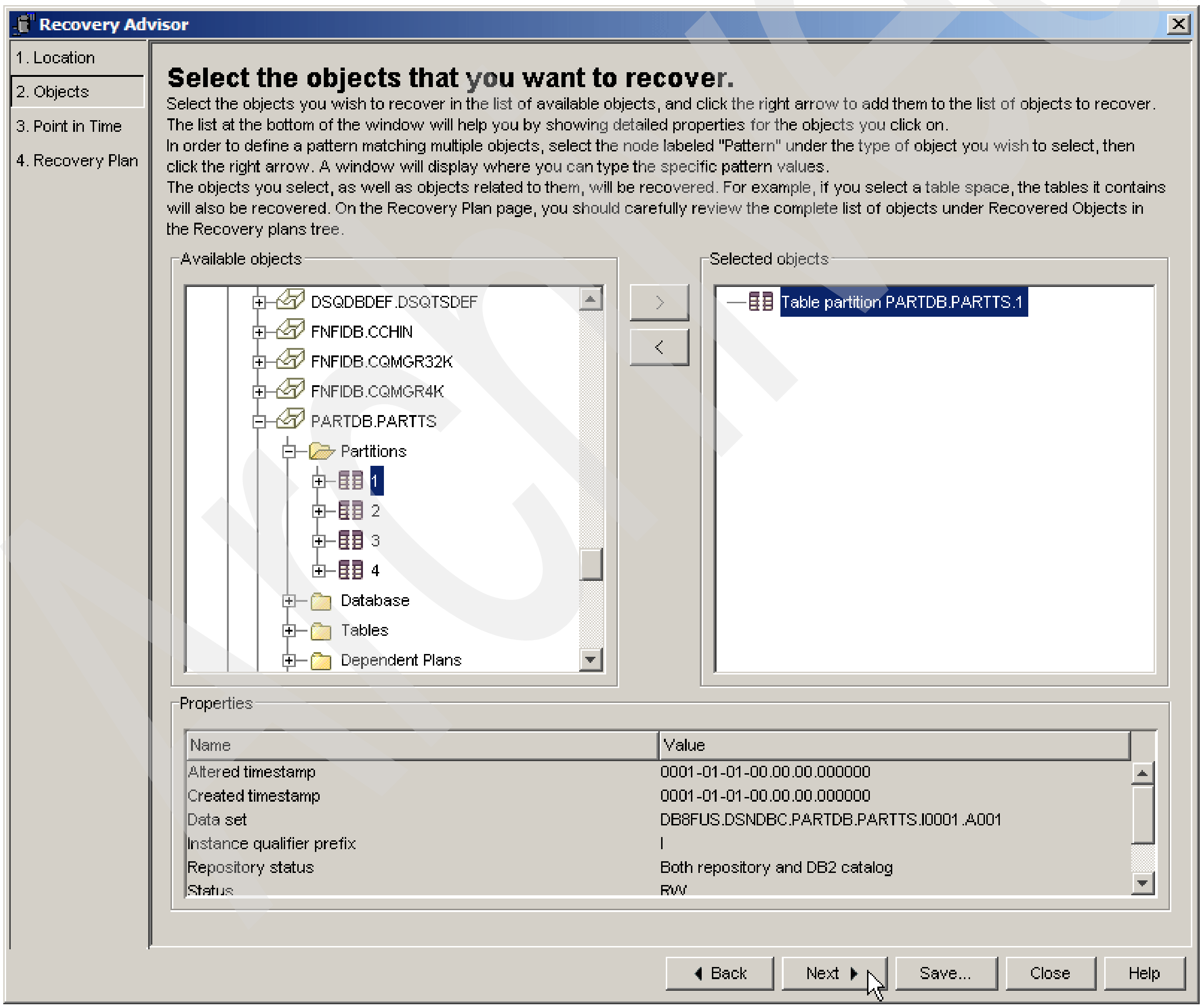

Figure 2-61 Selecting the partition to recover

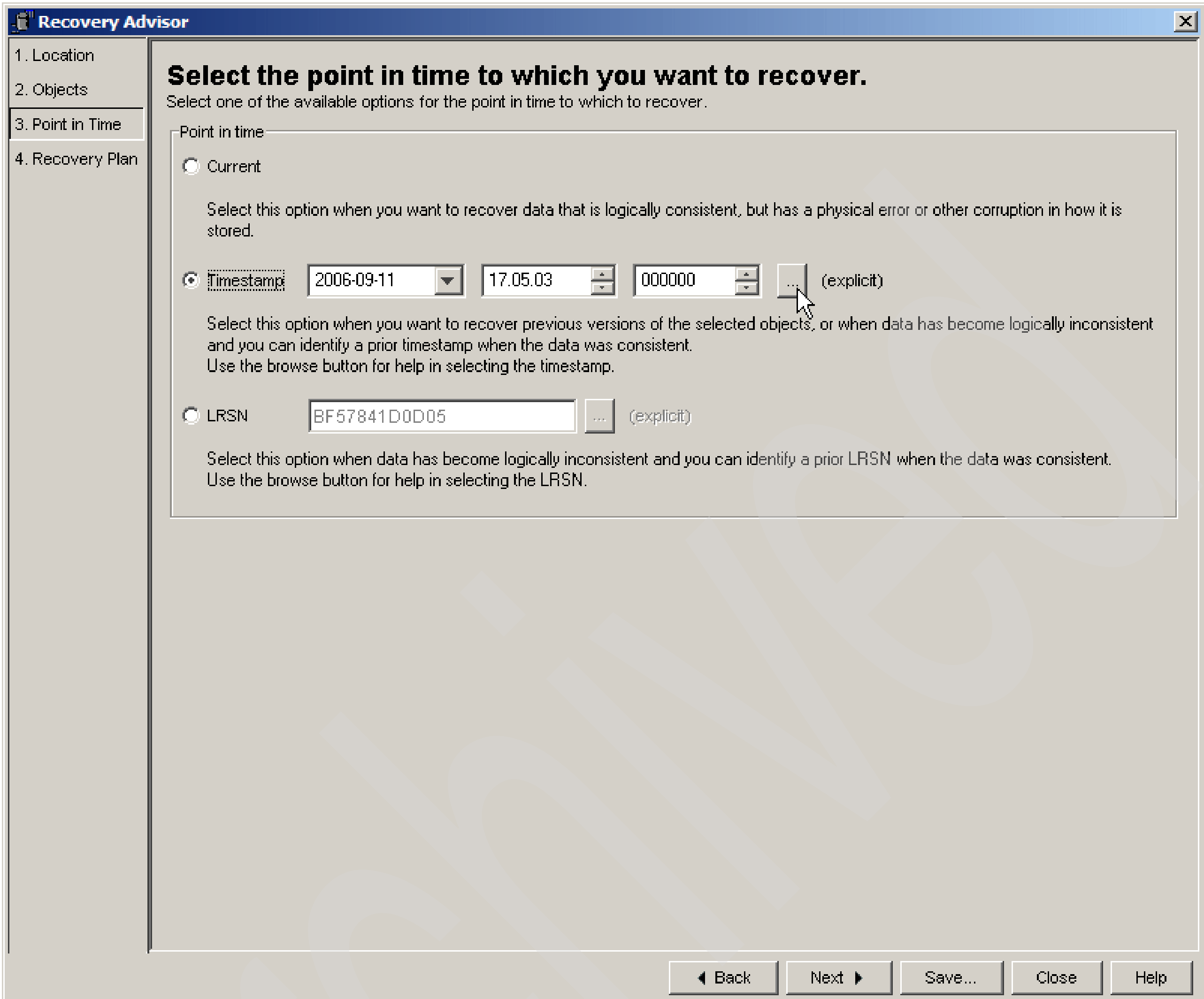

Figure 2-62 Selecting the point-in-time to recover to (1/3)

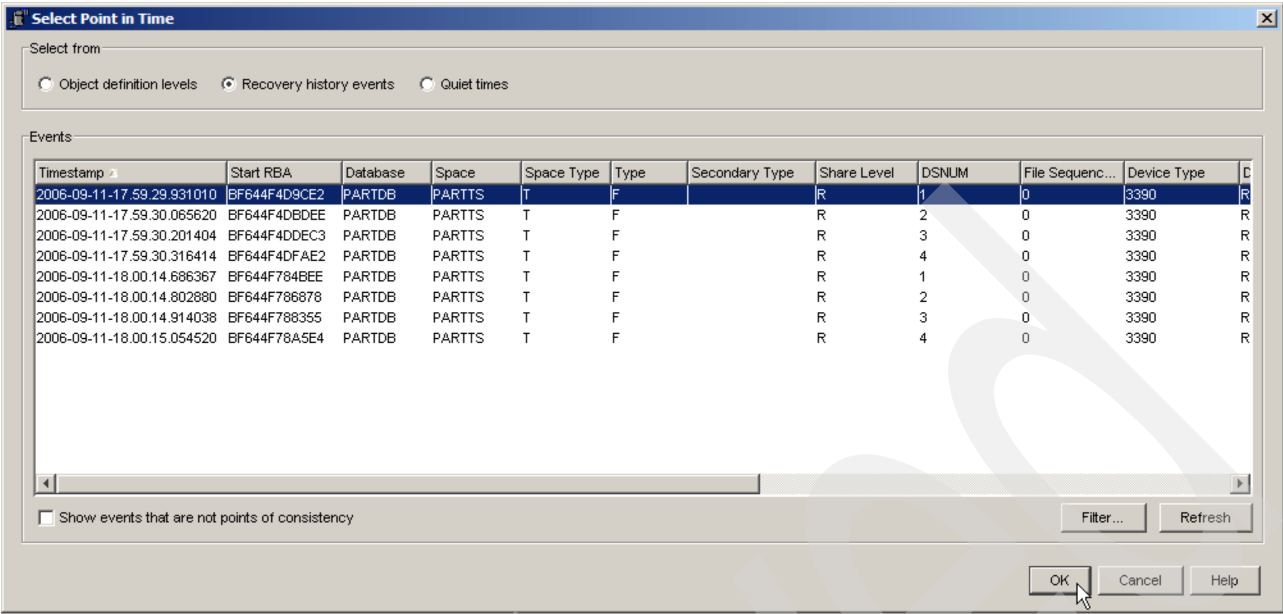

Figure 2-63 Selecting the point-in-time to recover to (2/3)

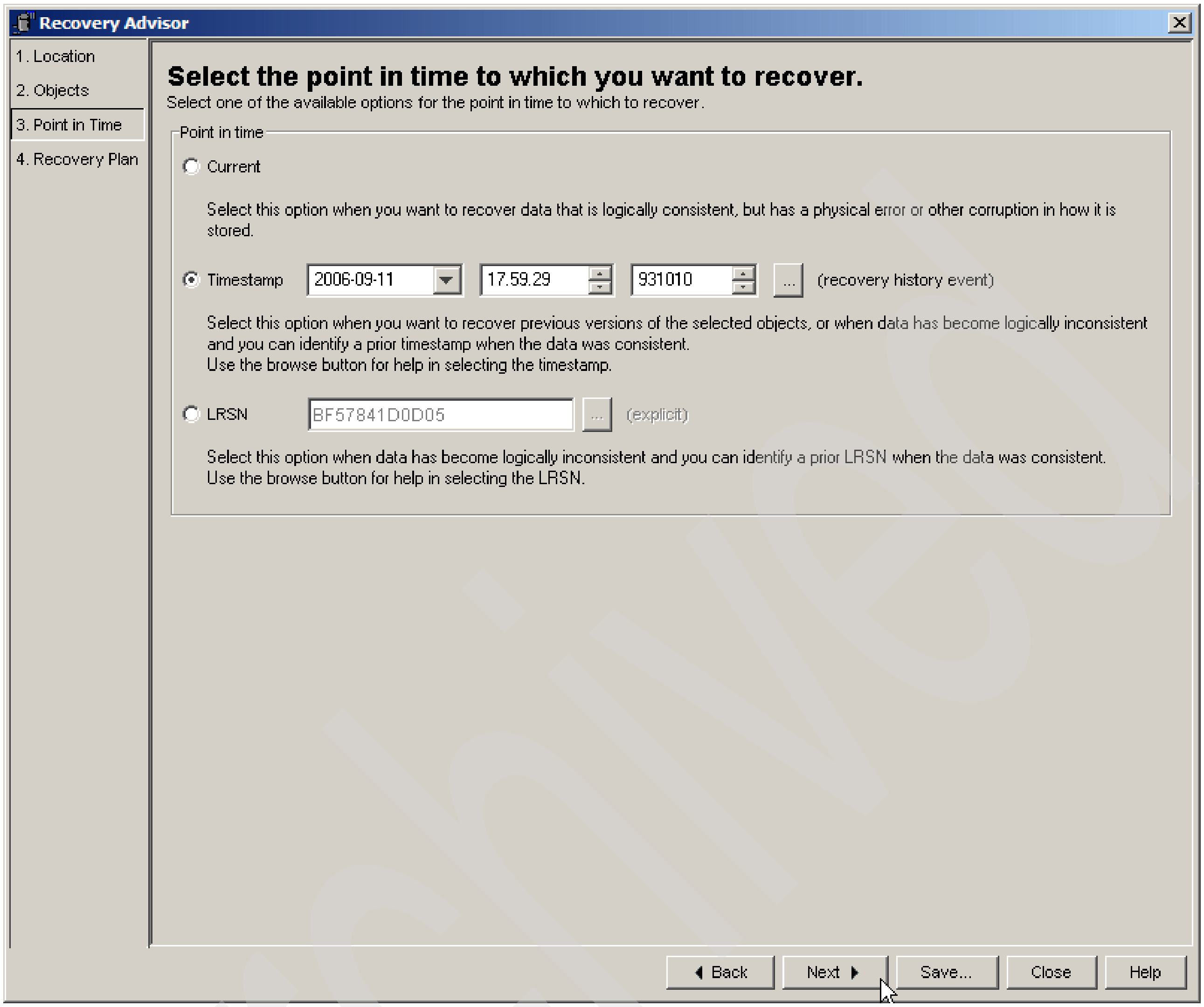

Figure 2-64 Selecting the point-in-time to recover to (3/3)

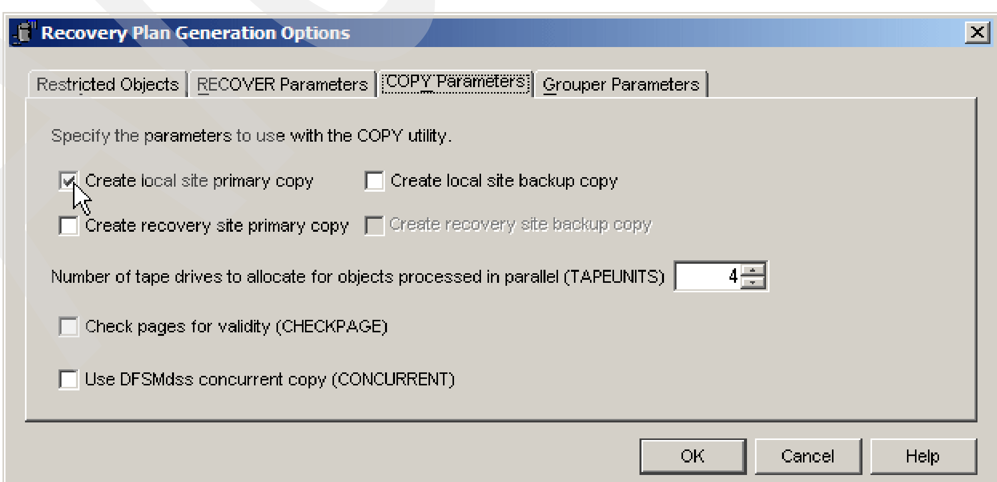

Figure 2-65 Copy parameters for recovery plan generation

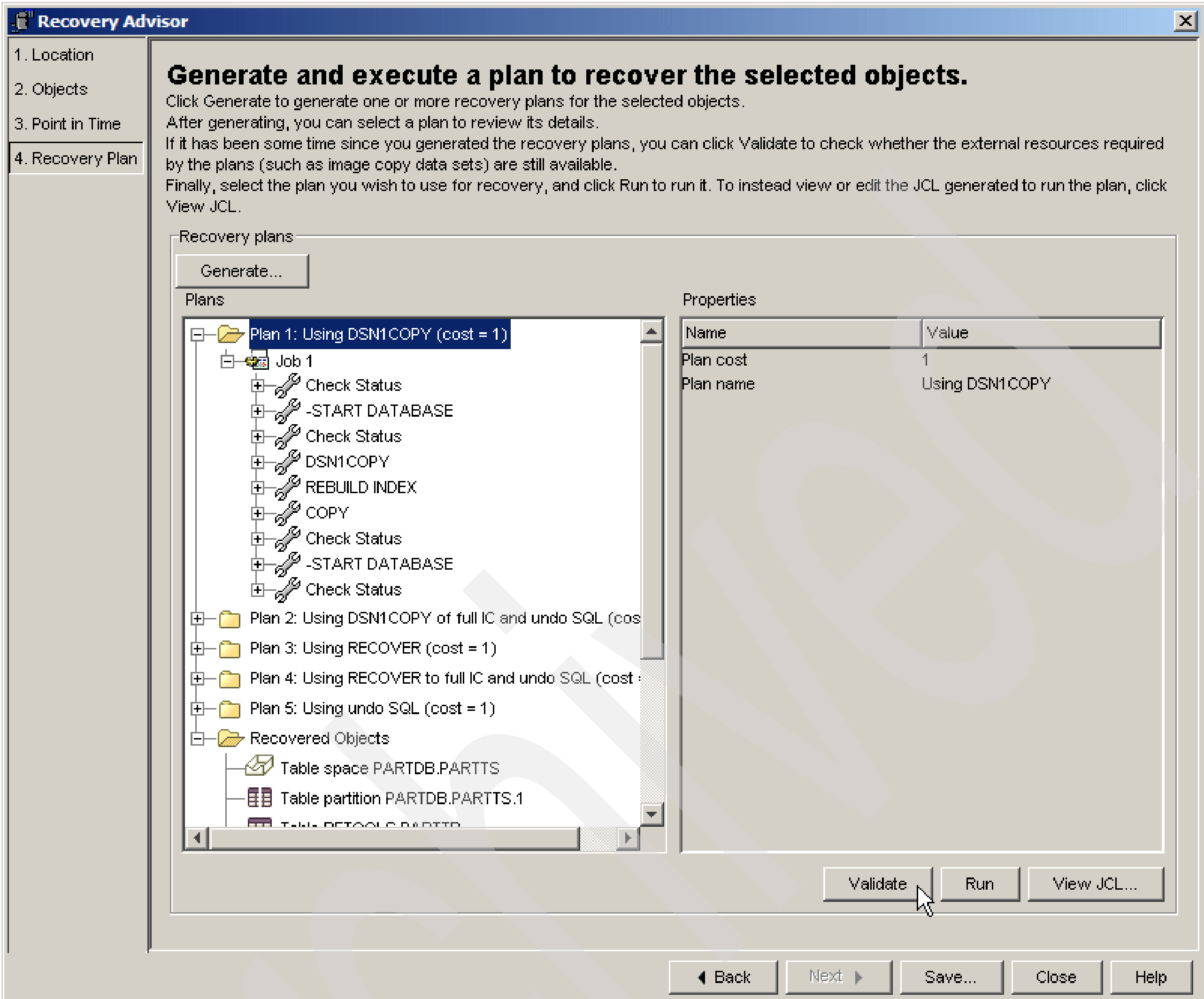

Figure 2-66 Generated recovery plans

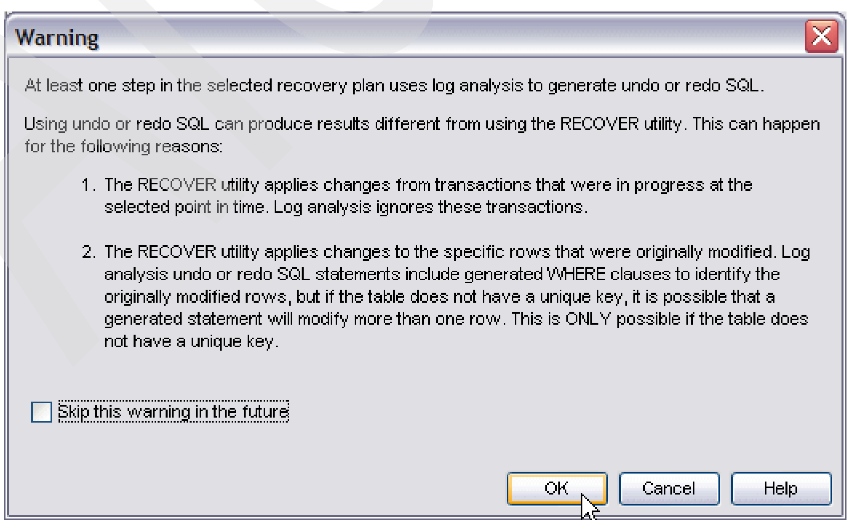

Figure 2-67 Warning message regarding log analysis being used in the selected recovery plan

Example 2-13 Partial contents of the generated JCL for Plan 1

```
//F8PREIC1 JOB ,'DB2 RECOVERY EXPERT',MSGCLASS=H,
//           REGION=0M,NOTIFY=&SYSUID
.................
.................
//DCPY0004 EXEC PGM=DSN1COPY,COND=(4,LT),
// PARM='CHECK,PAGESIZE(4K),FULLCOPY,NUMPARTS(4),RESET'
//STEPLIB  DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//SYSUT1   DD DISP=SHR,DSN=REIC.D8FG.PARTDB.PARTTS.FULL.P1.G0004V00
//SYSUT2   DD DISP=SHR,DSN=DB8FUS.DSNDBC.PARTDB.PARTTS.I0001.A001
//SYSPRINT DD   SYSOUT=*
//*
//RBLD0005 EXEC PGM=DSNUTILB,PARM=(D8F1),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//UTPRINT  DD   SYSOUT=*
//SYSOUT   DD   UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN    DD   *

  REBUILD INDEX
    ( "RETOOLS"."XPART1"
    , "RETOOLS"."XPART2" )
      SCOPE ALL

/*
//*
//COPY0006 EXEC PGM=DSNUTILB,PARM=(D8F1),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//CPY00010 DD   DSN=RETOOLS.ARY06254.S180813.M325966.COPY,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(NEW,CATLG,DELETE)
//SYSIN    DD   *

  COPY
    TABLESPACE PARTDB.PARTTS     DSNUM 1
      FULL YES
      COPYDDN(CPY00010)

/*
..................
..................
```

2.4.4 Recovering a partition to LRSN before current DDL version

This scenario describes the recovery of a single partition of a partitioned table space PARTTS to an LRSN corresponding to a time before the last DDL change that occurred on the partitioned table, that is, before the current DDL version of the object. The timeline shown in Figure 2-60 on page 117 is assumed for this scenario. This environment is a data sharing environment.

The following windows only represent the ones of particular interest (in our opinion) to this scenario. The windows leading up to these windows and those that follow these windows are almost identical to the windows described in 2.3.1, “Recovering multi-table table space” on page 56. We show the partial contents of the generated JCL for a given recovery plan, but not the execution job results.

Figure 2-68 on page 124 shows partition 1 of the partitioned table space PARTTS selected for recovery by expanding the table spaces folder in the navigation pane. Figure 2-69 on page 125 shows a user-specified LRSN as the point-in-time to recover to (we assume that the user has determined this LRSN to be the appropriate point to recover to by using some external mechanism such as a shutdown point of the DB2 subsystem). This LRSN corresponds to time t7 in Figure 2-60 on page 117. Figure 2-70 on page 125 shows the COPY parameters for recovery plan generation. Figure 2-71 on page 126 shows seven potential recovery plans being generated for the requested recovery action, and the expanded steps included in the recovery plan, Plan 5. Because the selected Plan 5 has a log analysis step, a warning message is issued, as shown in Figure 2-72 on page 126. A portion of the generated JCL for Plan 5 is shown in Example 2-14 on page 127.

The selected recovery plan, Plan 5, has multiple steps that include the restore of partition 1 from a full image copy followed by a rebuilding of the indexes. The next step is the generation of redo SQL from the DB2 log using log analysis for table space up to the RBA specified. The log analysis is bounded by the start LRSN (BF656182F9EC) corresponding to the image copy chosen by DSN1COPY to restore from, and the end LRSN (BF6561C00000) corresponding to the user-specified point to recover to in Figure 2-69 on page 125. This is followed by the execution of this generated SQL to complete the recovery action requested, and an image copy of the partition, as specified in Figure 2-70 on page 125.

Attention: The semantics of recovering a partition before the current DDL version has the same semantics as that of the base DB2 for z/OS engine, as discussed in 1.3.3, “Recovery semantics” on page 16.

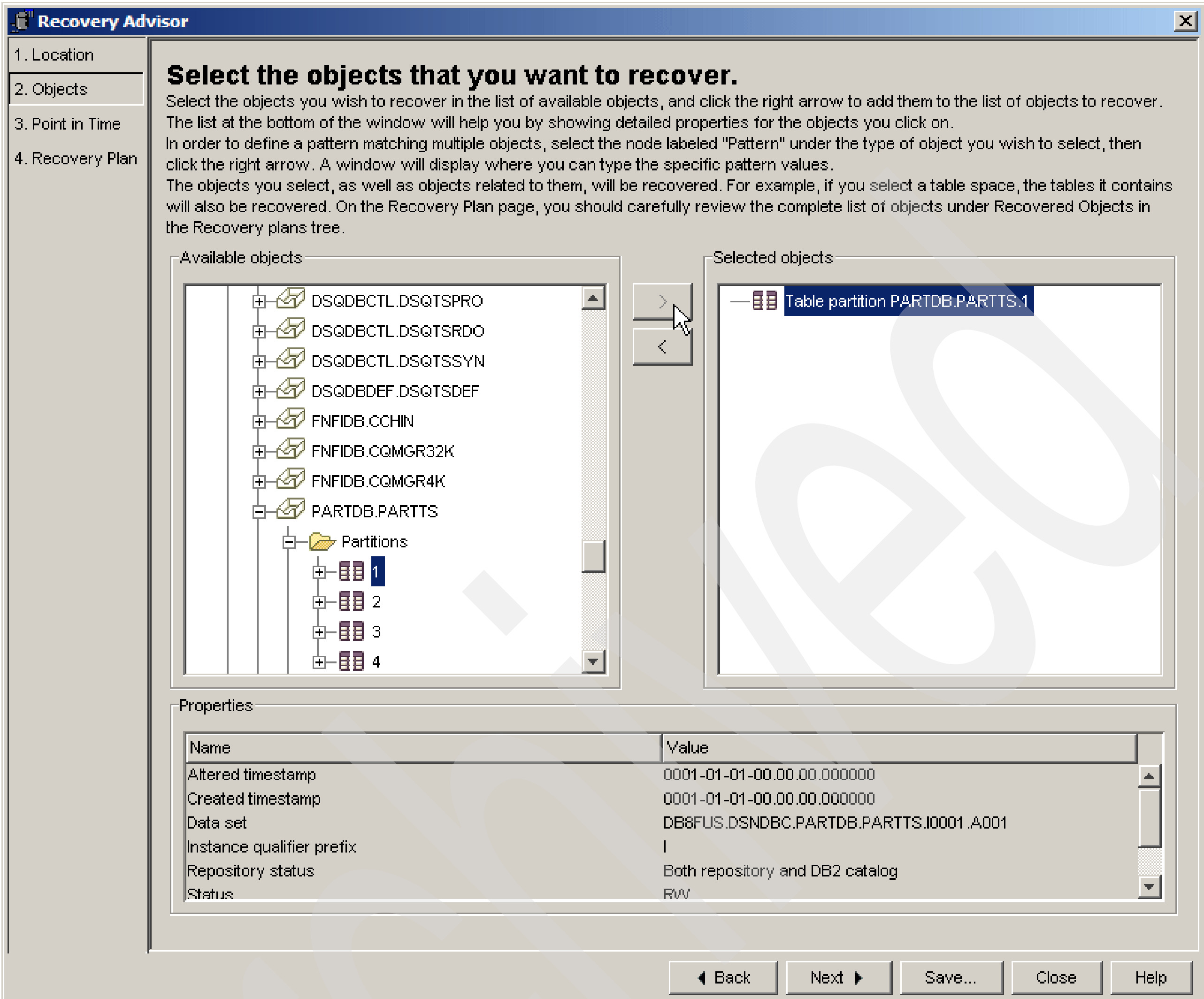

Figure 2-68 Selecting partition 1 of partitioned table space PARTTS

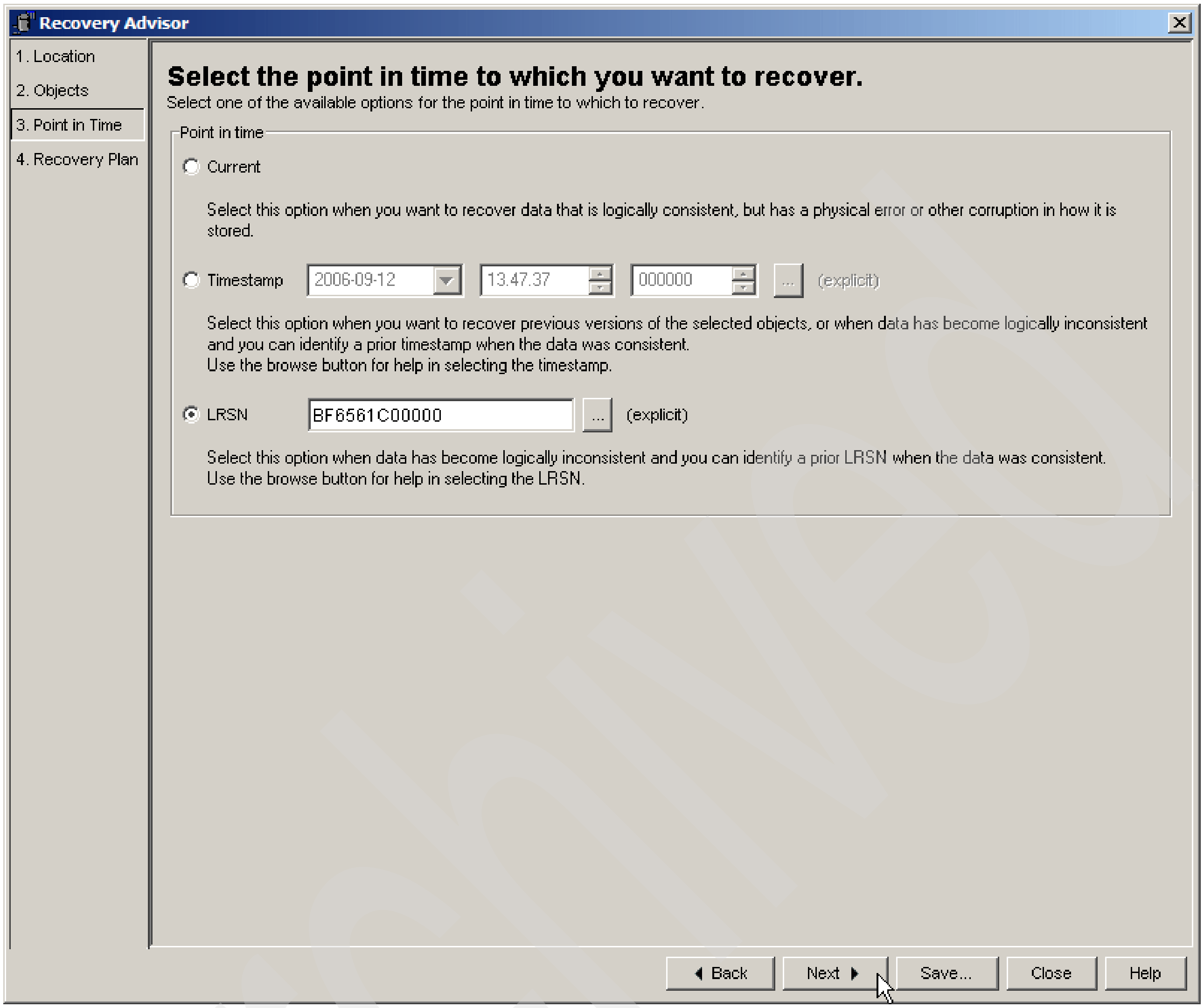

Figure 2-69   Selecting the LRSN to recover to

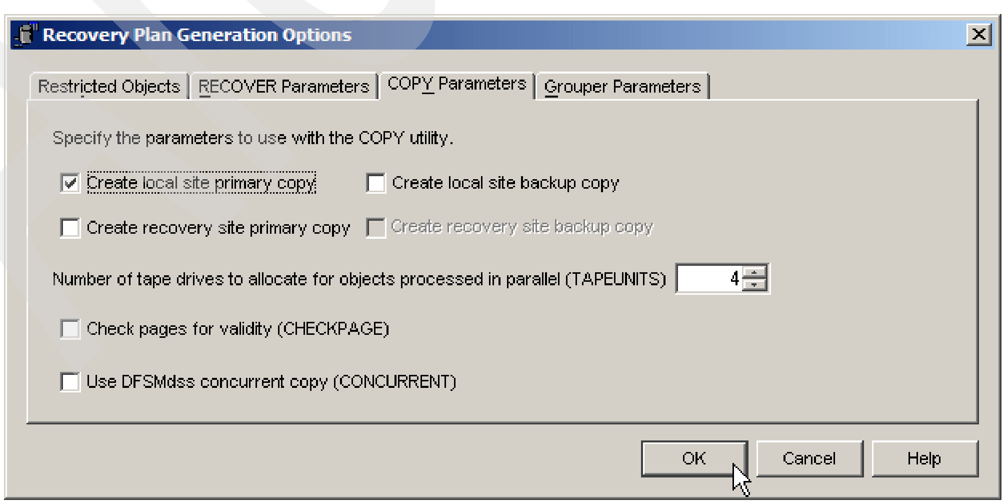

Figure 2-70   Copy parameters for recovery plan generation

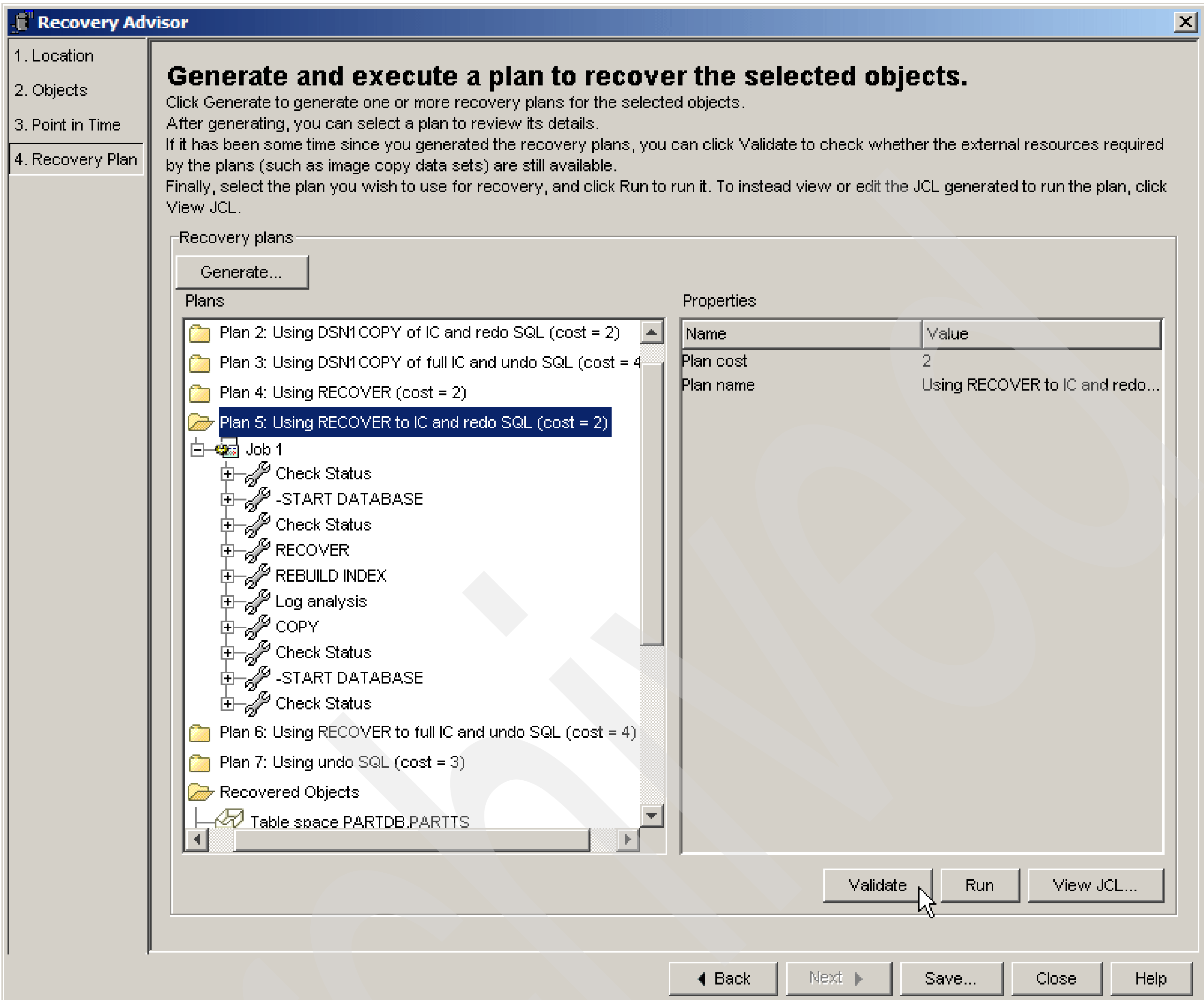

Figure 2-71 Generated recovery plans

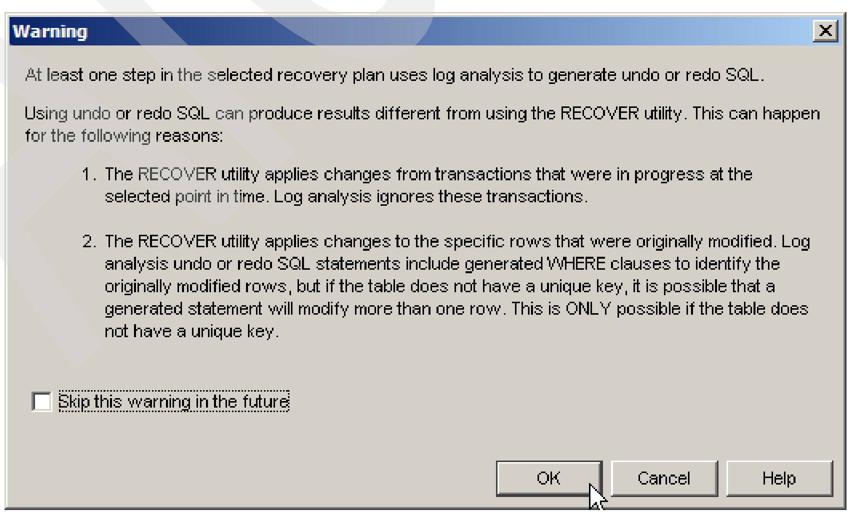

Figure 2-72 Warning message regarding log analysis step in the selected recovery plan

Example 2-14 Partial contents of the generated JCL for Plan 5

```
//F8PRERBA JOB ,'DB2 RECOVERY EXPERT',MSGCLASS=H,
//           REGION=0M,NOTIFY=&SYSUID
//*
...............
...............
//RCVR0004 EXEC PGM=DSNUTILB,PARM=(D8F1),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *

  RECOVER
    TABLESPACE PARTDB.PARTTS     DSNUM 1
      TOCOPY REIC.D8FG.PARTDB.PARTTS.FULL.P1.G0003V00
      LOCALSITE

/*
//*
//RBLD0005 EXEC PGM=DSNUTILB,PARM=(D8F1),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//UTPRINT  DD   SYSOUT=*
//SYSOUT   DD   UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN    DD   *

  REBUILD INDEX
    ( "RETOOLS"."XPART1"
    , "RETOOLS"."XPART2" )
      SCOPE ALL

/*
................
................
//*******************************************************************
//* STEP 2: READ THE DB2 LOG TO GENERATE THE GENERAL REPORT         *
//*******************************************************************
//LASR0002 EXEC PGM=ARYGEN1,REGION=0M,COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//MODEFILE DD   DSN=RETOOLS.ARYLAS.MODE.R0000181,
// DISP=OLD
//SYSOUT   DD   SYSOUT=*
//CFILES   DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//GENRPT   DD   SYSOUT=*
//EXTREP   DD   SYSOUT=*
//SUMREPT  DD   SYSOUT=*
//XDREPT   DD   SYSOUT=*
//QTRPT    DD   SYSOUT=*
//WARNINGS DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*
```

```
//SYSUDUMP DD   SYSOUT=*
//DATAIN   DD   *
SSID           =D8F1
START LRSN     =BF656182F9EC
END LRSN       =BF6561C00000
/*
//FILTERS  DD *
GENERAL FILTERS...........
SHOW UPDATES   =Y
SHOW INSERTS   =X
SHOW DELETES   =Y
SHOW ROLLBACKS =N
CATALOG DATA   =N
SHOW UNCOMMITS =N
OBJECT FILTERS-BY NAME....
DBNAME     =PARTDB
TSNAME     =PARTTS
PARTITION  =0001
SLRSN      =BF656182F9EC
ELRSN      =BF6561C00000
INC/EXC    =I
/*
//**********************************
//* PARAMETERS FOR JOB SHOWN BELOW *
//**********************************
//* SSID           =D8F1
//* START LRSN     =BF656182F9EC
//* END LRSN       =BF6561C00000
//* /*
//* //FILTERS  DD *
//* GENERAL FILTERS...........
//* SHOW UPDATES   =Y
//* SHOW INSERTS   =X
//* SHOW DELETES   =Y
//* SHOW ROLLBACKS =N
//* CATALOG DATA   =N
//* SHOW UNCOMMITS =N
//* OBJECT FILTERS-BY NAME....
//* DBNAME     =PARTDB
//* TSNAME     =PARTTS
//* PARTITION  =0001
//* SLRSN      =BF656182F9EC
//* ELRSN      =BF6561C00000
//* INC/EXC    =I
//*
.................
.................
//LASR0006 EXEC PGM=ARYDTL4,REGION=0M,COND=(4,LT)
//*******************************************************************
//*  STEP 6: THIS STEP DOES THE FOLLOWING:
//*         -SQL GENERATION
//*******************************************************************
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
```

```
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//MODEFILE DD   DSN=RETOOLS.ARYLAS.MODE.R0000181,
// DISP=OLD
//SQLOUT   DD   DSN=RETOOLS.ARY06255.S145023.M495175.EXECSQL,
//         DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE),
//         DCB=(LRECL=80,BLKSIZE=23440,RECFM=FB)
//SYSOUT   DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*                                                01870000
//SYSUDUMP DD   SYSOUT=*                                                01880000
//WARNINGS DD   SYSOUT=*                                                01890000
//CFILES   DD   SYSOUT=*
//*                                                                     01920000
//*
....................
....................
//LGNL0006 EXEC PGM=ARY#UTIL,PARM='EXECSQL,D8F1 ',
//             COND=((4,LT),(0,NE,LASR0002))
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
CONFIG DEFAULT
CONTINUE ON WARNING
//SYSIN    DD   DSN=RETOOLS.ARY06255.S145023.M495175.EXECSQL,
//             DISP=(OLD,KEEP)
//*
..................
..................
//COPY0007 EXEC PGM=DSNUTILB,PARM=(D8F1),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//CPY00010 DD   DSN=RETOOLS.ARY06255.S145024.M172581.COPY,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(NEW,CATLG,DELETE)
//SYSIN    DD   *

  COPY
    TABLESPACE PARTDB.PARTTS     DSNUM 1
      FULL YES
      COPYDDN(CPY00010)

/*
........................
........................
```

2.4.5 Recovering a partitioned table space to IC at current DDL version

This scenario describes the recovery of a partitioned table space to an image copy that was taken after the last DDL change that occurred on the partitioned table, that is, at the current DDL version of the object. The timeline shown in Figure 2-60 on page 117 is assumed for this scenario. This environment is a data sharing environment.

The following windows only represent the ones of particular interest (in our opinion) to this scenario. The windows leading up to these windows and those that following these windows are almost identical to the windows described in 2.3.1, “Recovering multi-table table space” on page 56. We show the partial contents of the generated JCL for a given recovery plan, but not the execution job results.

Figure 2-73 on page 131 shows the entire partitioned table space selected for recovery in the navigation pane. Figure 2-74 on page 132 through Figure 2-76 on page 134 show the selection of a full image copy (share level reference) from a list of available point-of-consistency entries in SYSIBM.SYSCOPY (recovery history events); not all window interactions are shown here. Figure 2-76 on page 134 shows this image copy selection translated to a timestamp corresponding to the image copy entry that is selected. Figure 2-77 on page 135 shows seven recovery plans generated for the requested recovery action, the recovered objects, and the expanded steps included in the recovery plan, Plan 7. Figure 2-78 on page 135 shows the warning message issued because a log analysis step is included in the selected recovery plan. A portion of the generated JCL for Plan 7 is shown in Example 2-15 on page 136. The selected recovery plan, Plan 7, has multiple steps that include a log analysis step that generates undo SQL, which is then run to complete the requested recovery action.

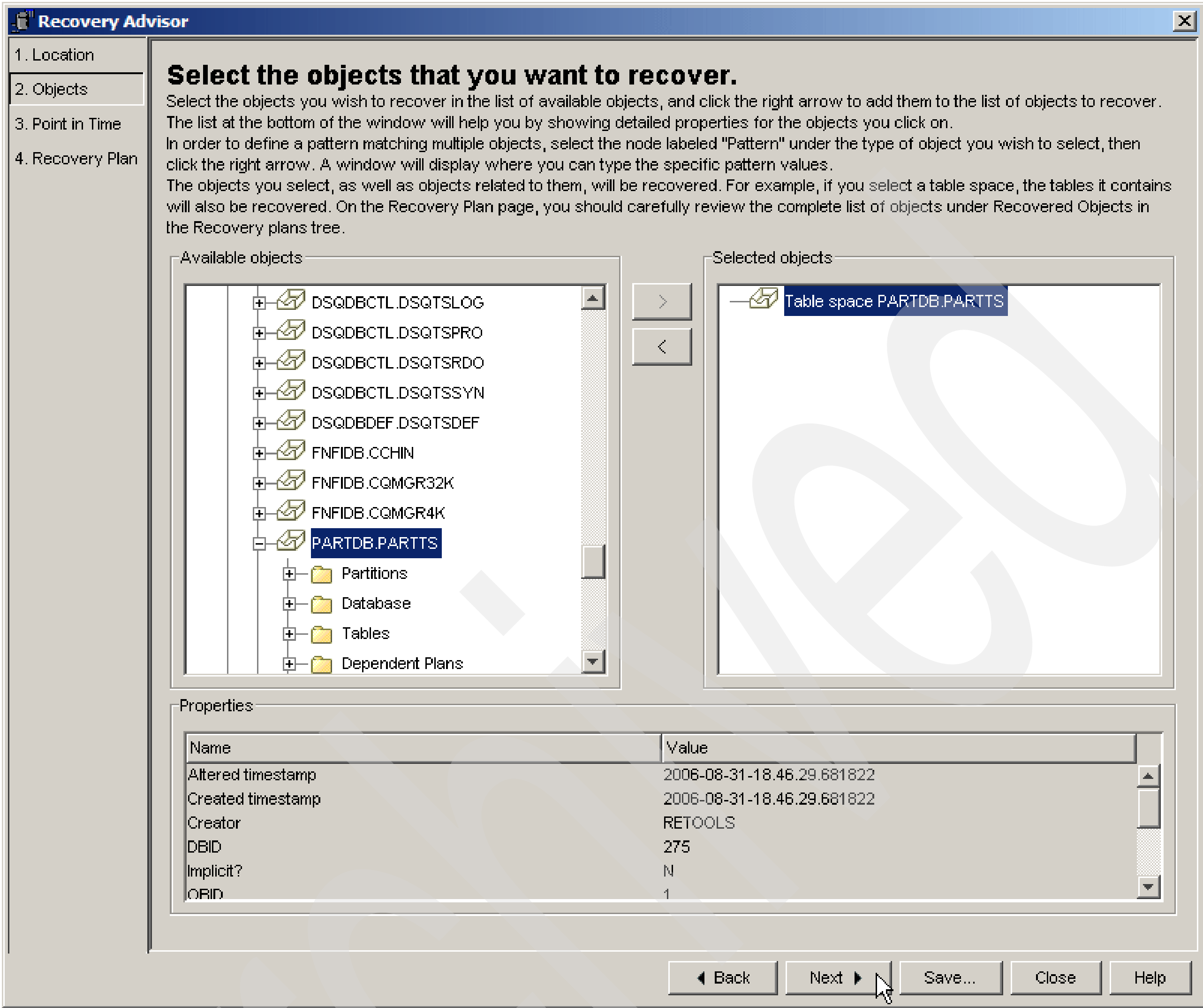

Figure 2-73 Selecting the partitioned table space PARTTS to recover

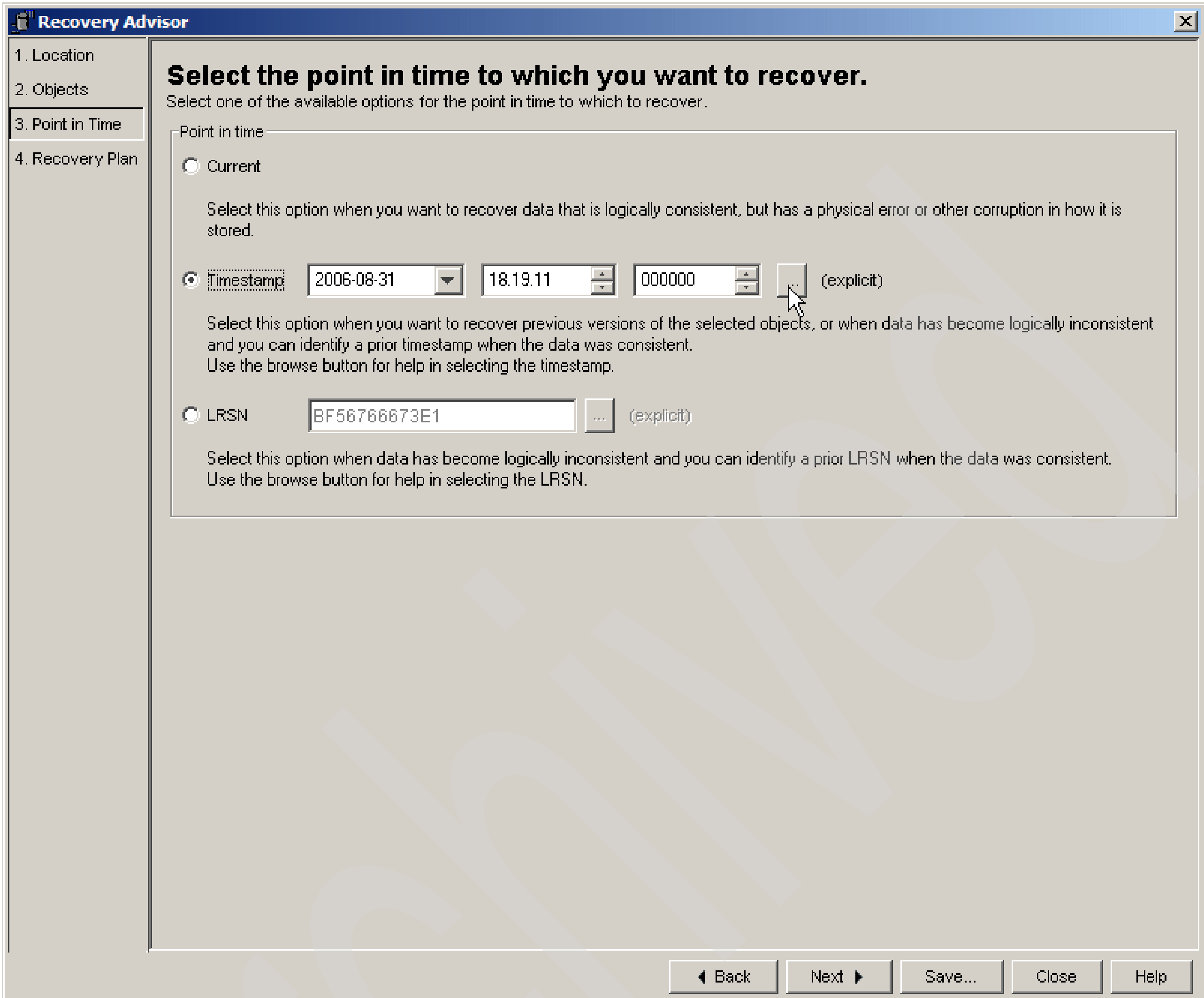

Figure 2-74 Selecting the point-in-time to recover to (1/3)

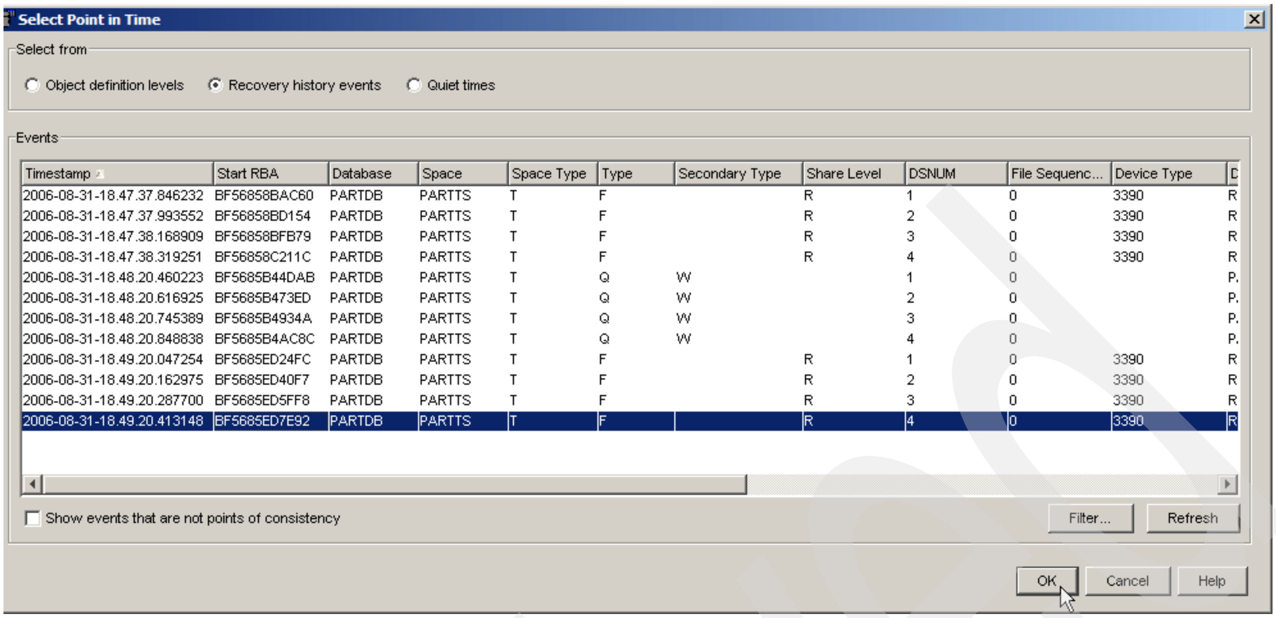

Figure 2-75 Selecting the point-in-time to recover to (2/3)

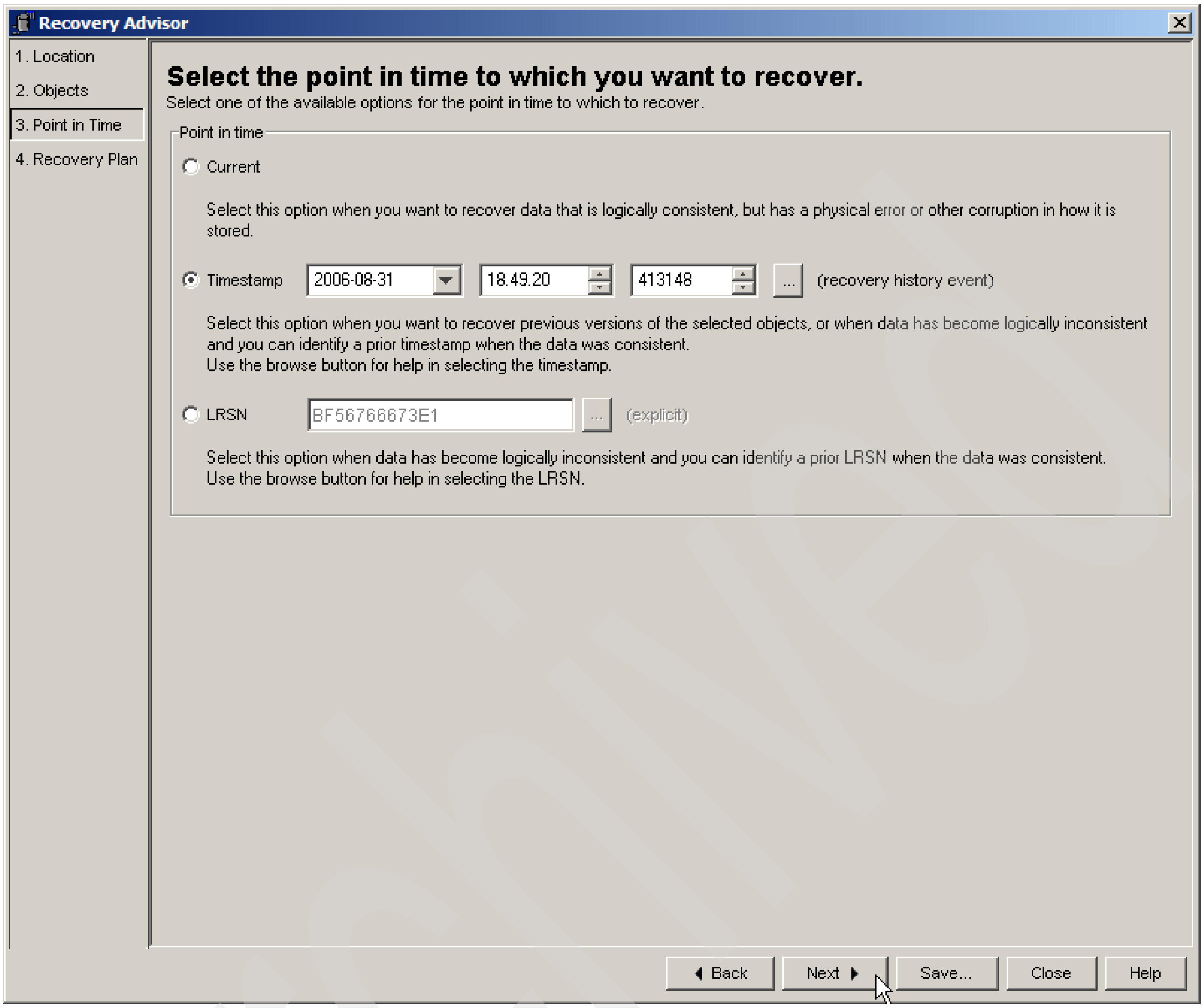

Figure 2-76 Selecting the point-in-time to recover to (3/3)

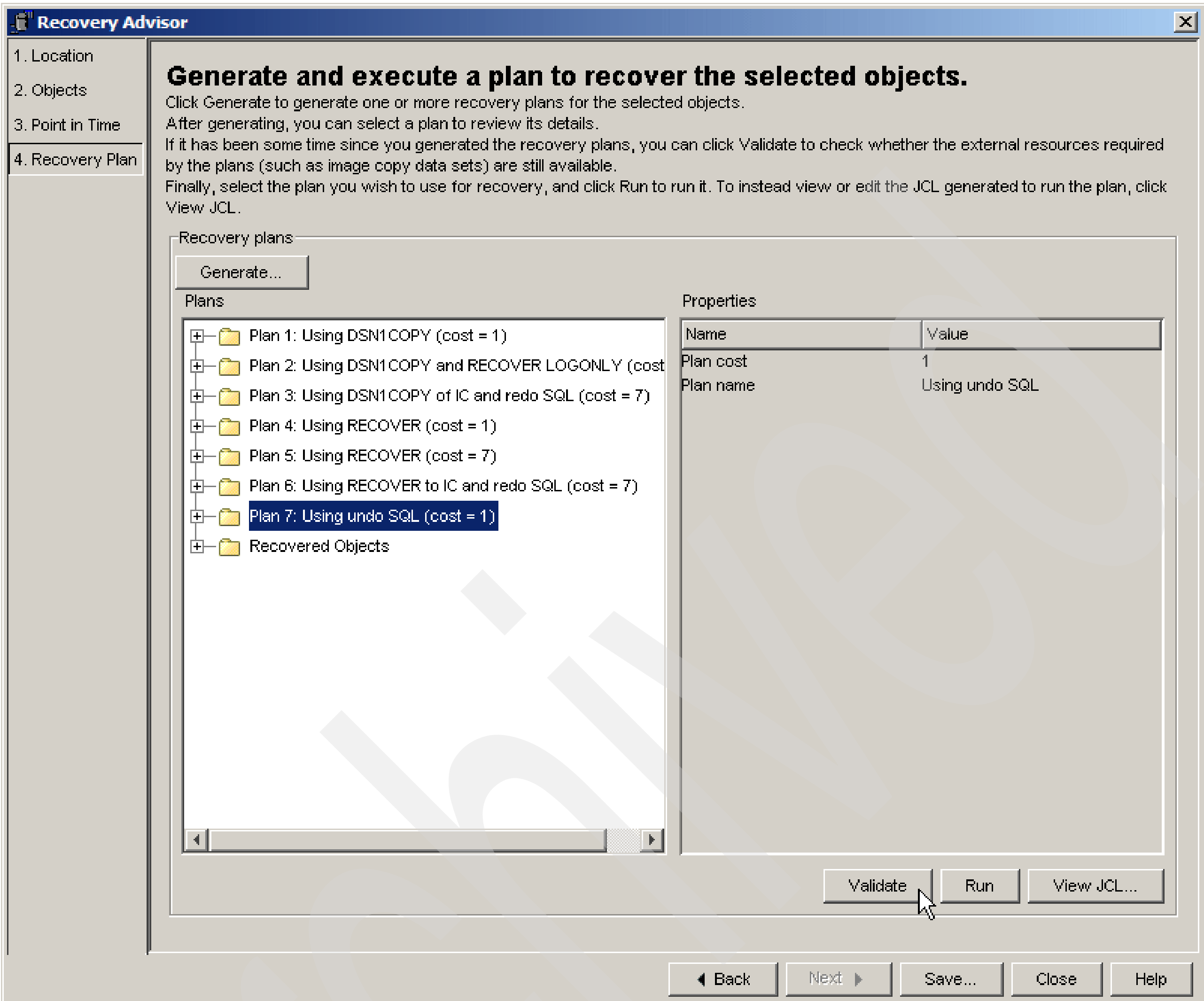

Figure 2-77 Generated recovery plans

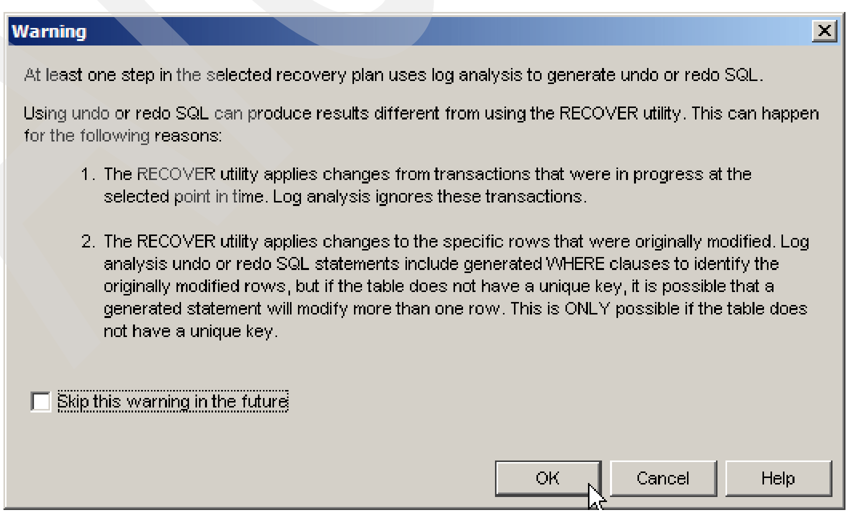

Figure 2-78 Warning message regarding log analysis step in the selected recovery plan

Example 2-15 Partial contents of the generated JCL for Plan 7

```
//F8PRECSQ JOB ,'DB2 RECOVERY EXPERT',MSGCLASS=H,
//            REGION=0M,NOTIFY=&SYSUID
...................
...................
//LASU0006 EXEC PGM=ARYDTL4,REGION=0M,COND=(4,LT)
//*******************************************************************
//*  STEP 6: THIS STEP DOES THE FOLLOWING:
//*         -SQL GENERATION
//*******************************************************************
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//MODEFILE DD   DSN=RETOOLS.ARYLAS.MODE.U0000478,
// DISP=OLD
//SQLOUT   DD   DSN=RETOOLS.ARY06243.S192053.M804018.EXECSQL,
//         DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE),
//         DCB=(LRECL=80,BLKSIZE=23440,RECFM=FB)
//SYSOUT   DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*                                            01870000
//SYSUDUMP DD   SYSOUT=*                                            01880000
//WARNINGS DD   SYSOUT=*                                            01890000
//CFILES   DD   SYSOUT=*
//*                                                                 01920000
//*
........................
........................
//LGNL0004 EXEC PGM=ARY#UTIL,PARM='EXECSQL,D8F1 ',
//             COND=((4,LT),(0,NE,LASU0002))
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
CONFIG DEFAULT
CONTINUE ON WARNING
//SYSIN    DD   DSN=RETOOLS.ARY06243.S192053.M804018.EXECSQL,
//             DISP=(OLD,KEEP)
//*
...................
...................
```

2.4.6 Recovering a database to a timestamp at current DDL version

This scenario describes the recovery of an entire database (aggregate object) to a timestamp that corresponds to a moment after the last DDL change that occurred on any object in that database, that is, at the current DDL version of all the objects in the database. In this scenario, we assume the timeline that corresponds in general to the type shown in Figure 2-60 on page 117. This environment is a data sharing environment.

The following windows only represent the ones of particular interest (in our opinion) to this scenario. The windows leading up to these windows and those that follow these windows are almost identical to the windows described in 2.3.1, “Recovering multi-table table space” on page 56. We show the partial contents of the generated JCL for a given recovery plan, but not the execution job results.

Figure 2-79 on page 138 shows the entire PARTDB database selected for recovery in the navigation pane. Figure 2-80 on page 139 shows the specification of a timestamp as the point to recover to (we assume that the user has determined this recovery point using some external source outside the scope of what is stored in the system). The timestamp being recovered to is some time between time t8 and t13 (current) in Figure 2-60 on page 117. Figure 2-81 on page 139 shows a warning message issued by Recovery Expert for z/OS because it cannot verify that the selected timestamp for recovery represents a point-of-consistency. Figure 2-82 on page 140 shows COPY parameters for recovery plan generation. Figure 2-83 on page 140 shows seven recovery plans being generated for the requested recovery action, the recovered objects, and the expanded steps included in the recovery plan, Plan 6. Figure 2-84 on page 141 shows the warning message issued because a log analysis step is included in the selected recovery plan. A portion of the generated JCL for Plan 6 is shown in Example 2-16 on page 141.

The selected recovery plan, Plan 6, has multiple steps that include a RECOVER of the individual partitions to the latest image copy associated with each partition, followed by a rebuilding of the indexes. The next step is log analysis that is bounded by the end LRSN (BF6686A42179) that corresponds to the timestamp specified in Figure 2-80 on page 139, and the start LRSN (BF6686063C48) that corresponds to the earliest timestamp of all the image copies used in the RECOVER to image copy statements. The applying of the generated SQL completes the recovery action requested. It is followed by image copies of all the table space partitions in the partitioned database, as specified in Figure 2-82 on page 140.

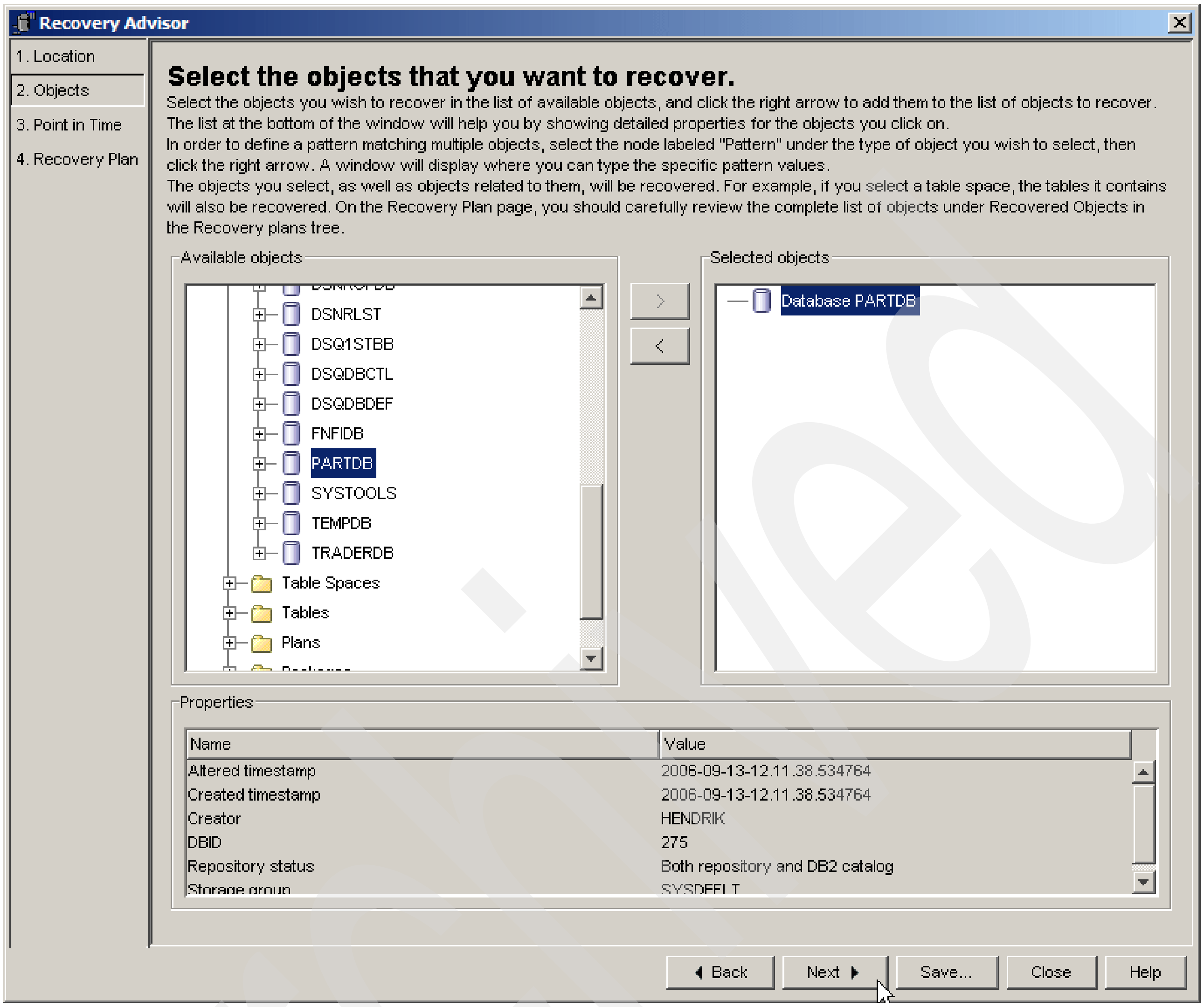

Figure 2-79 Selecting the database PARTDB to be recovered

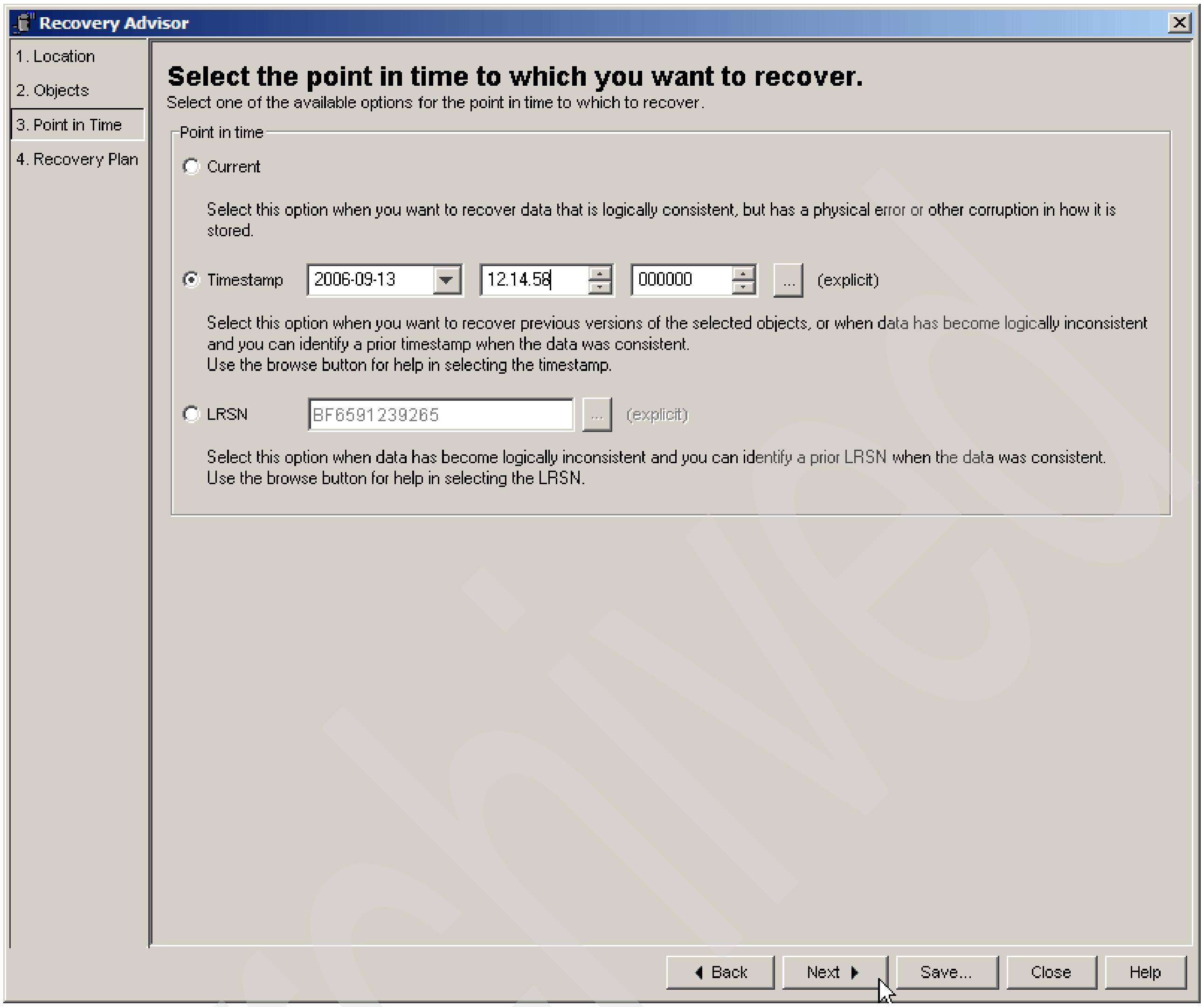

Figure 2-80 Specifying the timestamp to be recovered to

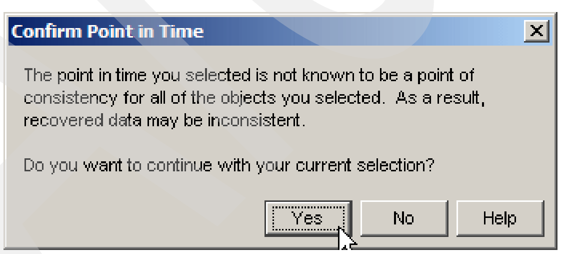

Figure 2-81 Warning message about potential point-of-consistency problem

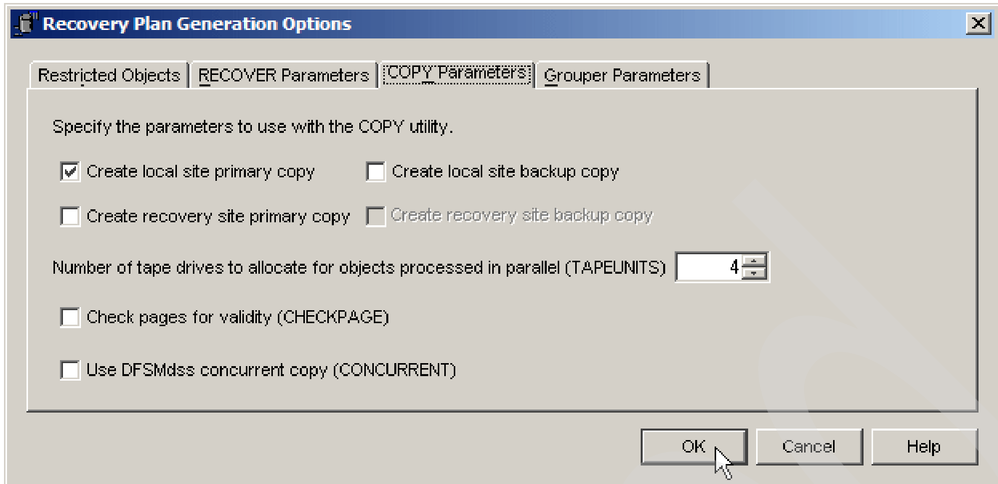

Figure 2-82 Copy parameters for recovery plan generation

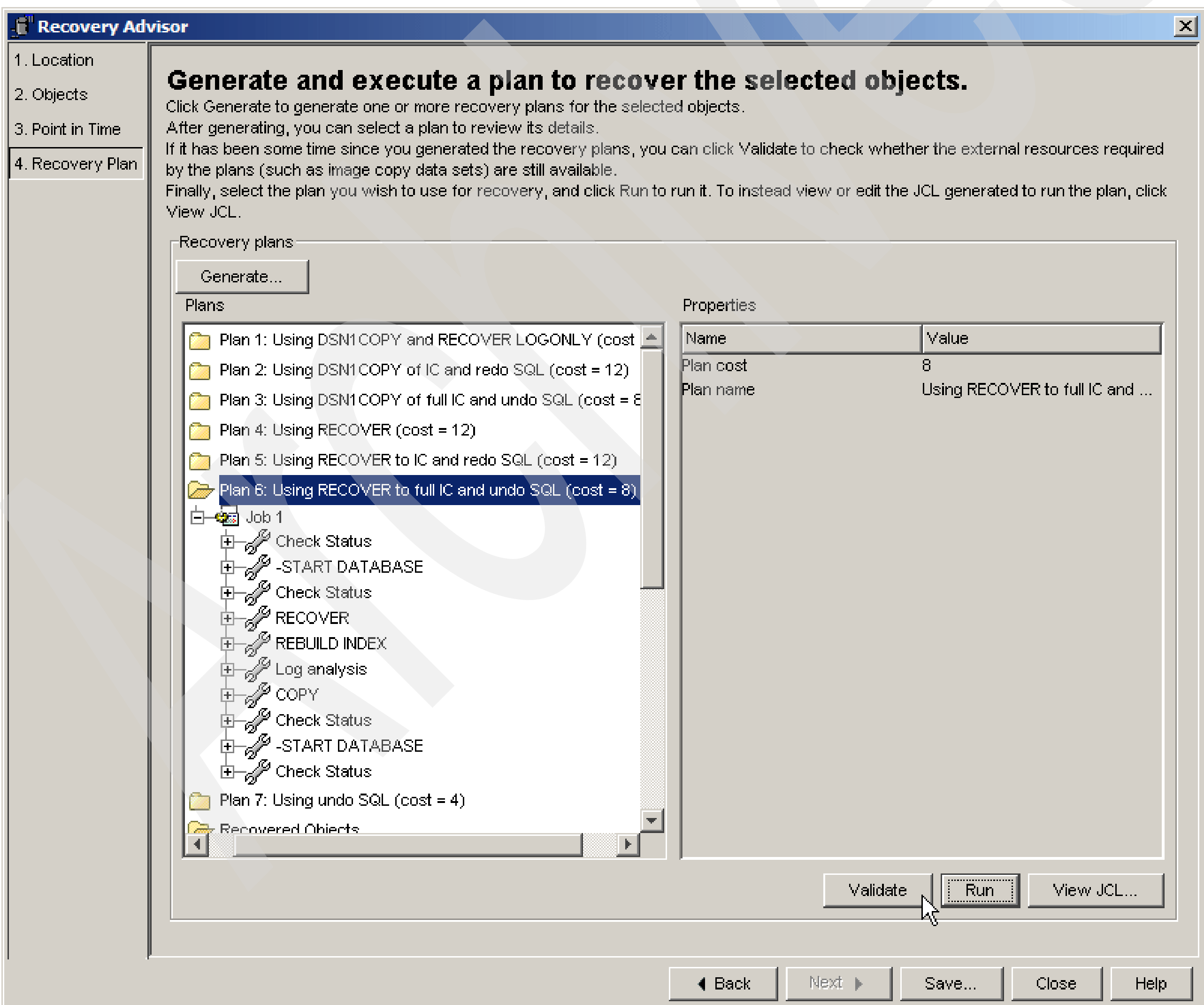

Figure 2-83 Generated recovery plans

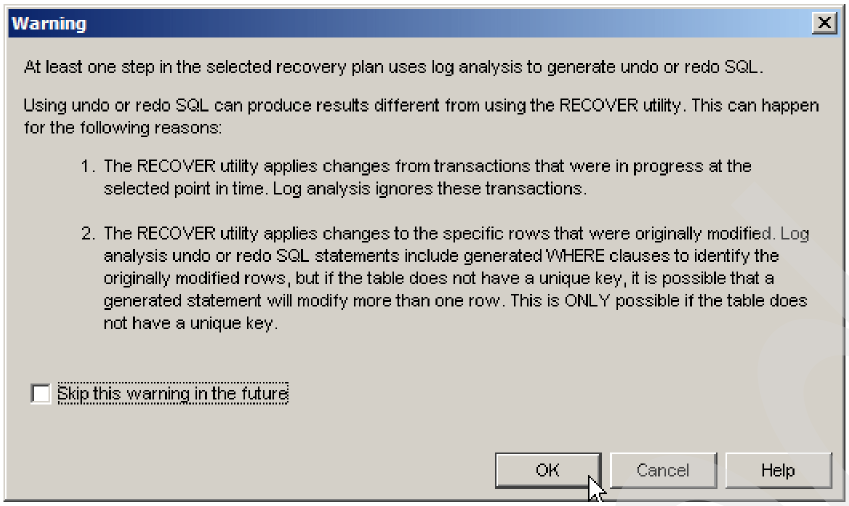

Figure 2-84 Warning message regarding log analysis step in the selected recovery plan

Example 2-16 Partial contents of the generated JCL for Plan 6

```
//F8PRECTS JOB ,'DB2 RECOVERY EXPERT',MSGCLASS=H,
//            REGION=0M,NOTIFY=&SYSUID
//*
................
................
//RCVR0004 EXEC PGM=DSNUTILB,PARM=(D8F1),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *

  RECOVER
    TABLESPACE PARTDB.PARTTS     DSNUM 1
      TOCOPY REIC.D8FG.PARTDB.PARTTS.FULL.P1.G0003V00
      LOCALSITE

  RECOVER
    TABLESPACE PARTDB.PARTTS     DSNUM 2
      TOCOPY REIC.D8FG.PARTDB.PARTTS.FULL.P2.G0003V00
      LOCALSITE

  RECOVER
    TABLESPACE PARTDB.PARTTS     DSNUM 3
      TOCOPY REIC.D8FG.PARTDB.PARTTS.FULL.P3.G0003V00
      LOCALSITE

  RECOVER
    TABLESPACE PARTDB.PARTTS     DSNUM 4
      TOCOPY REIC.D8FG.PARTDB.PARTTS.FULL.P4.G0003V00
      LOCALSITE

/*
//*
//RBLD0005 EXEC PGM=DSNUTILB,PARM=(D8F1),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8F8.SDSNEXIT
```

```
//          DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//UTPRINT  DD   SYSOUT=*
//SYSOUT   DD   UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN    DD   *

  REBUILD INDEX
    ( "RETOOLS"."XPART1"
    , "RETOOLS"."XPART2" )
      SCOPE ALL

/*
................
................
//*********************************************************************
//* STEP 2: READ THE DB2 LOG TO GENERATE THE GENERAL REPORT           *
//*********************************************************************
//LASU0002 EXEC PGM=ARYGEN1,REGION=0M,COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//          DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//          DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//MODEFILE DD   DSN=RETOOLS.ARYLAS.MODE.U0000279,
// DISP=OLD
//SYSOUT   DD   SYSOUT=*
//CFILES   DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//GENRPT   DD   SYSOUT=*
//EXTREP   DD   SYSOUT=*
//SUMREPT  DD   SYSOUT=*
//XDREPT   DD   SYSOUT=*
//QTRPT    DD   SYSOUT=*
//WARNINGS DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//DATAIN   DD   *
SSID           =D8F1
START LRSN     =BF6686063C48
END LRSN       =BF6686A42179
/*
//FILTERS  DD *
GENERAL FILTERS...........
SHOW UPDATES   =Y
SHOW INSERTS   =X
SHOW DELETES   =Y
SHOW ROLLBACKS =N
CATALOG DATA   =N
SHOW UNCOMMITS =N
OBJECT FILTERS-BY NAME....
DBNAME     =PARTDB
TSNAME     =PARTTS
PARTITION  =0001
SLRSN      =BF6686063C48
ELRSN      =BF6686A3BF55
INC/EXC    =I
```

```
DBNAME      =PARTDB
TSNAME      =PARTTS
PARTITION   =0002
SLRSN       =BF6686063C48
ELRSN       =BF6686A3DD53
INC/EXC     =I
DBNAME      =PARTDB
TSNAME      =PARTTS
PARTITION   =0003
SLRSN       =BF6686063C48
ELRSN       =BF6686A3FB1B
INC/EXC     =I
DBNAME      =PARTDB
TSNAME      =PARTTS
PARTITION   =0004
SLRSN       =BF6686063C48
ELRSN       =BF6686A42179
INC/EXC     =I
/*
//**********************************
//* PARAMETERS FOR JOB SHOWN BELOW *
//**********************************
//* SSID           =D8F1
//* START LRSN     =BF6686063C48
//* END LRSN       =BF6686A42179
//* /*
//* //FILTERS  DD *
//* GENERAL FILTERS...........
//* SHOW UPDATES   =Y
//* SHOW INSERTS   =X
//* SHOW DELETES   =Y
//* SHOW ROLLBACKS =N
//* CATALOG DATA   =N
//* SHOW UNCOMMITS =N
//* OBJECT FILTERS-BY NAME....
//* DBNAME     =PARTDB
//* TSNAME     =PARTTS
//* PARTITION  =0001
//* SLRSN      =BF6686063C48
//* ELRSN      =BF6686A3BF55
//* INC/EXC    =I
//* DBNAME     =PARTDB
//* TSNAME     =PARTTS
//* PARTITION  =0002
//* SLRSN      =BF6686063C48
//* ELRSN      =BF6686A3DD53
//* INC/EXC    =I
//* DBNAME     =PARTDB
//* TSNAME     =PARTTS
//* PARTITION  =0003
//* SLRSN      =BF6686063C48
//* ELRSN      =BF6686A3FB1B
//* INC/EXC    =I
//* DBNAME     =PARTDB
//* TSNAME     =PARTTS
```

```
//* PARTITION  =0004
//* SLRSN      =BF6686063C48
//* ELRSN      =BF6686A42179
//* INC/EXC    =I
//*
...................
...................
//LASU0006 EXEC PGM=ARYDTL4,REGION=0M,COND=(4,LT)
//*********************************************************************
//*  STEP 6: THIS STEP DOES THE FOLLOWING:
//*         -SQL GENERATION
//*********************************************************************
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//MODEFILE DD   DSN=RETOOLS.ARYLAS.MODE.U0000279,
// DISP=OLD
//SQLOUT   DD   DSN=RETOOLS.ARY06256.S122901.M489026.EXECSQL,
//         DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE),
//         DCB=(LRECL=80,BLKSIZE=23440,RECFM=FB)
//SYSOUT   DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*                                                01870000
//SYSUDUMP DD   SYSOUT=*                                                01880000
//WARNINGS DD   SYSOUT=*                                                01890000
//CFILES   DD   SYSOUT=*
//*                                                                     01920000
//*
..............
..............
//LGNL0006 EXEC PGM=ARY#UTIL,PARM='EXECSQL,D8F1 ',
//             COND=((4,LT),(0,NE,LASU0002))
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
CONFIG DEFAULT
CONTINUE ON WARNING
//SYSIN    DD   DSN=RETOOLS.ARY06256.S122901.M489026.EXECSQL,
//             DISP=(OLD,KEEP)
//*
,,,,,,,,,,,,,,,,,,,
...................
//COPY0007 EXEC PGM=DSNUTILB,PARM=(D8F1),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//CPY00010 DD   DSN=RETOOLS.ARY06256.S122902.M107212.COPY,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(NEW,CATLG,DELETE)
//CPY00020 DD   DSN=RETOOLS.ARY06256.S122902.M111790.COPY,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(NEW,CATLG,DELETE)
```

```
//CPY00030 DD   DSN=RETOOLS.ARY06256.S122902.M113383.COPY,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(NEW,CATLG,DELETE)
//CPY00040 DD   DSN=RETOOLS.ARY06256.S122902.M114924.COPY,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(NEW,CATLG,DELETE)
//SYSIN    DD   *

  COPY
    TABLESPACE PARTDB.PARTTS     DSNUM 1
      FULL YES
      COPYDDN(CPY00010)
    TABLESPACE PARTDB.PARTTS     DSNUM 2
      FULL YES
      COPYDDN(CPY00020)
    TABLESPACE PARTDB.PARTTS     DSNUM 3
      FULL YES
      COPYDDN(CPY00030)
    TABLESPACE PARTDB.PARTTS     DSNUM 4
      FULL YES
      COPYDDN(CPY00040)

/*
......................
......................
```

2.4.7 Recovering a table space in referential set to IC at current DDL version

This scenario describes the recovery of a single table space RQUIETS3, which is part of a referential set, to an image copy that corresponds to a moment after the last DDL change that occurred on any object in that table space, that is, at the current DDL version of all the objects in the table space. In this scenario, we assume the timeline that corresponds in general to the type shown in Figure 2-50 on page 104. This environment is a data sharing environment.

The following windows only represent the ones of particular interest (in our opinion) to this scenario. The windows leading up to these windows and those that follow these windows are almost identical to the windows described in 2.3.1, “Recovering multi-table table space” on page 56. We show the partial contents of the generated JCL for a given recovery plan, but not the execution job results.

Figure 2-85 on page 146 shows the single table space RQUIETS3 in the database RQUIESCE that is part of a referential set selected for recovery in the navigation pane. Figure 2-86 on page 147 through Figure 2-88 on page 148 show the selection of an image copy (share level reference) from a list of available point-of-consistency entries in SYSIBM.SYSCOPY (recovery history events); not all window interactions are shown here. This corresponds to time t10 in Figure 2-50 on page 104. Figure 2-88 on page 148 shows this image copy selection translated to a timestamp corresponding to the image copy entry selected in Figure 2-86 on page 147. Figure 2-89 on page 148 shows COPY parameters being set for recovery plan generation. Figure 2-90 on page 149 shows Recovery Expert for z/OS identifying all the tables in the referential set for the selected table space RQUIETS3 that participates in the recovery. This is a most useful facility that eliminates user error that tends to occur when referential sets are involved in point-in-time recoveries. Figure 2-91 on page 150 shows nine recovery plans being generated for the requested recovery action, the recovered objects, and the expanded steps included in recovery plan, Plan 4. Figure 2-92 on page 151 shows the warning issued when log analysis is included in the selected recovery plan, Plan 4. A portion of the generated JCL for Plan 4 is shown in Example 2-17 on page 151.

The selected recovery plan, Plan 4, has multiple steps that include the restore of the selected table space using DSN1COPY, followed by a rebuilding of the indexes on the table in this table space. It also includes a step to perform CHECK DATA on this recovered table space. The next step is to generate undo SQL to be run on the remaining members of the referential set to correspond to the same PIT as the image copy. Therefore, the log analysis is bounded by the end of the log (LRSN FFFFFFFFFFFF) and the starting LRSN (BF57712BDFBC) corresponding to the image copy restored to, as shown in Figure 2-87 on page 147. The final step is to copy all the table spaces in the referential set as specified in the COPY parameters recovery plan generation shown in Figure 2-89 on page 148.

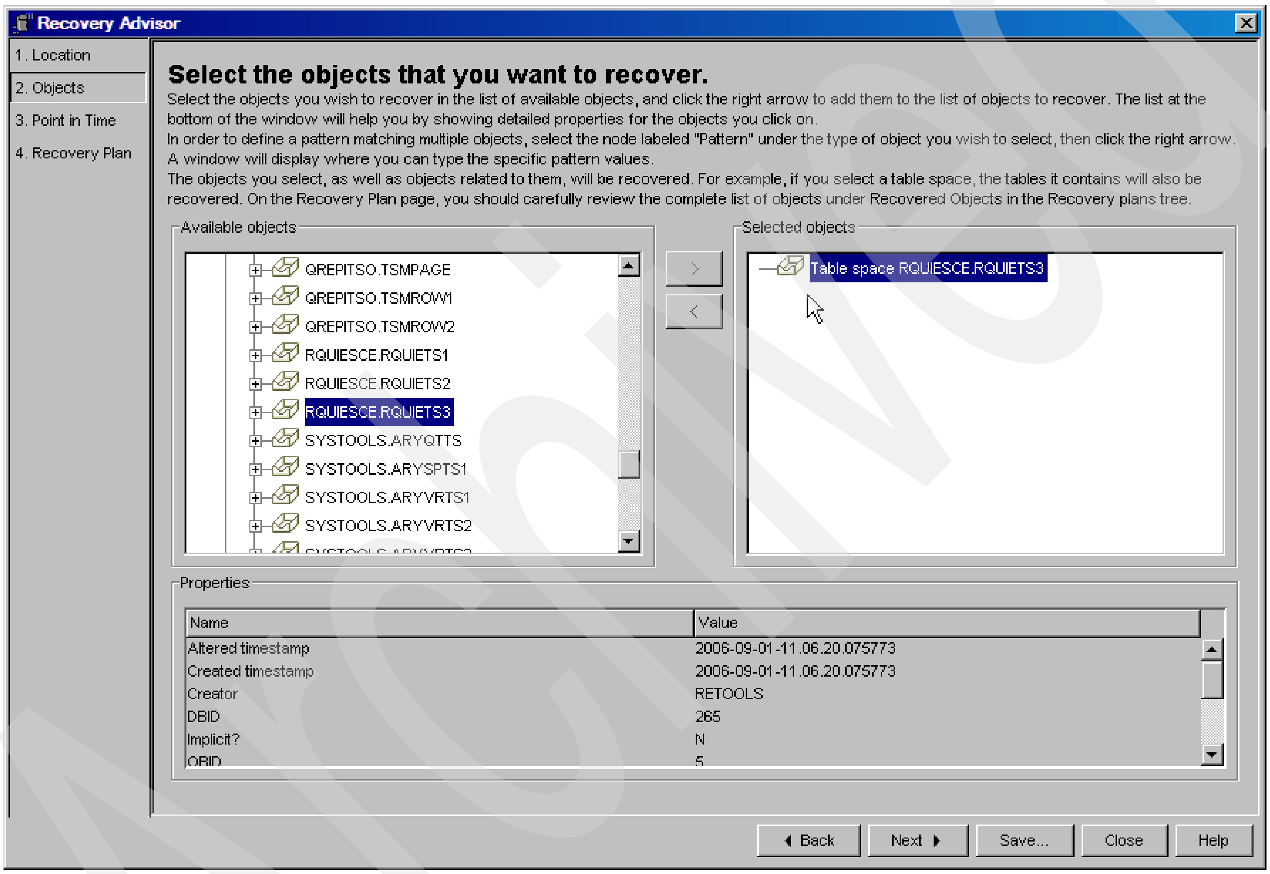

Figure 2-85 Selecting a single table space RQUIETSS3 in a referential set to recover

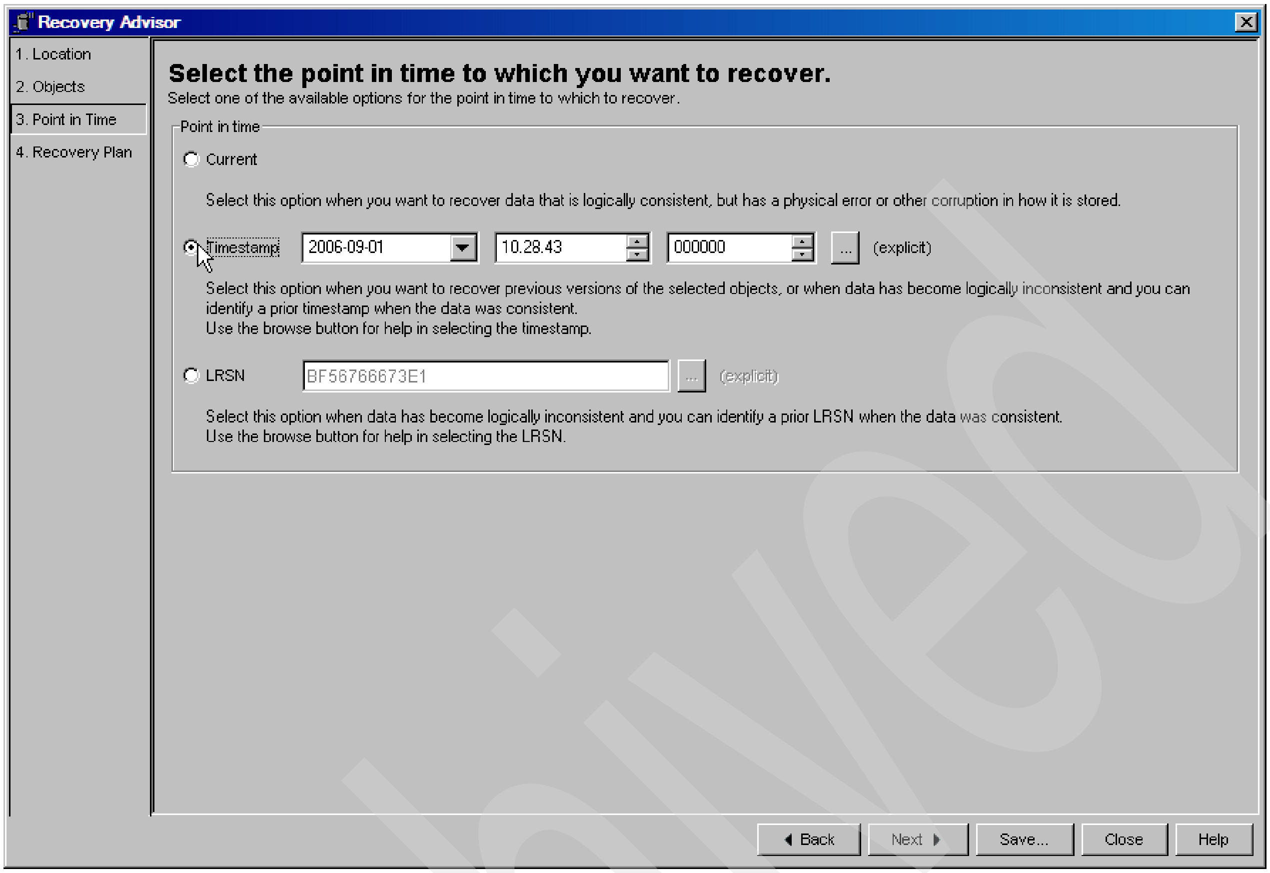

Figure 2-86 Selecting the point-in-time to recover to (1/3)

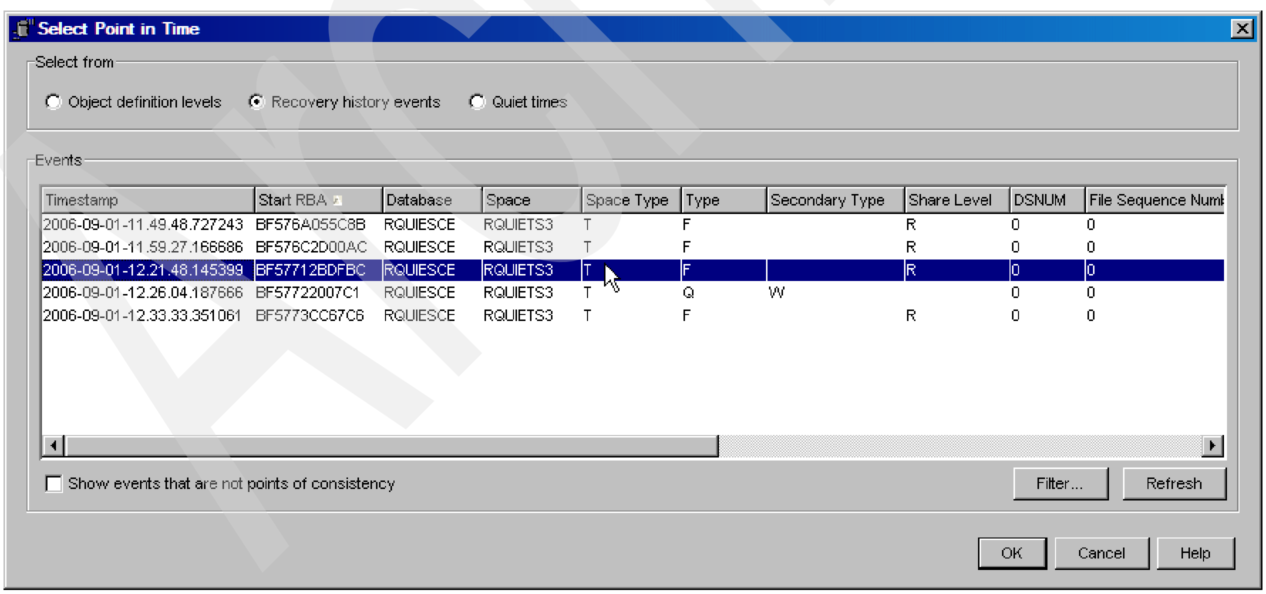

Figure 2-87 Selecting the point-in-time to recover to (2/3)

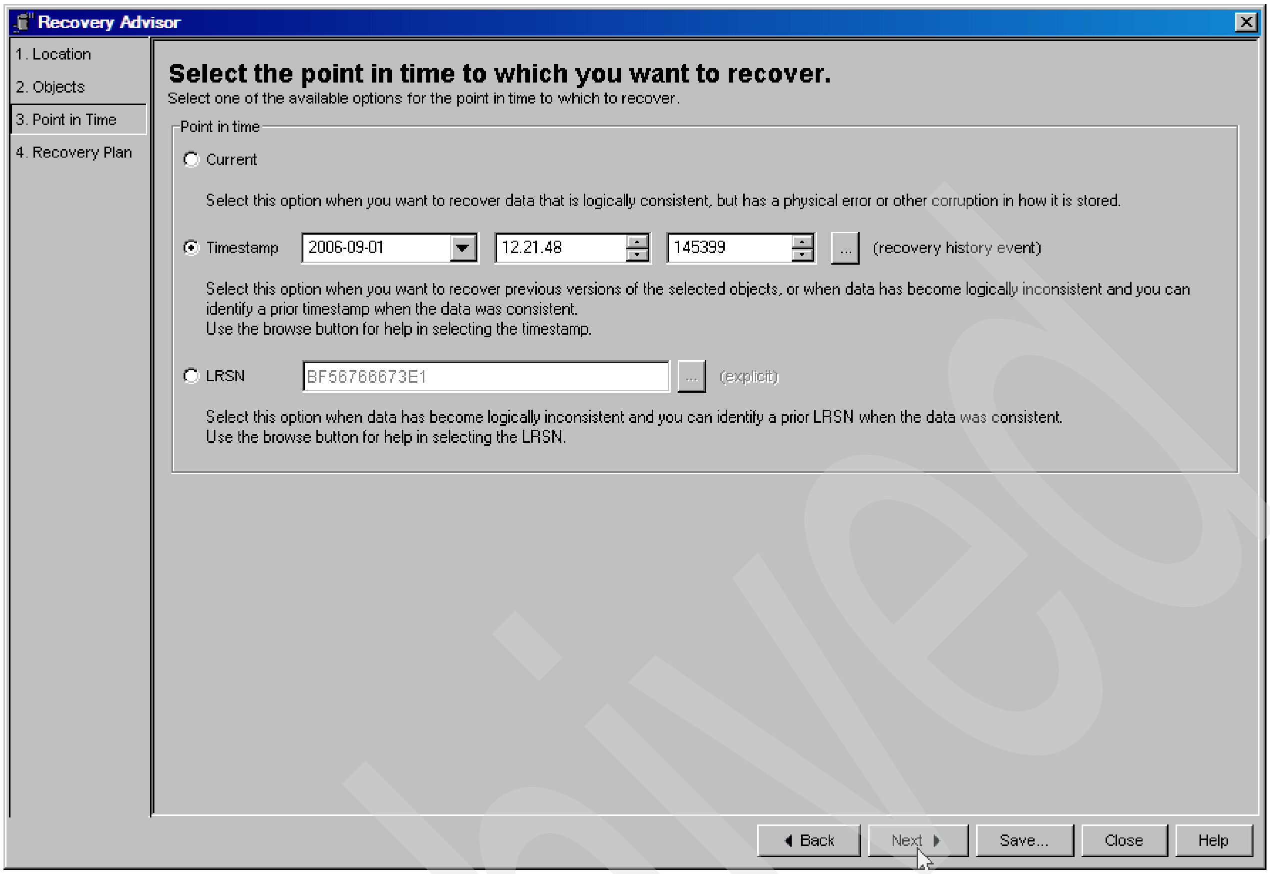

Figure 2-88 Selecting the point-in-time to recover to (3/3)

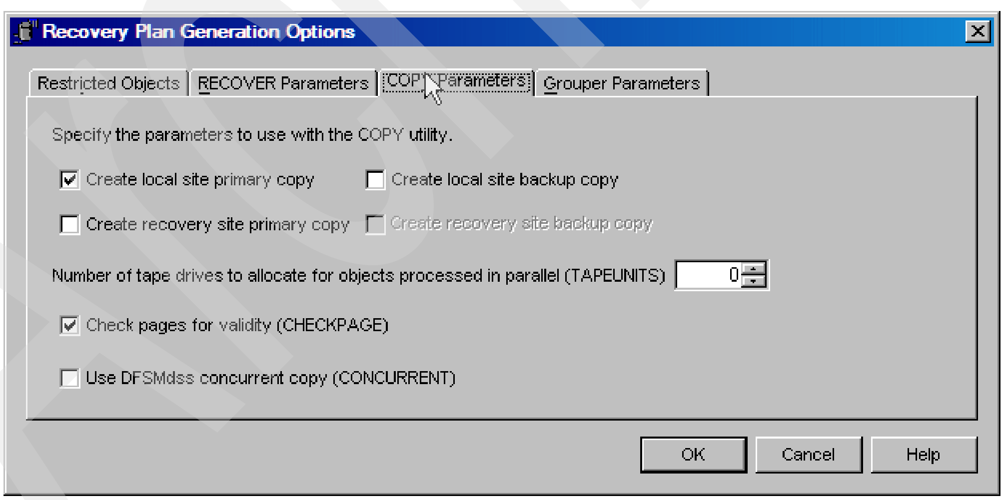

Figure 2-89 Copy parameters for recovery plan generation

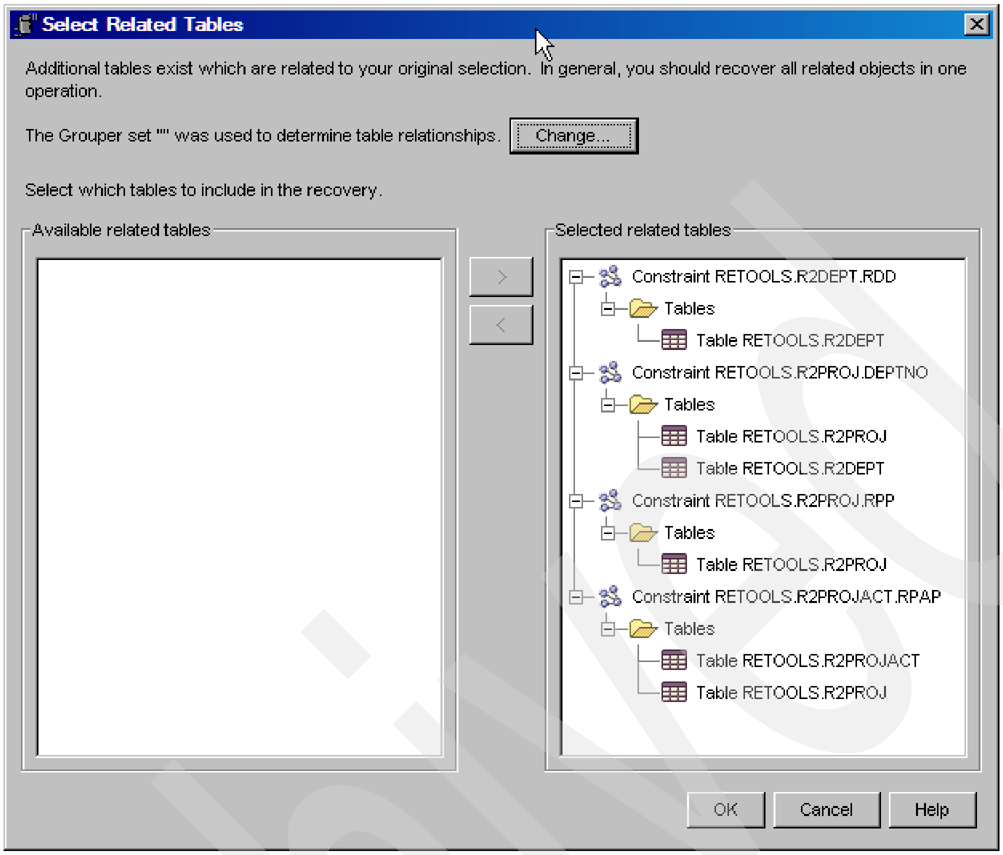

Figure 2-90 List of related tables in the referential set

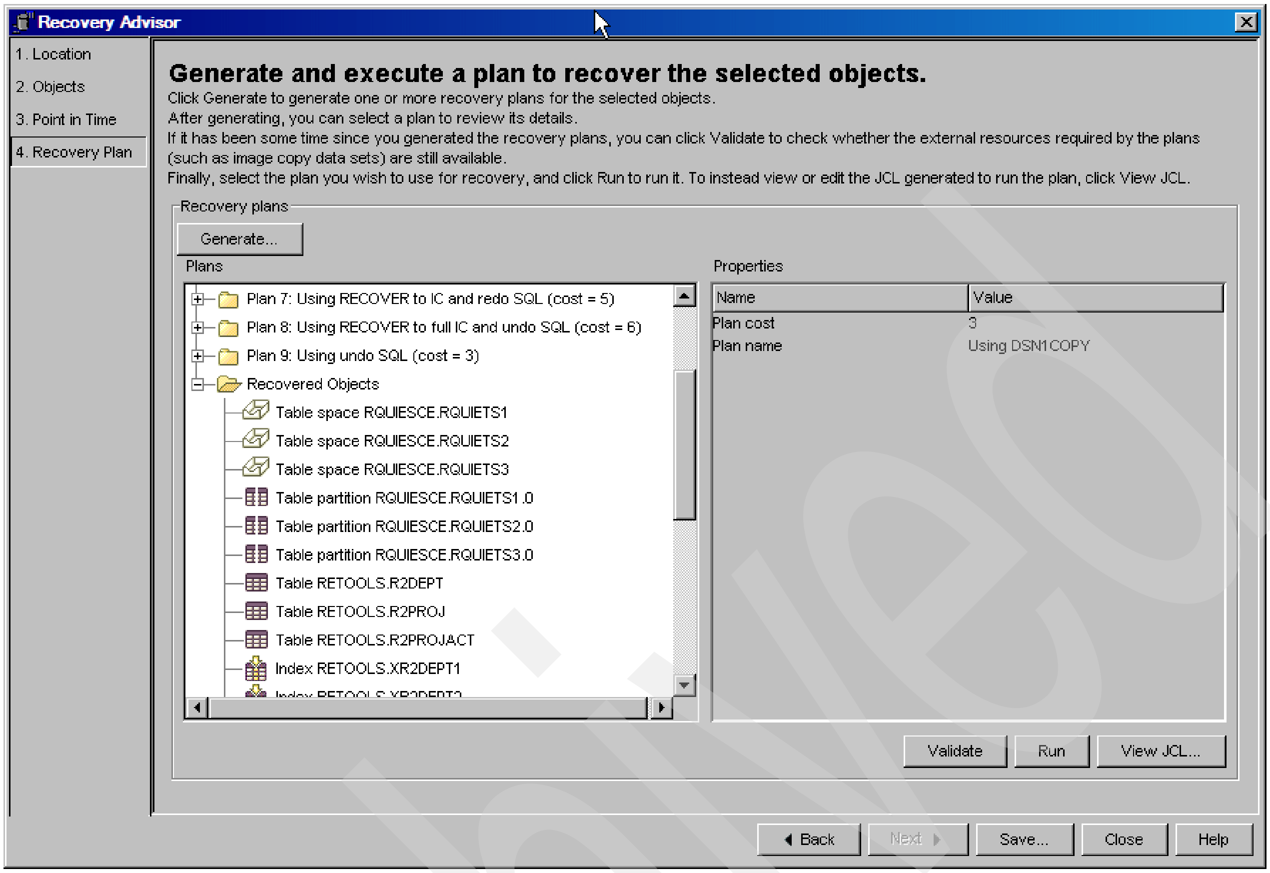

Figure 2-91 Generated recovery plan

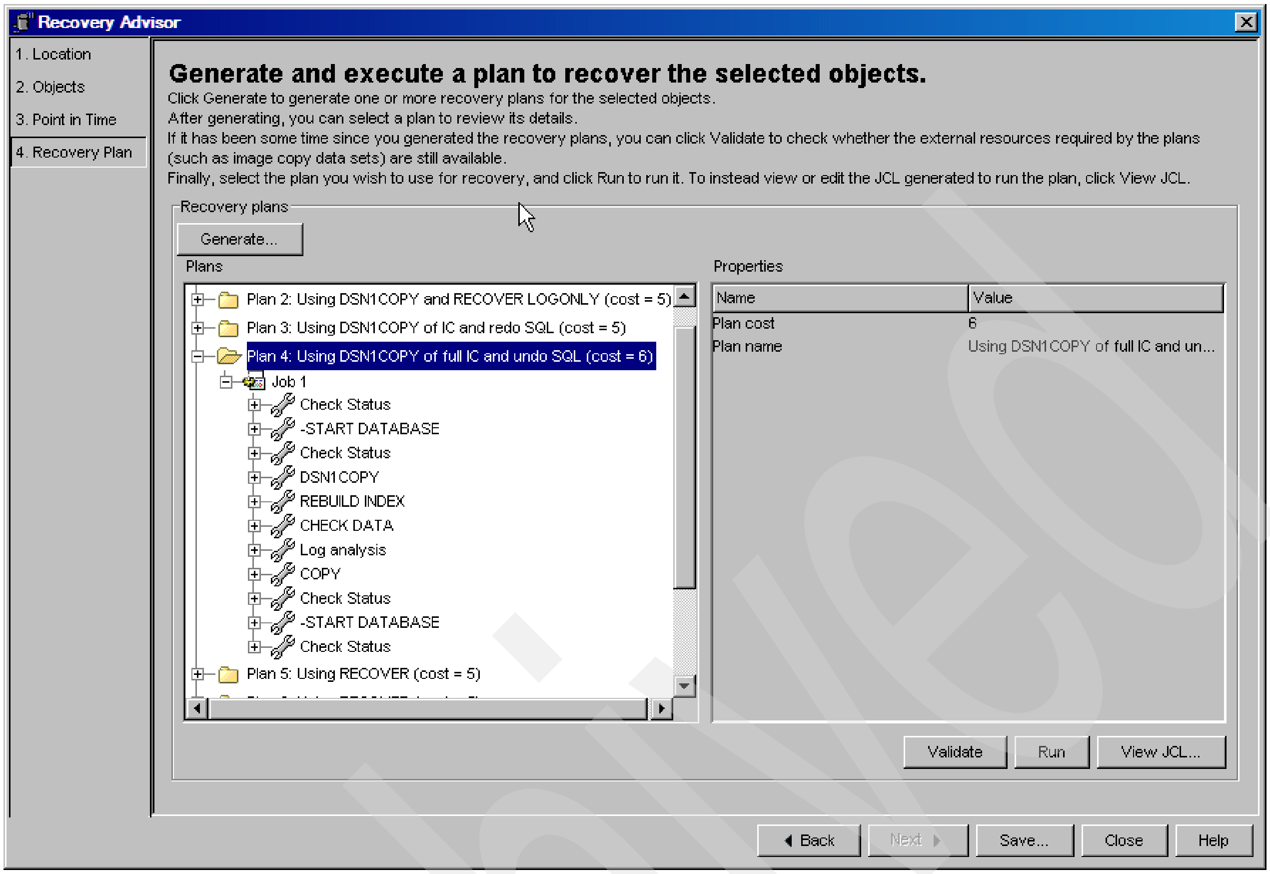

Figure 2-92 Generated recovery plans

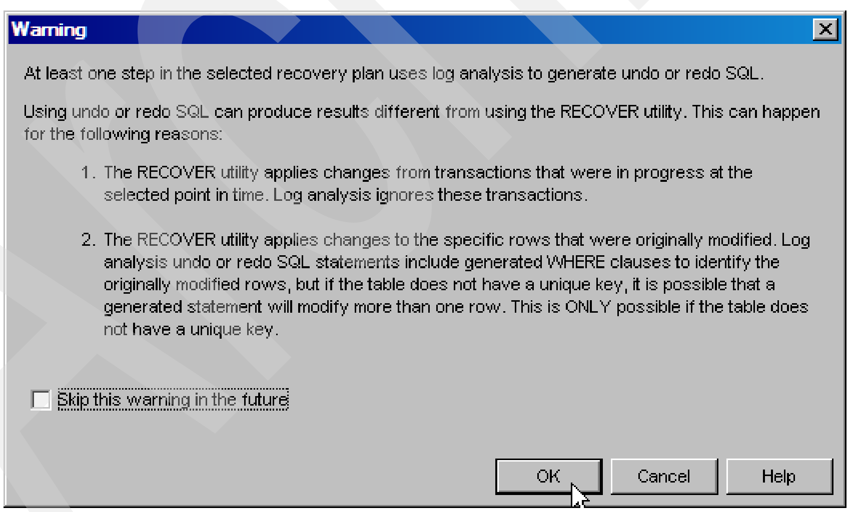

Figure 2-93 Warning message regarding log analysis step in the selected recovery plan

Example 2-17 Partial contents of the generated JCL for Plan 4

```
//GGREJCL5 JOB (999,POK),'DB2 RECOVERY EXPERT',MSGLEVEL=(1,1),CLASS=M,
//             MSGCLASS=H,REGION=4096K,NOTIFY=&SYSUID
//*  MSGCLASS=H (for EC Machine)
//*  /*ROUTE PRINT STLVM14.GPALMER (for EC Machine)
```

```
.......................
.......................
//DCPY0004 EXEC PGM=DSN1COPY,COND=(4,LT),
// PARM='CHECK,PAGESIZE(4K),FULLCOPY,RESET'
//STEPLIB  DD DISP=SHR,DSN=DB8G8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8G8.SDSNLOAD
//SYSUT1   DD DISP=SHR,DSN=REIC.D8GG.RQUIESCE.RQUIETS3.FULL.G0003V00
//SYSUT2   DD DISP=SHR,DSN=DB8GUS.DSNDBC.RQUIESCE.RQUIETS3.I0001.A001
//SYSPRINT DD   SYSOUT=*
//*
//RBLD0005 EXEC PGM=DSNUTILB,PARM=(D8G1),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8G8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8G8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//UTPRINT  DD   SYSOUT=*
//SYSOUT   DD   UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN    DD   *

  REBUILD INDEX
    ( "RETOOLS"."XR2PROJA" )
      SCOPE ALL

/*
//*
//CHKD0006 EXEC PGM=DSNUTILB,PARM=(D8G1),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8G8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8G8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//UTPRINT  DD   SYSOUT=*
//SYSUT1   DD   DSN=RETOOLS.ARY06244.S133901.M411906.SYSUT1,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(MOD,DELETE,CATLG)
//SORTOUT  DD   DSN=RETOOLS.ARY06244.S133901.M416547.SORTOUT,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(MOD,DELETE,CATLG)
//SYSERR   DD   DSN=RETOOLS.ARY06244.S133901.M418044.SYSERR,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(MOD,DELETE,CATLG)
//SYSIN    DD   *

  CHECK DATA
    TABLESPACE RQUIESCE.RQUIETS3

.........
.........
//*********************************************************************
//* STEP 2: READ THE DB2 LOG TO GENERATE THE GENERAL REPORT           *
//*********************************************************************
//LASU0002 EXEC PGM=ARYGEN1,REGION=0M,COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8G8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8G8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//MODEFILE DD   DSN=RETOOLS.ARYLAS.MODE.U0000717,
// DISP=OLD
```

```
//SYSOUT   DD   SYSOUT=*
//CFILES   DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//GENRPT   DD   SYSOUT=*
//EXTREP   DD   SYSOUT=*
//SUMREPT  DD   SYSOUT=*
//XDREPT   DD   SYSOUT=*
//QTRPT    DD   SYSOUT=*
//WARNINGS DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//DATAIN   DD   *
SSID           =D8G1
START LRSN     =BF57712BDFBC
END LRSN       =FFFFFFFFFFFF
/*
//FILTERS  DD *
GENERAL FILTERS...........
SHOW UPDATES   =Y
SHOW INSERTS   =X
SHOW DELETES   =Y
SHOW ROLLBACKS =N
CATALOG DATA   =N
SHOW UNCOMMITS =N
OBJECT FILTERS-BY NAME....
DBNAME     =RQUIESCE
TSNAME     =RQUIETS1
PARTITION  =0000
SLRSN      =BF57712BDFBC
ELRSN      =FFFFFFFFFFFF
INC/EXC    =I
DBNAME     =RQUIESCE
TSNAME     =RQUIETS2
PARTITION  =0000
SLRSN      =BF57712BDFBC
ELRSN      =FFFFFFFFFFFF
INC/EXC    =I
/*
//*********************************
//* PARAMETERS FOR JOB SHOWN BELOW *
//*********************************
//* SSID           =D8G1
//* START LRSN     =BF57712BDFBC
//* END LRSN       =FFFFFFFFFFFF
//* /*
//* //FILTERS  DD *
//* GENERAL FILTERS...........
//* SHOW UPDATES   =Y
//* SHOW INSERTS   =X
//* SHOW DELETES   =Y
//* SHOW ROLLBACKS =N
//* CATALOG DATA   =N
//* SHOW UNCOMMITS =N
//* OBJECT FILTERS-BY NAME....
//* DBNAME     =RQUIESCE
```

```
//* TSNAME     =RQUIETS1
//* PARTITION  =0000
//* SLRSN      =BF57712BDFBC
//* ELRSN      =FFFFFFFFFFFF
//* INC/EXC    =I
//* DBNAME     =RQUIESCE
//* TSNAME     =RQUIETS2
//* PARTITION  =0000
//* SLRSN      =BF57712BDFBC
//* ELRSN      =FFFFFFFFFFFF
//* INC/EXC    =I
//*
...................
...................
//LASU0006 EXEC PGM=ARYDTL4,REGION=0M,COND=(4,LT)
//*********************************************************************
//*  STEP 6: THIS STEP DOES THE FOLLOWING:
//*         -SQL GENERATION
//*********************************************************************
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8G8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8G8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//MODEFILE DD   DSN=RETOOLS.ARYLAS.MODE.U0000717,
// DISP=OLD
//SQLOUT   DD   DSN=RETOOLS.ARY06244.S133901.M467978.EXECSQL,
//         DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE),
//         DCB=(LRECL=80,BLKSIZE=23440,RECFM=FB)
//SYSOUT   DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*                                                01870000
//SYSUDUMP DD   SYSOUT=*                                                01880000
//WARNINGS DD   SYSOUT=*                                                01890000
//CFILES   DD   SYSOUT=*
//*                                                                     01920000
..........................
..........................
//LGNL0007 EXEC PGM=ARY#UTIL,PARM='EXECSQL,D8G1 ',
//             COND=((4,LT),(0,NE,LASU0002))
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8G8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8G8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
CONFIG DEFAULT
CONTINUE ON WARNING
//SYSIN    DD   DSN=RETOOLS.ARY06244.S133901.M467978.EXECSQL,
//             DISP=(OLD,KEEP)
//*
......................
......................
//COPY0008 EXEC PGM=DSNUTILB,PARM=(D8G1),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8G8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8G8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
```

```
//CPY00010 DD   DSN=RETOOLS.ARY06244.S133902.M114131.COPY,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(NEW,CATLG,DELETE)
//CPY00020 DD   DSN=RETOOLS.ARY06244.S133902.M119196.COPY,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(NEW,CATLG,DELETE)
//CPY00030 DD   DSN=RETOOLS.ARY06244.S133902.M120948.COPY,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(NEW,CATLG,DELETE)
//SYSIN    DD   *

  COPY
    TABLESPACE RQUIESCE.RQUIETS1
      FULL YES
      COPYDDN(CPY00010)
    TABLESPACE RQUIESCE.RQUIETS2
      FULL YES
      COPYDDN(CPY00020)
    TABLESPACE RQUIESCE.RQUIETS3
      FULL YES
      COPYDDN(CPY00030)
      CHECKPAGE

/*
//*
..................
..................
```

2.4.8 Recovering a table space in referential set to QT at current DDL version

This scenario describes the recovery of a single table space RITSTTS2, which is part of a referential set, to a quiet time that corresponds to a moment after the last DDL change that occurred on any object in that table space, that is, at the current DDL version of all the objects in the table space. In this scenario, we assume the timeline that corresponds, in general, to the type shown in Figure 2-50 on page 104. This environment is a data sharing environment.

The following windows only represent the ones of particular interest (in our opinion) to this scenario. The windows leading up to these windows and those that follow these windows are almost identical to the windows described in 2.3.1, “Recovering multi-table table space” on page 56. We show the partial contents of the generated JCL for a given recovery plan, but not the execution job results.

Figure 2-94 on page 156 shows the single table space RITSTTS2 that is part of a referential set selected for recovery in the navigation pane. Figure 2-95 on page 157 through Figure 2-101 on page 161 show the execution of log analysis to determine quiet times for the objects of interest (RITSTTS2[5]); not all window interactions are shown here. Figure 2-102 on page 162 shows how the selected quiet time entry in Figure 2-101 on page 161 is translated to a corresponding timestamp. Figure 2-103 on page 162 shows the COPY parameters for recovery plan generation. Figure 2-104 on page 163 shows Recovery Expert for z/OS identifying all the tables in the referential set for the selected table space RITSTTS2 that participates in the recovery. This is a most useful facility that eliminates user error that tends to occur when referential sets are involved in point-in-time recoveries. Figure 2-105 on page 164 shows five recovery plans being generated for the requested recovery action, the

[5] The other table spaces in the referential set are not automatically included in the list.

recovered objects, and the expanded steps included in the recovery plan, Plan 5. Figure 2-106 on page 164 shows the warning issued when log analysis is included in the selected Plan 5. A portion of the generated JCL for Plan 5 is shown in Example 2-18 on page 164.

The selected recovery plan, Plan 5, has multiple steps that include the generation of undo SQL to be run on all the members of the referential set to correspond to the selected quiet time recovery point. Therefore, the log analysis is bounded by the end of the log (LRSN FFFFFFFFFFFF) and the starting LRSN (BF56591F3E74) corresponding to the quiet time restored to, as shown in Figure 2-101 on page 161. The final step is to copy all the table spaces in the referential set as specified in the COPY parameters for recovery plan generation, as shown in Figure 2-103 on page 162.

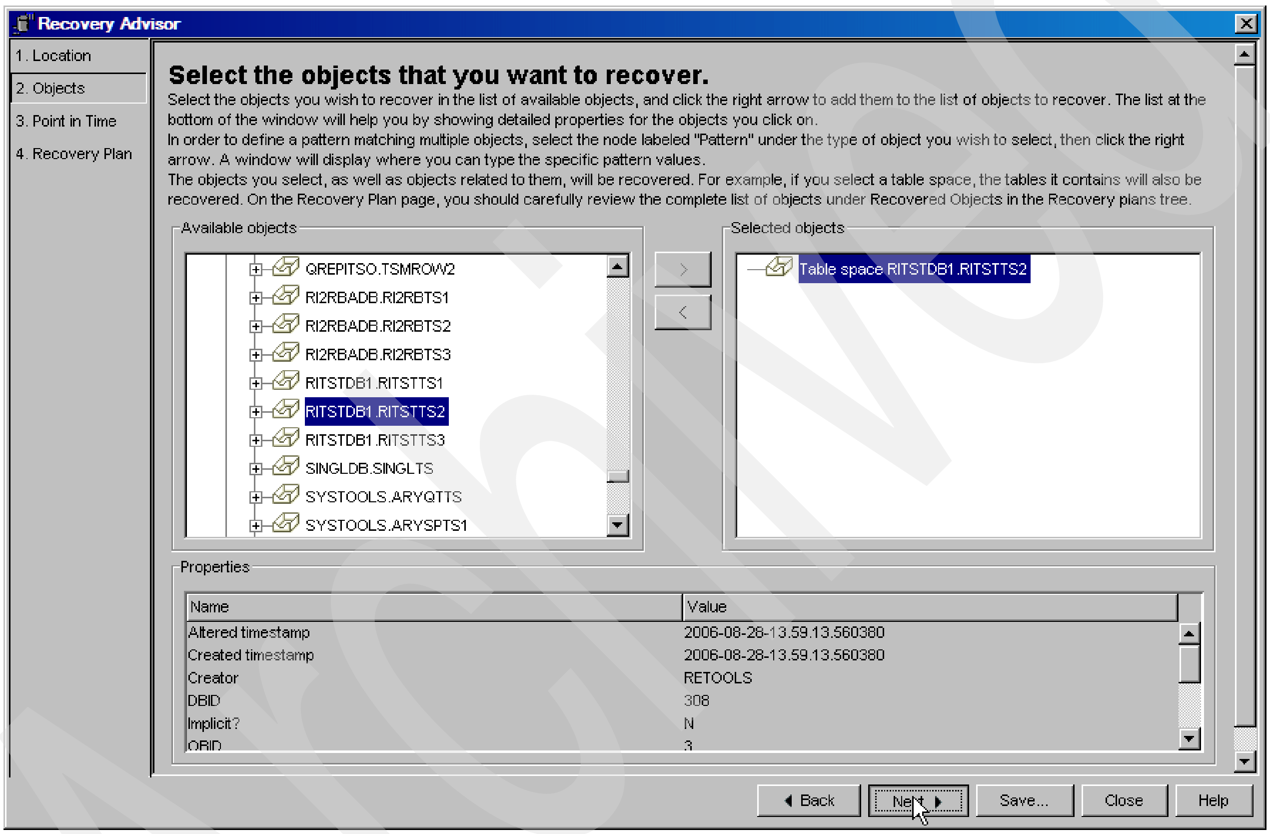

Figure 2-94 Selecting the table space in a referential set to recover

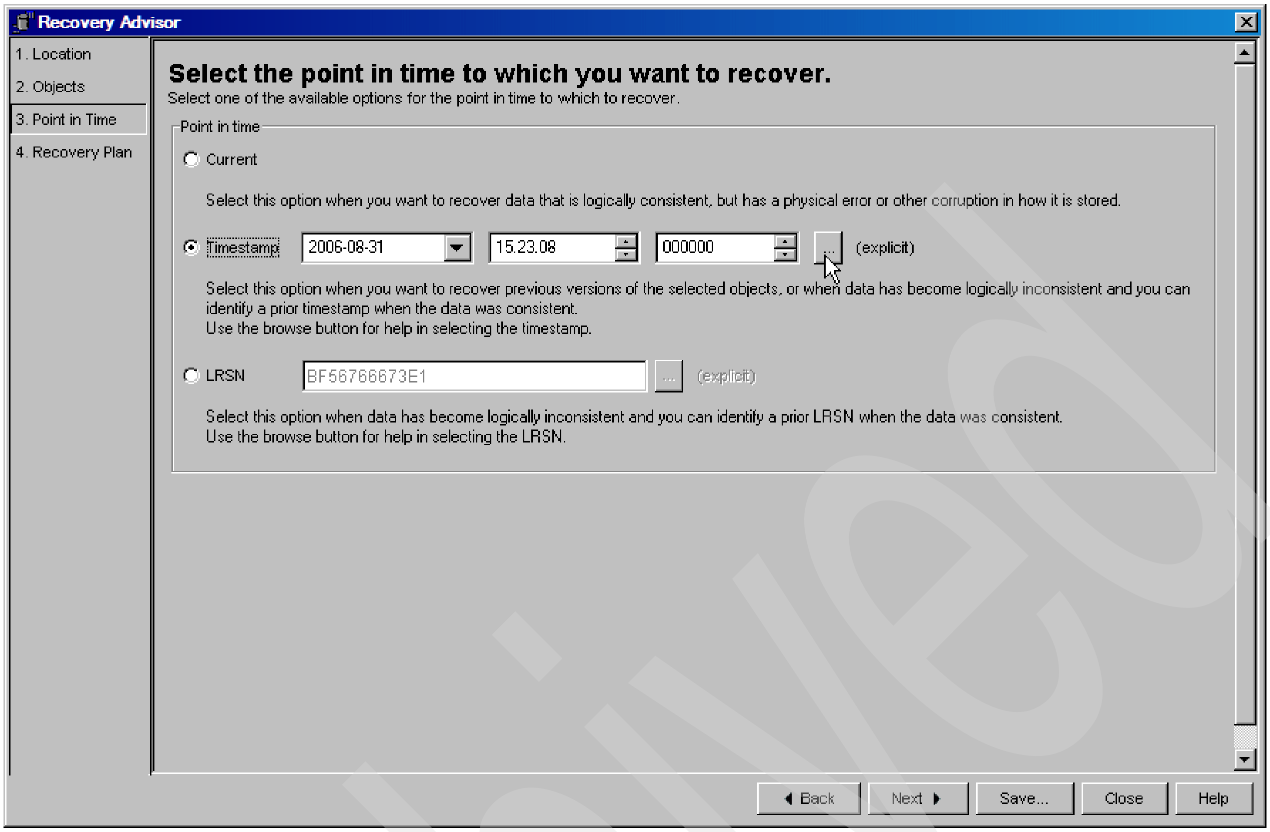

Figure 2-95 Selecting the point to recover to

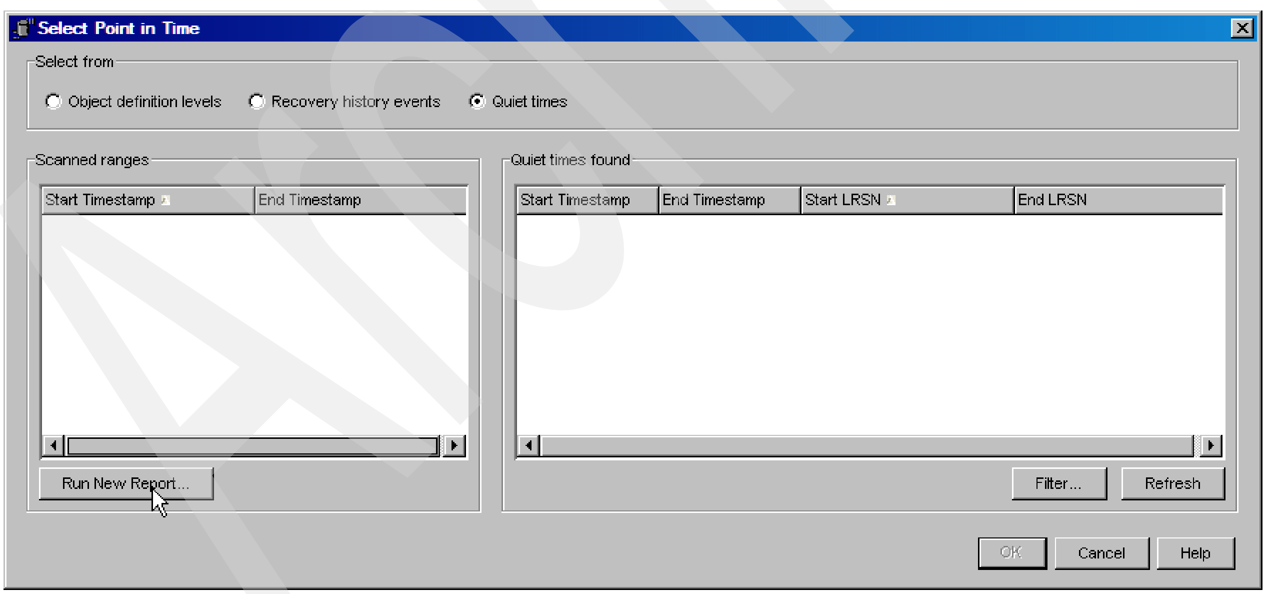

Figure 2-96 Selecting quiet times

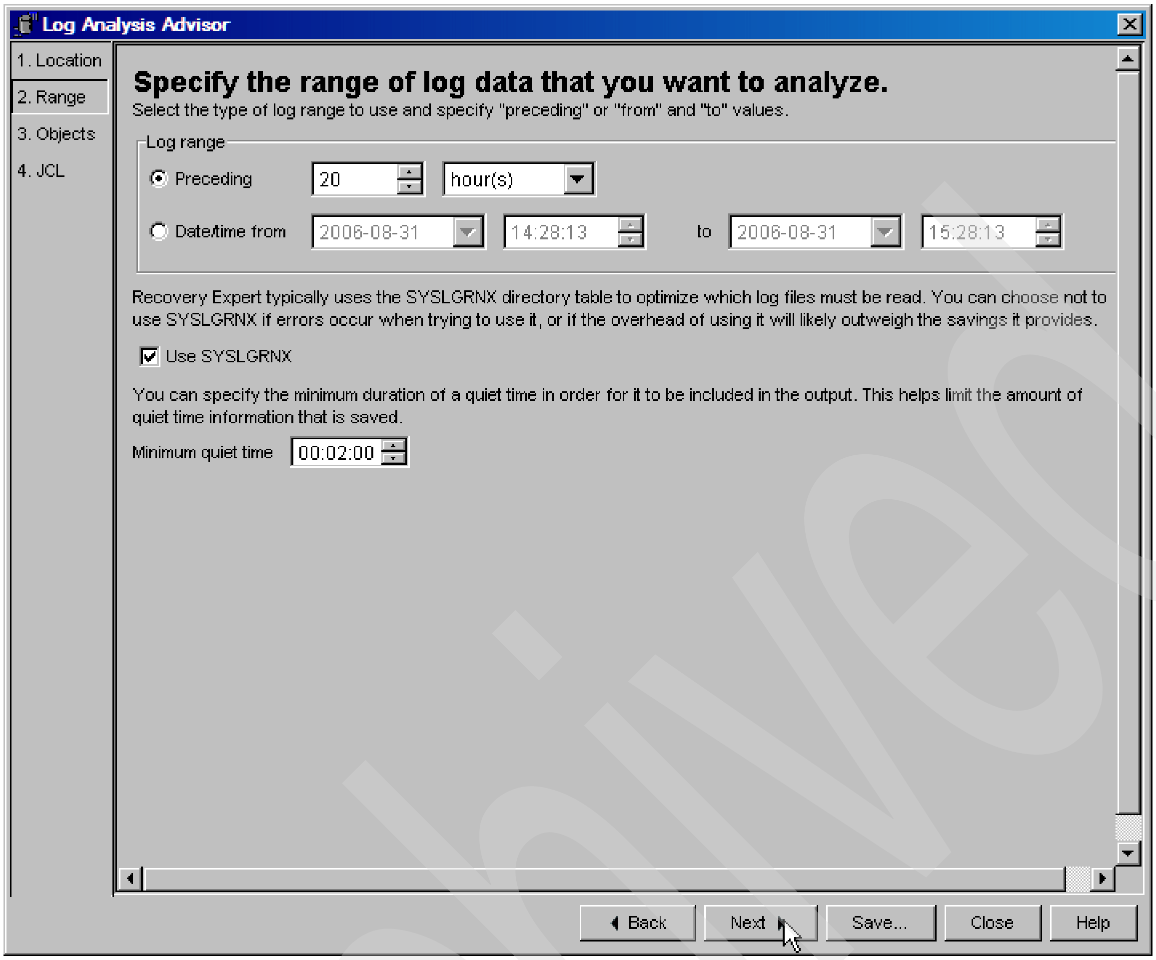

Figure 2-97 Selecting the time range in which to look for quiet times

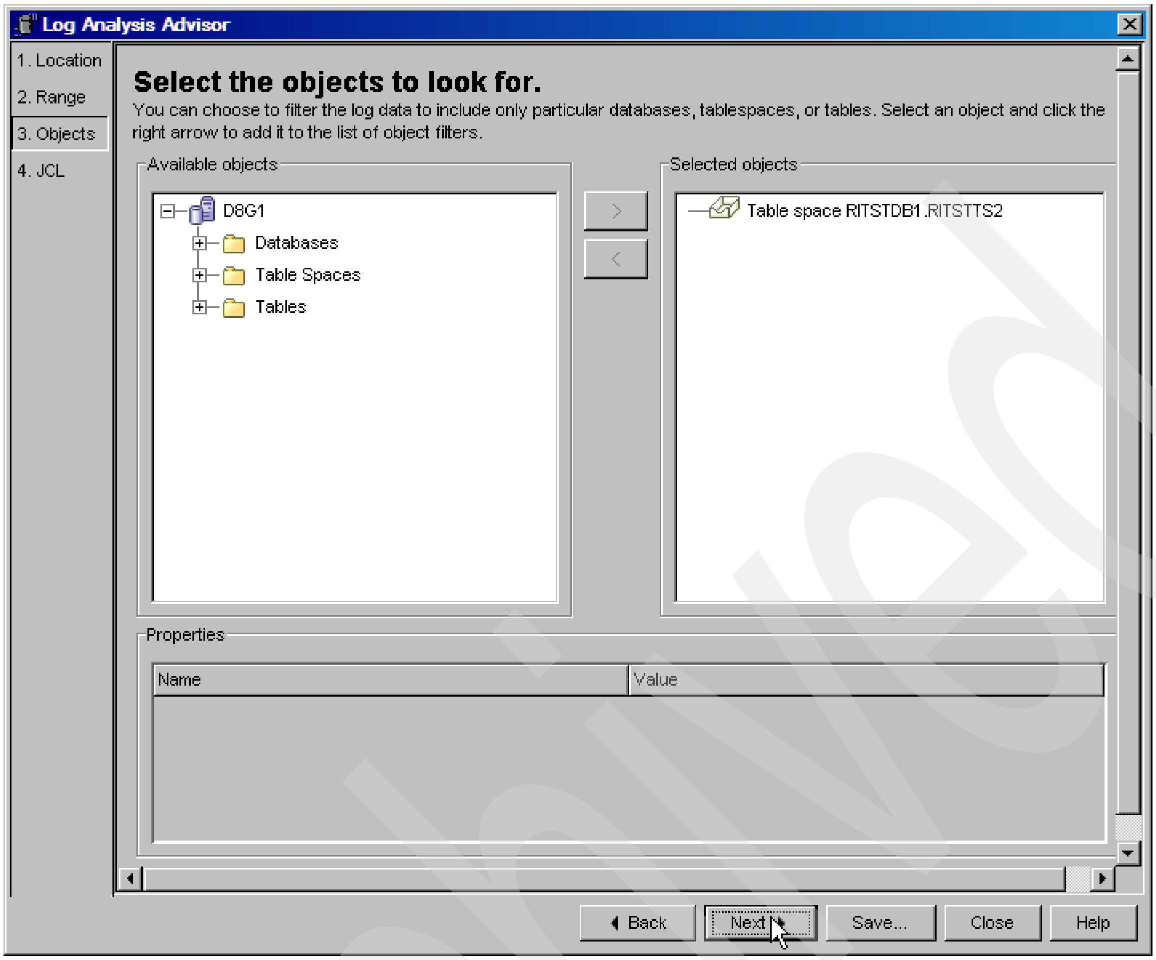

Figure 2-98 Identifying the quiet time for objects of interest

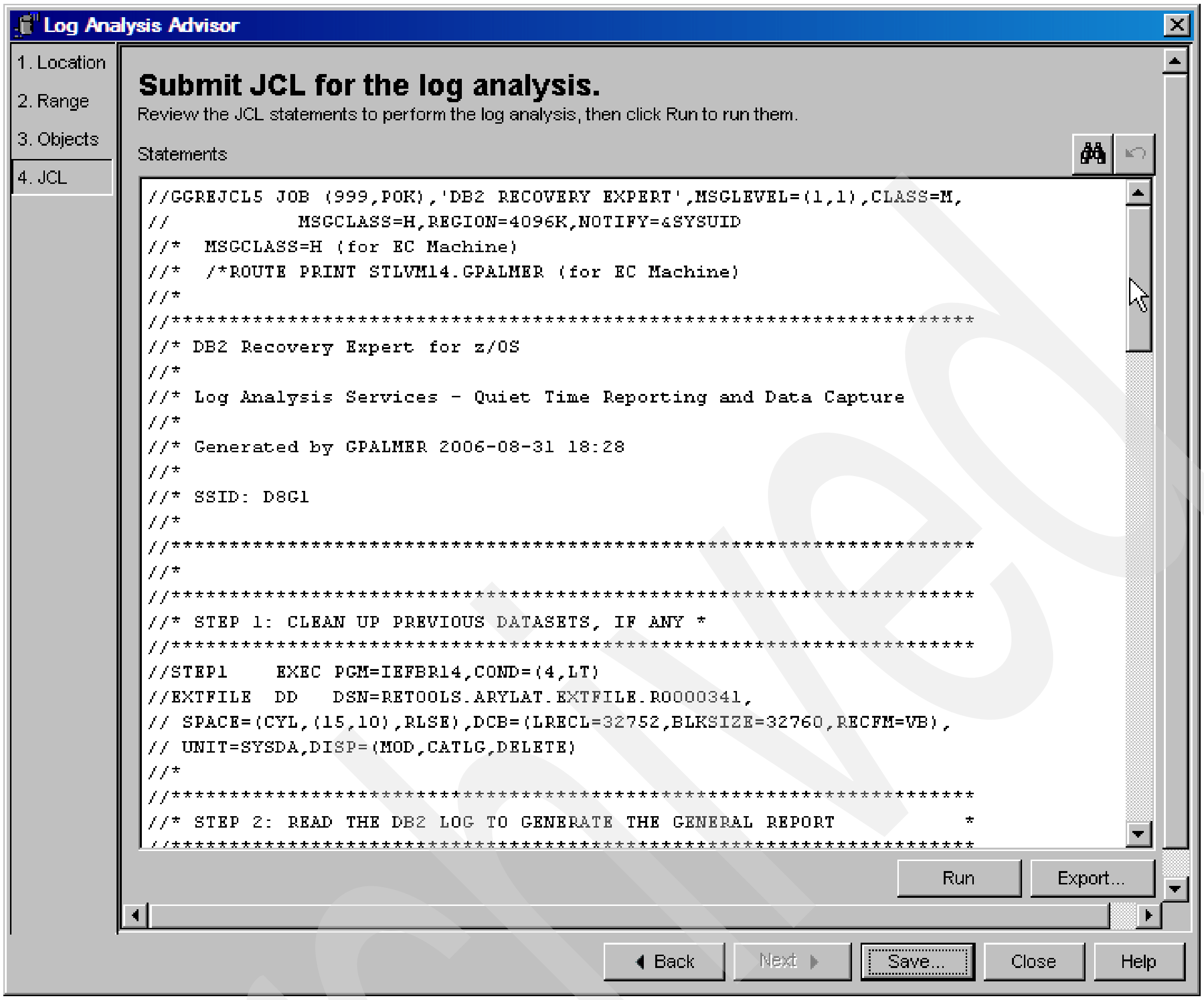

Figure 2-99   JCL to run the log analysis to ascertain quiet times

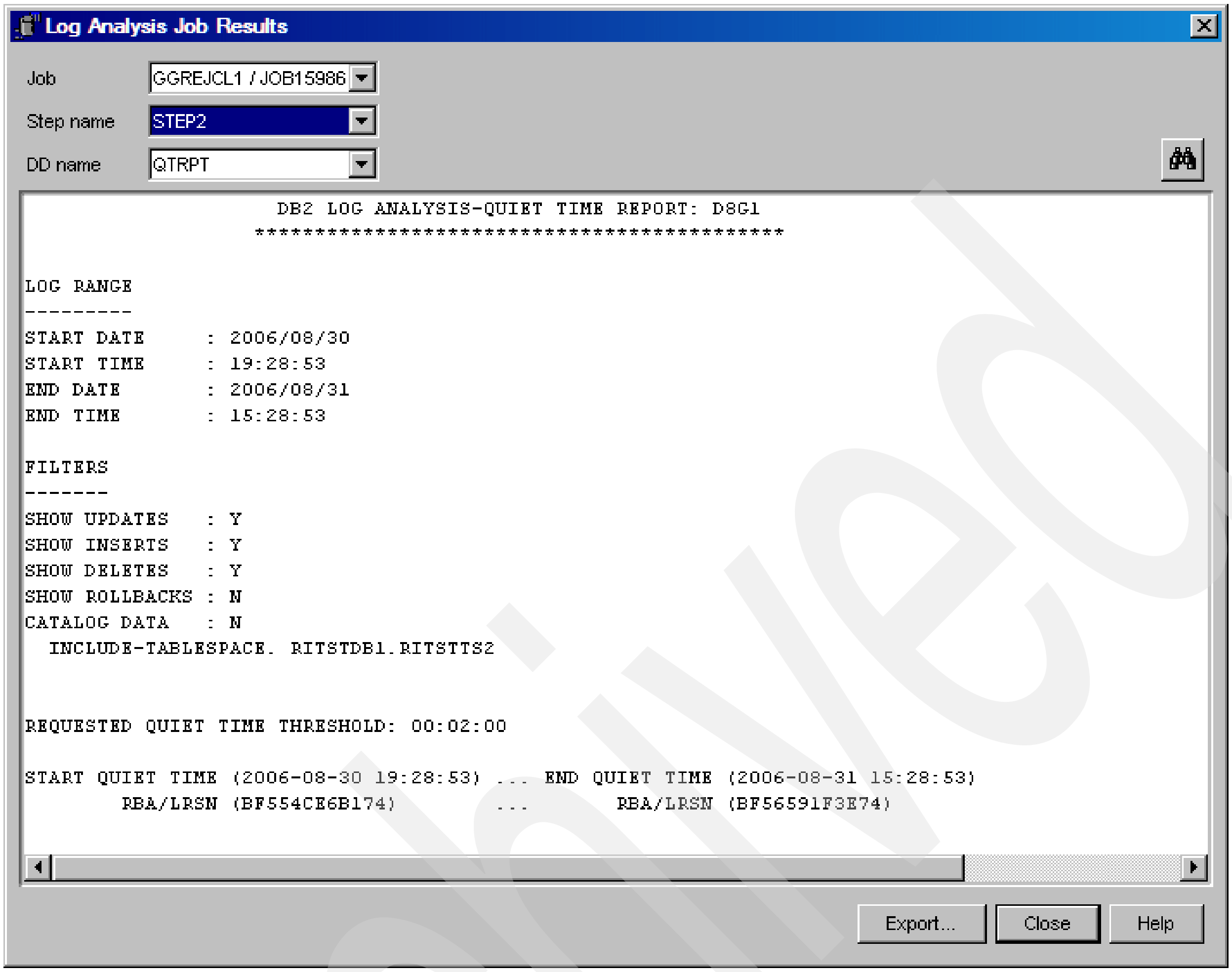

Figure 2-100 Log analysis output

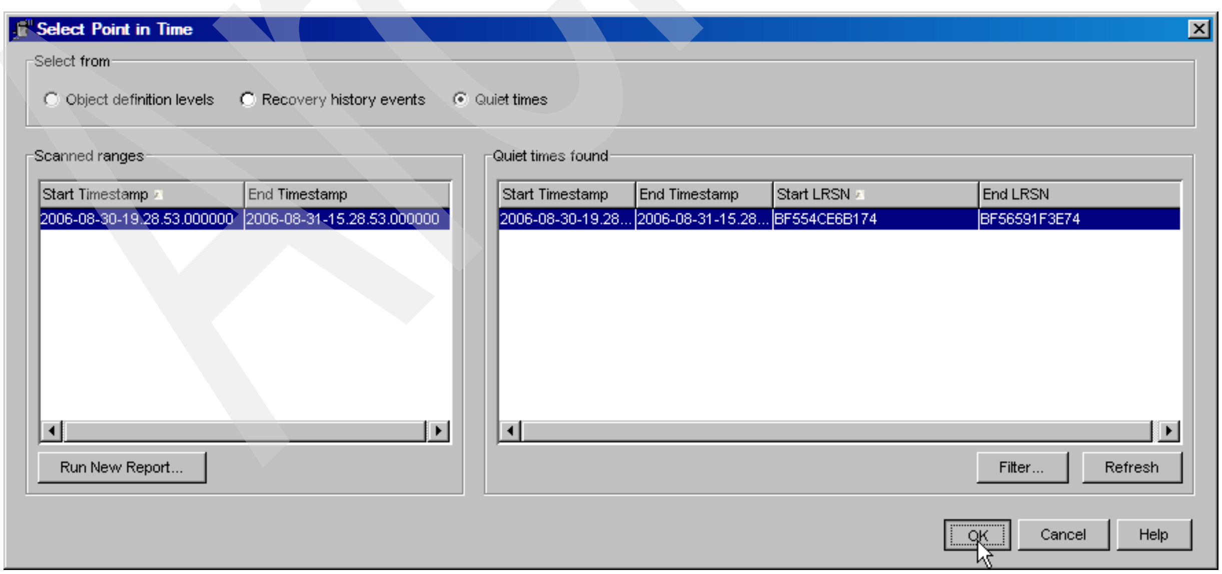

Figure 2-101 Quiet times detected

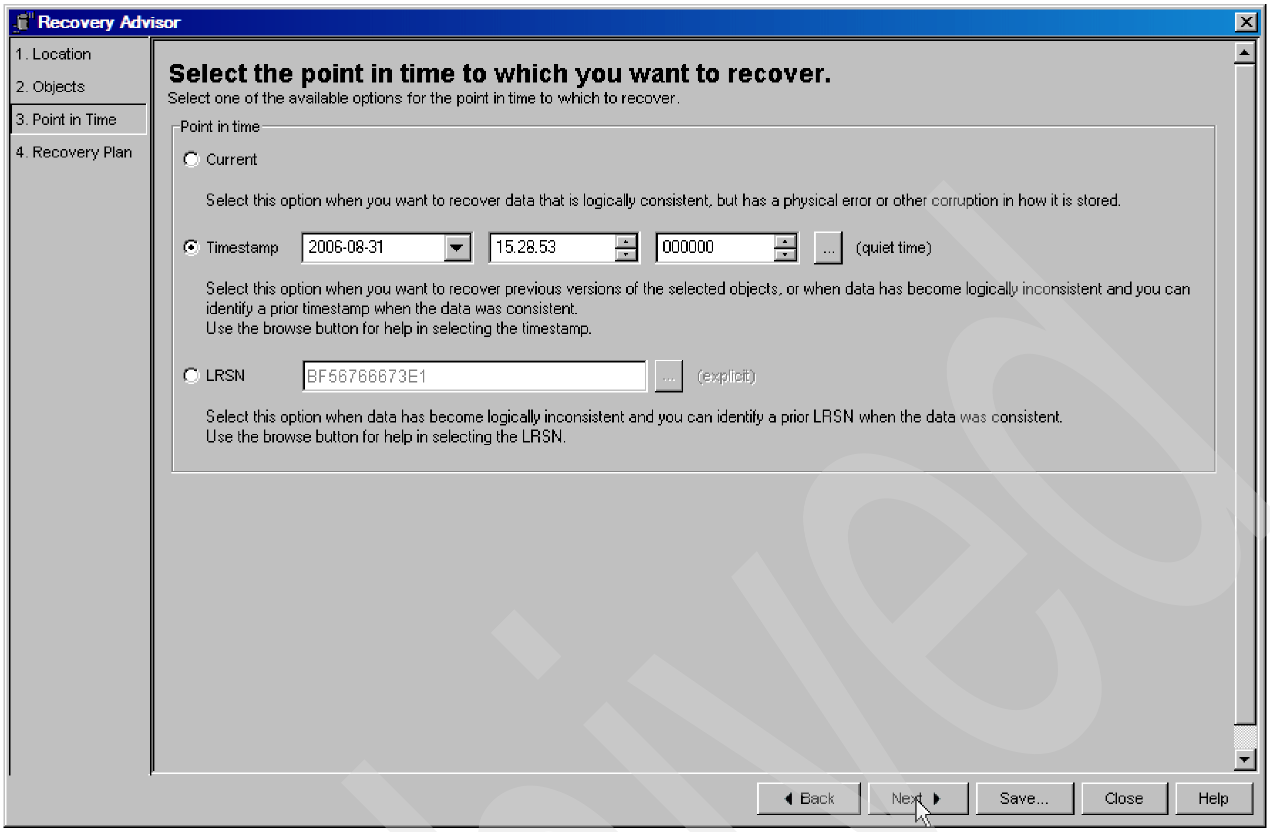

Figure 2-102 Timestamp to recover to

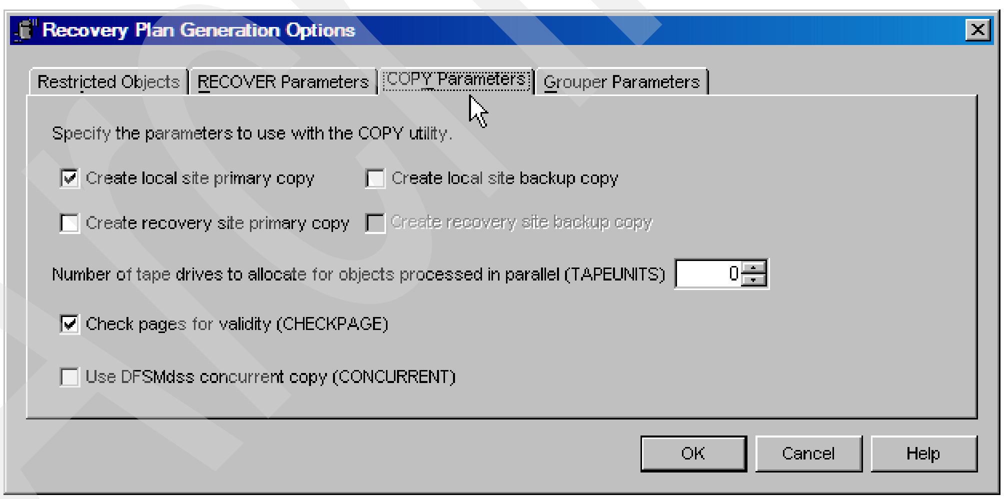

Figure 2-103 Copy parameters for recovery plan generation

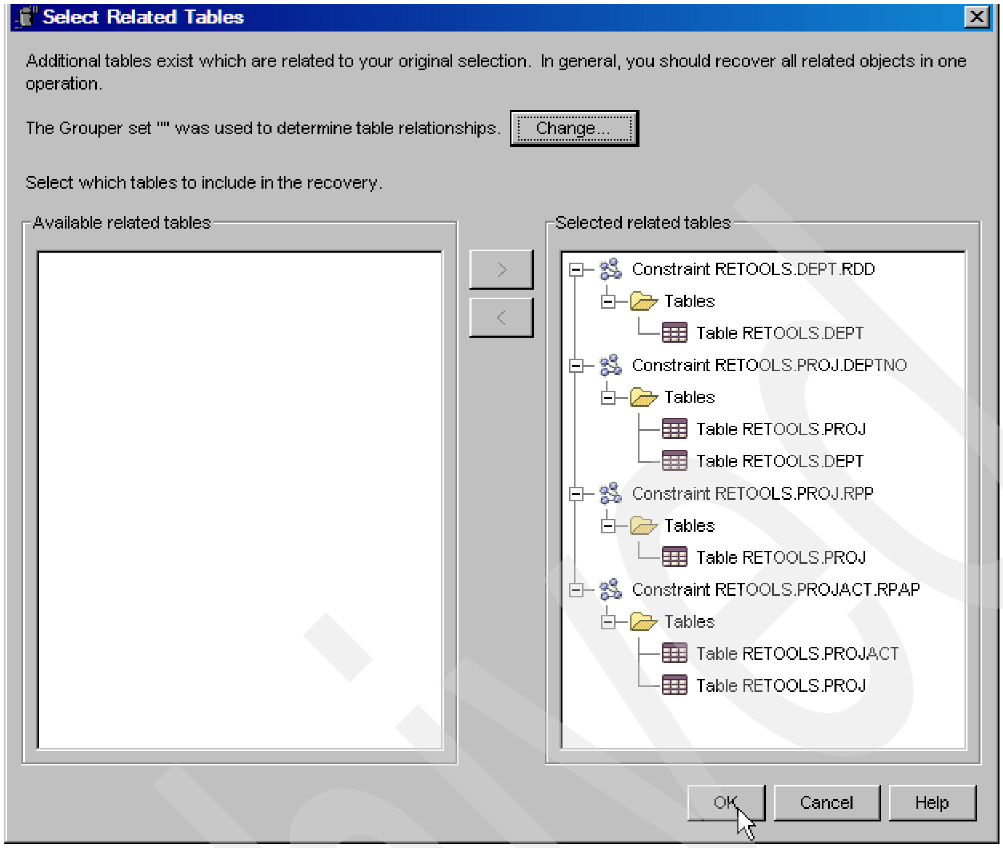

Figure 2-104 Related tables through referential constraints

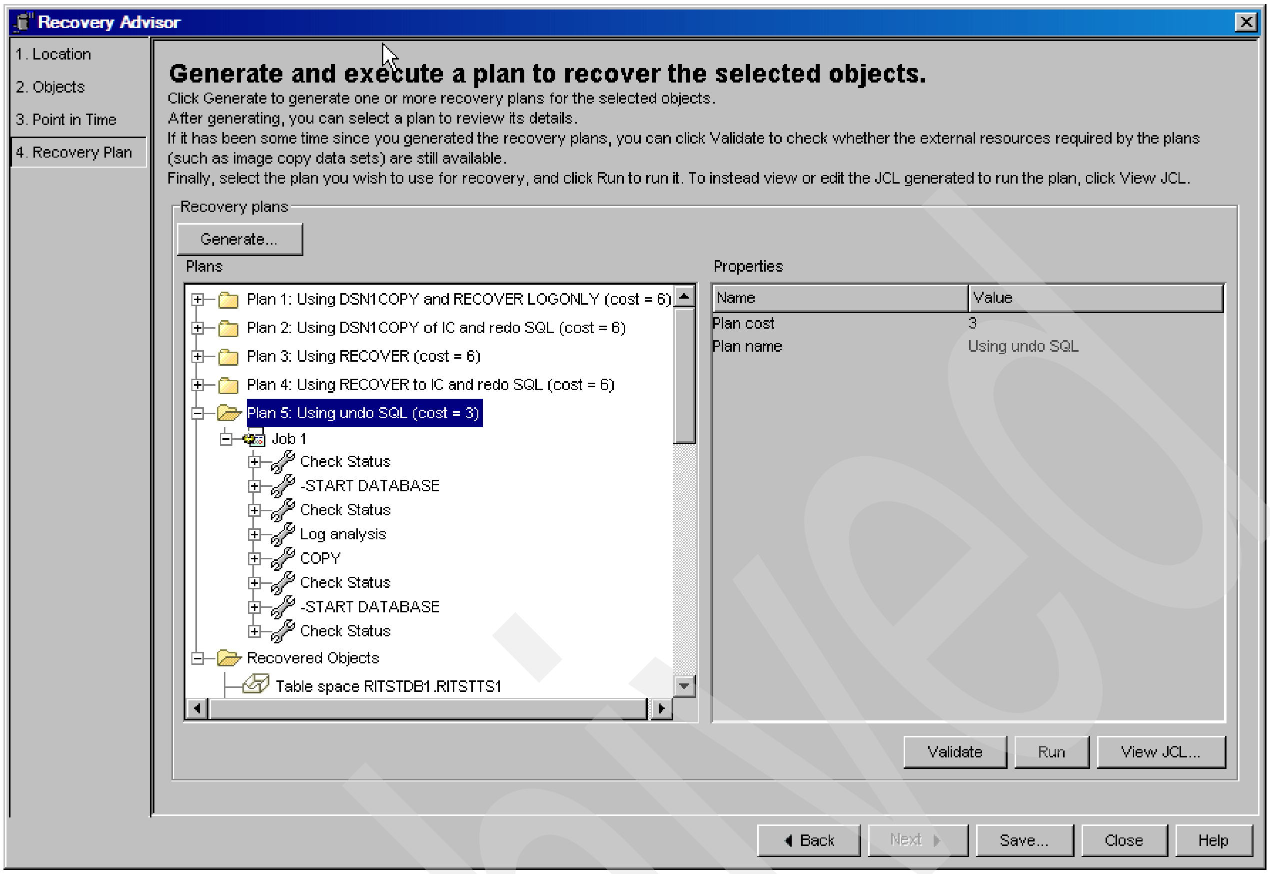

Figure 2-105   Generated recovery plans

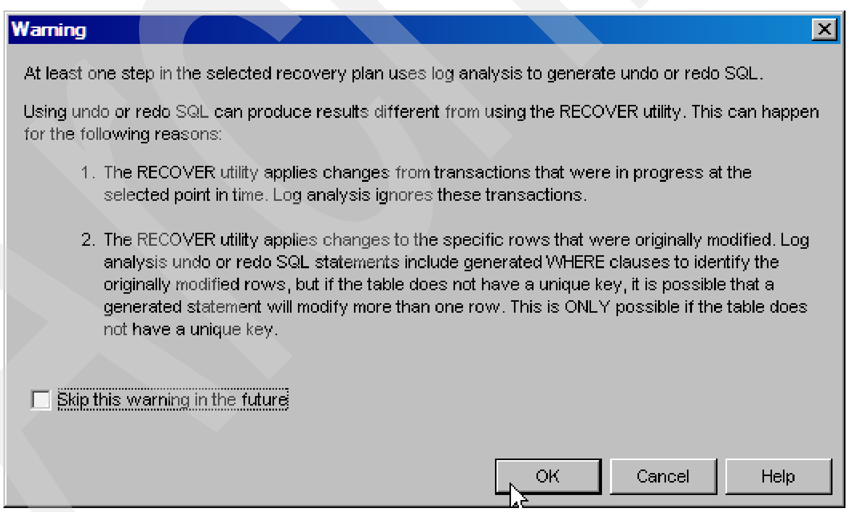

Figure 2-106   Warning message regarding log analysis step in the selected recovery plan

Example 2-18   Partial contents of the generated JCL for Plan 5

```
//GGREJCL5 JOB (999,POK),'DB2 RECOVERY EXPERT',MSGLEVEL=(1,1),CLASS=M,
//            MSGCLASS=H,REGION=4096K,NOTIFY=&SYSUID
//*  MSGCLASS=H (for EC Machine)
//*  /*ROUTE PRINT STLVM14.GPALMER (for EC Machine)
//*
```

```
......................
......................
//*******************************************************************
//* STEP 2: READ THE DB2 LOG TO GENERATE THE GENERAL REPORT          *
//*******************************************************************
//LASU0002 EXEC PGM=ARYGEN1,REGION=0M,COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8G8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8G8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//MODEFILE DD   DSN=RETOOLS.ARYLAS.MODE.U0000426,
// DISP=OLD
//SYSOUT   DD   SYSOUT=*
//CFILES   DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//GENRPT   DD   SYSOUT=*
//EXTREP   DD   SYSOUT=*
//SUMREPT  DD   SYSOUT=*
//XDREPT   DD   SYSOUT=*
//QTRPT    DD   SYSOUT=*
//WARNINGS DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//DATAIN   DD   *
SSID           =D8G1
START LRSN     =BF56591F3E74
END LRSN       =FFFFFFFFFFFF
/*
//FILTERS  DD *
GENERAL FILTERS...........
SHOW UPDATES   =Y
SHOW INSERTS   =X
SHOW DELETES   =Y
SHOW ROLLBACKS =N
CATALOG DATA   =N
SHOW UNCOMMITS =N
OBJECT FILTERS-BY NAME....
DBNAME     =RITSTDB1
TSNAME     =RITSTTS1
PARTITION  =0000
SLRSN      =BF56591F3E74
ELRSN      =FFFFFFFFFFFF
INC/EXC    =I
DBNAME     =RITSTDB1
TSNAME     =RITSTTS2
PARTITION  =0000
SLRSN      =BF56591F3E74
ELRSN      =FFFFFFFFFFFF
INC/EXC    =I
DBNAME     =RITSTDB1
TSNAME     =RITSTTS3
PARTITION  =0000
SLRSN      =BF56591F3E74
ELRSN      =FFFFFFFFFFFF
INC/EXC    =I
```

```
/*
//*********************************
//* PARAMETERS FOR JOB SHOWN BELOW *
//*********************************
//* SSID           =D8G1
//* START LRSN     =BF56591F3E74
//* END LRSN       =FFFFFFFFFFFF
//* /*
//* //FILTERS  DD *
//* GENERAL FILTERS...........
//* SHOW UPDATES   =Y
//* SHOW INSERTS   =X
//* SHOW DELETES   =Y
//* SHOW ROLLBACKS =N
//* CATALOG DATA   =N
//* SHOW UNCOMMITS =N
//* OBJECT FILTERS-BY NAME....
//* DBNAME     =RITSTDB1
//* TSNAME     =RITSTTS1
//* PARTITION  =0000
//* SLRSN      =BF56591F3E74
//* ELRSN      =FFFFFFFFFFFF
//* INC/EXC    =I
//* DBNAME     =RITSTDB1
//* TSNAME     =RITSTTS2
//* PARTITION  =0000
//* SLRSN      =BF56591F3E74
//* ELRSN      =FFFFFFFFFFFF
//* INC/EXC    =I
//* DBNAME     =RITSTDB1
//* TSNAME     =RITSTTS3
//* PARTITION  =0000
//* SLRSN      =BF56591F3E74
//* ELRSN      =FFFFFFFFFFFF
//* INC/EXC    =I
//*
................
................
//LASU0006 EXEC PGM=ARYDTL4,REGION=0M,COND=(4,LT)
//*******************************************************************
//*  STEP 6: THIS STEP DOES THE FOLLOWING:
//*        -SQL GENERATION
//*******************************************************************
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8G8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8G8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//MODEFILE DD   DSN=RETOOLS.ARYLAS.MODE.U0000426,
// DISP=OLD
//SQLOUT   DD   DSN=RETOOLS.ARY06243.S185335.M208816.EXECSQL,
//         DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE),
//         DCB=(LRECL=80,BLKSIZE=23440,RECFM=FB)
//SYSOUT   DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*                                              01870000
//SYSUDUMP DD   SYSOUT=*                                              01880000
```

```
//WARNINGS DD   SYSOUT=*                                                01890000
//CFILES   DD   SYSOUT=*
//*                                                                     01920000
//*
.................
.................
//*
//LGNL0004 EXEC PGM=ARY#UTIL,PARM='EXECSQL,D8G1 ',
//              COND=((4,LT),(0,NE,LASU0002))
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8G8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8G8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
CONFIG DEFAULT
CONTINUE ON WARNING
//SYSIN    DD   DSN=RETOOLS.ARY06243.S185335.M208816.EXECSQL,
//              DISP=(OLD,KEEP)
//*
..................
..................
//*
//COPY0005 EXEC PGM=DSNUTILB,PARM=(D8G1),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8G8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8G8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//CPY00010 DD   DSN=RETOOLS.ARY06243.S185335.M763605.COPY,
//              UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//              DISP=(NEW,CATLG,DELETE)
//CPY00020 DD   DSN=RETOOLS.ARY06243.S185335.M768283.COPY,
//              UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//              DISP=(NEW,CATLG,DELETE)
//CPY00030 DD   DSN=RETOOLS.ARY06243.S185335.M769688.COPY,
//              UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//              DISP=(NEW,CATLG,DELETE)
//SYSIN    DD   *

  COPY
    TABLESPACE RITSTDB1.RITSTTS1
      FULL YES
      COPYDDN(CPY00010)
    TABLESPACE RITSTDB1.RITSTTS2
      FULL YES
      COPYDDN(CPY00020)
    TABLESPACE RITSTDB1.RITSTTS3
      FULL YES
      COPYDDN(CPY00030)
      CHECKPAGE

/*
//*
..................
..................
```

2.5 Recovering dropped objects

When an object is accidentally dropped, typically you try to restore it back to the point just before it was dropped, or to an image copy before the drop (the only option available in the base DB2 for z/OS engine to the DBA, using handcrafted DSN1COPY and OBID translate jobs). DB2 for z/OS does not provide the required facilities to recover dropped objects. However, Recovery Expert for z/OS provides capabilities to not only recover dropped objects to the point just before they were dropped, but also to other points such as an image copy, RBA, or even an object definition level. This section describes a number of potential recovery scenarios for a dropped object.

As far as possible, we have used a common timeline showing a sequence of events in all the scenarios. The events in the timeline are not the most optimal of operations, but we have selected them to demonstrate the recovery semantics of Recovery Expert for z/OS.

The recover to point-in-time scenarios for a dropped object described in this section are:

- Recovering table in single-table table space to object definition level (ODL) at current DDL version
- Recovering table in single-table table space to IC at current DDL version
- Recovering table in single-table table space to IC before current DDL version
- Recovering table in single-table table space to ODL before current DDL version

2.5.1 Recovering table in single-table table space to ODL at current DDL version

This scenario describes the recovery of a table in a single-table table space to an object definition level that corresponds to the current DDL version just before it was dropped. This corresponds to recovering the object to the way it was just before the drop. The timeline shown in Figure 2-107 on page 169 is assumed for this scenario. Perform an SLR update (time t14) after the drop (time t13) in order for an end time to be recorded in the object definition level, before you attempt a recovery of the dropped table. This environment is a non-data sharing environment.

The following windows only represent the ones of particular interest (in our opinion) to this scenario. The windows leading up to these windows and those that follow these windows are almost identical to the windows described in 2.3.1, “Recovering multi-table table space” on page 56. We show the partial contents of the generated JCL for a given recovery plan, but not the execution job results.

Figure 2-108 on page 170 shows the dropped table RETOOLS.SINGLE selected for recovery in the navigation pane. Its red color in the navigation pane confirms that this table has been dropped. Figure 2-109 on page 171 through Figure 2-111 on page 172 show the selection of an object definition level corresponding to time t11 in Figure 2-107 on page 169 from a list of available object definition level entries from the SLR tables; not all window interactions are shown here. Figure 2-111 on page 172 shows this object definition level translated to a timestamp corresponding to the selected object definition level entry’s “Level End Timestamp” as shown in Figure 2-110 on page 171. Figure 2-112 on page 172 shows a warning message issued by Recovery Expert for z/OS that the selected timestamp might not be a point-of-consistency. Figure 2-113 on page 173 shows a single recovery plan generated for the requested recovery action. Figure 2-114 on page 173 shows a warning that the selected recovery plan, Plan 1, has a log analysis step in it. A portion of the generated JCL for Plan 1 is shown in Example 2-19 on page 174.

The selected recovery plan, Plan 1, has multiple steps that include the creation of the dropped table and its dependencies (such as indexes and views, if any), followed by a DSN1COPY restore of the latest image copy with an OBID translate to the new OBID of the object. This is followed by a rebuilding of the indexes, and a log analysis that has a boundary interval corresponding to a start RBA of 00001481695A, which corresponds to the “Level Create Timestamp” and an end RBA of 000014855441, which corresponds to the “Level End Timestamp” for the object definition level entry that is selected in Figure 2-110 on page 171. The generated SQL is then applied to complete the recovery action.

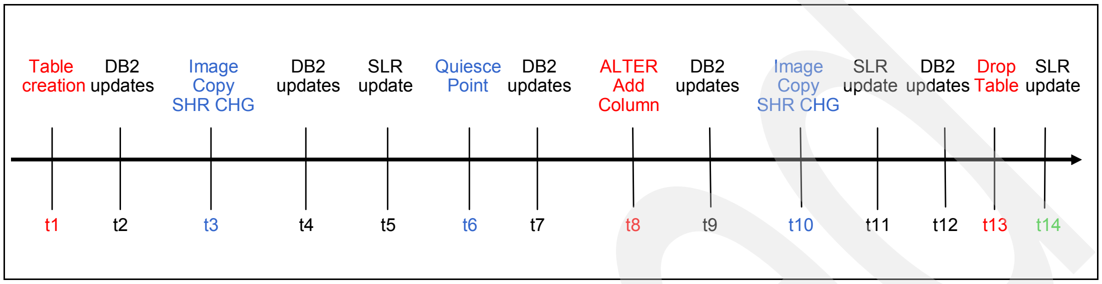

Figure 2-107 Recovering dropped table in single-table table space to ODL at current DDL scenario timeline

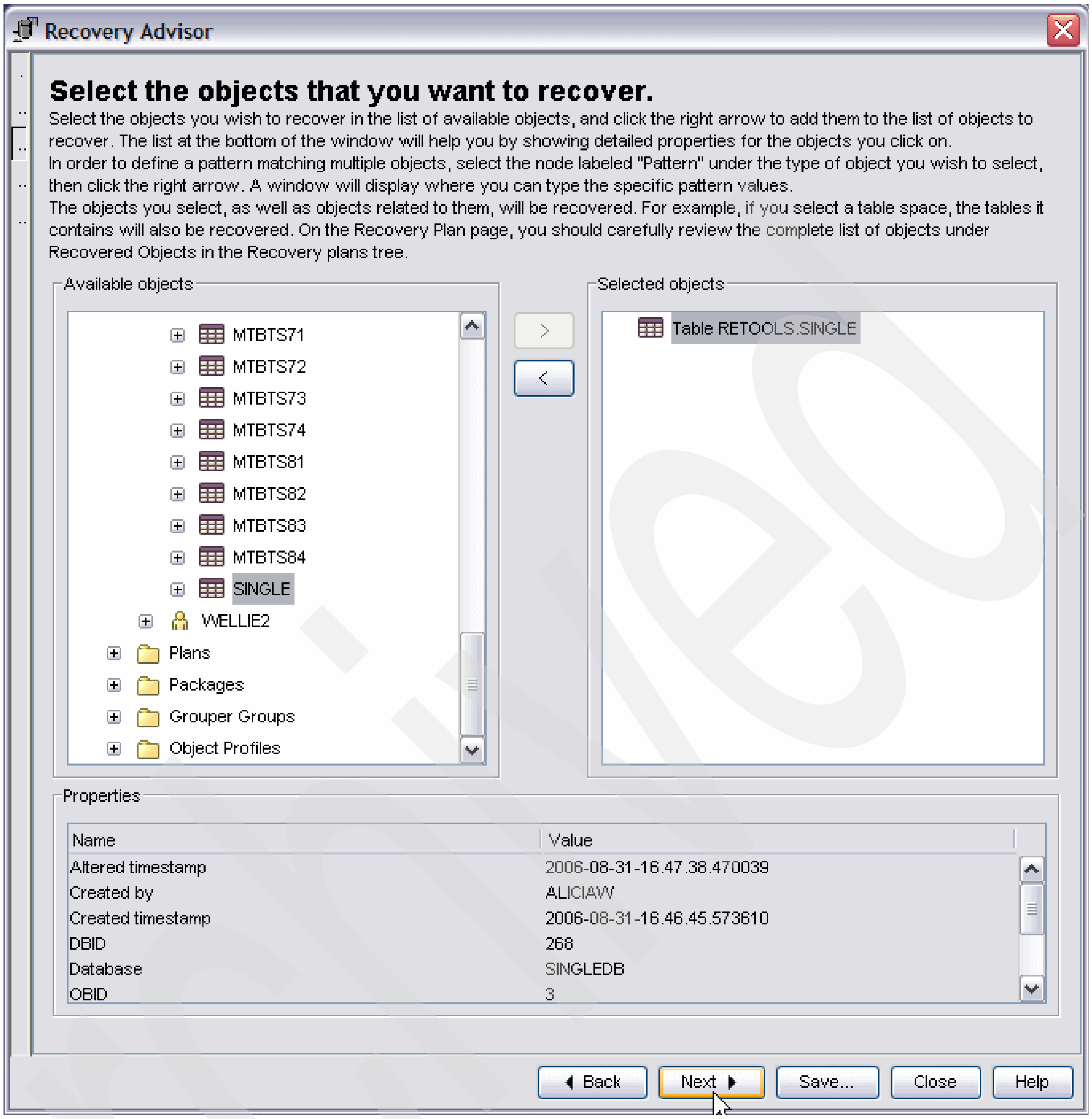

Figure 2-108 Selecting the dropped table RETOOLS.SINGLE to recover

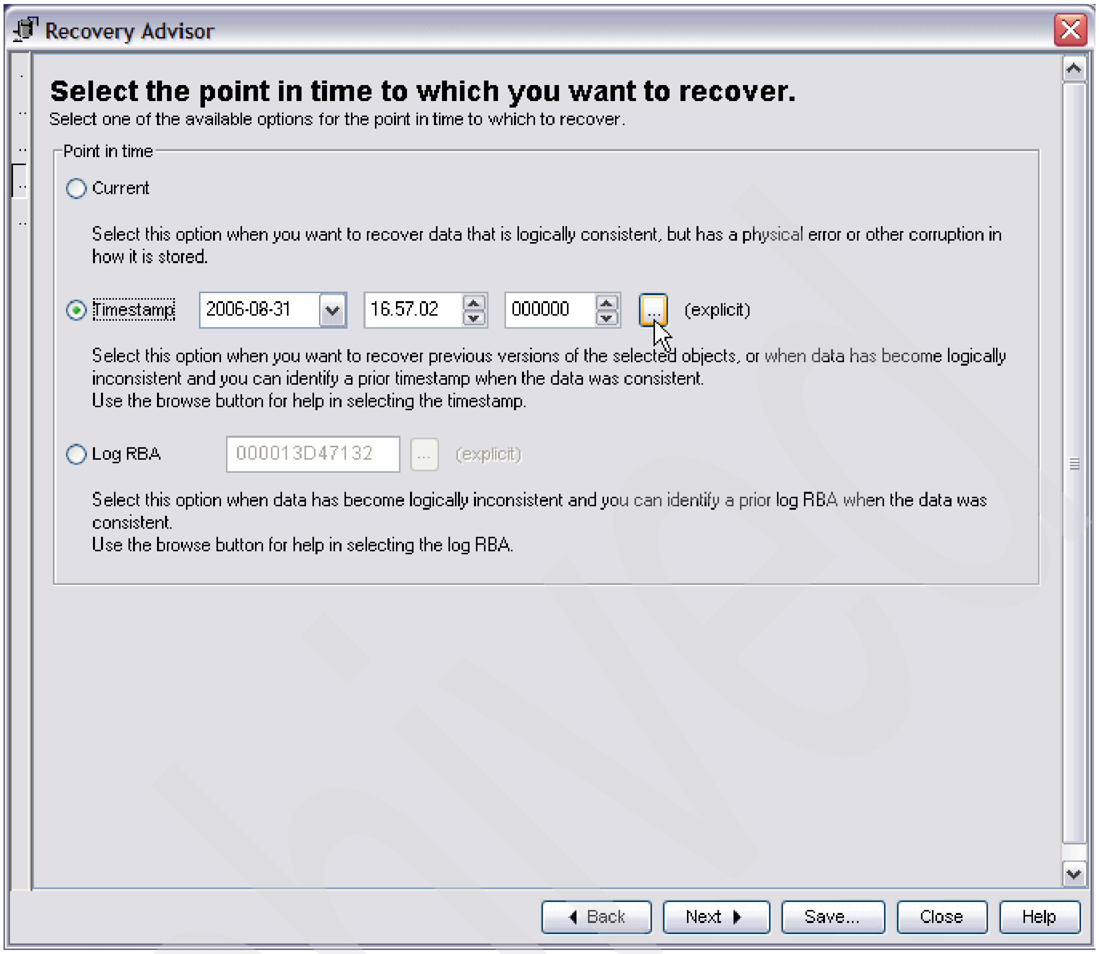

Figure 2-109 Selecting the point to recover to (1/3)

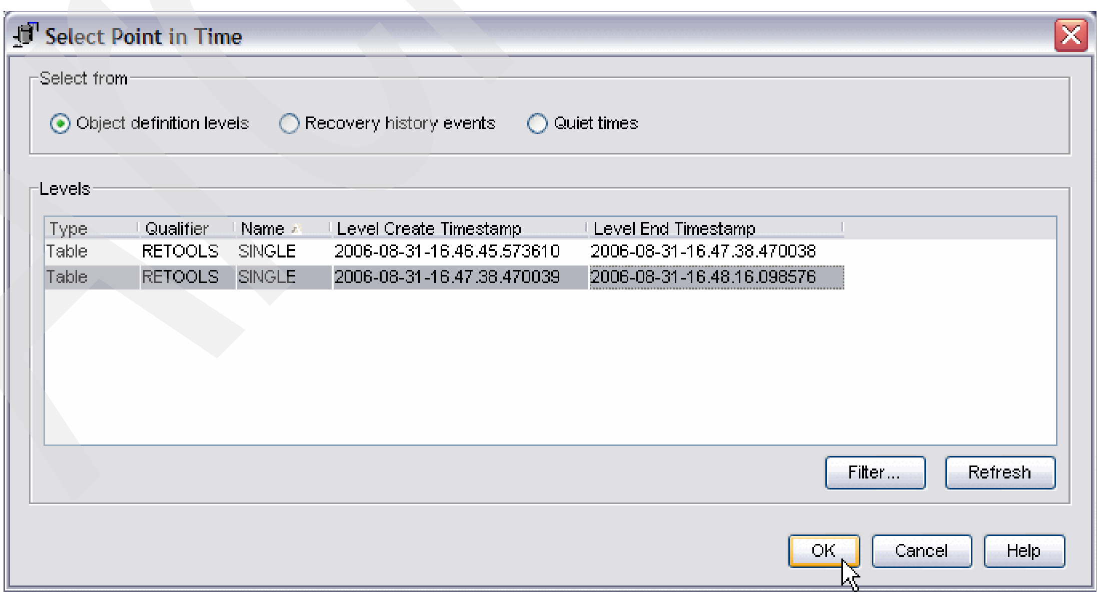

Figure 2-110 Selecting the point to recover to (2/3)

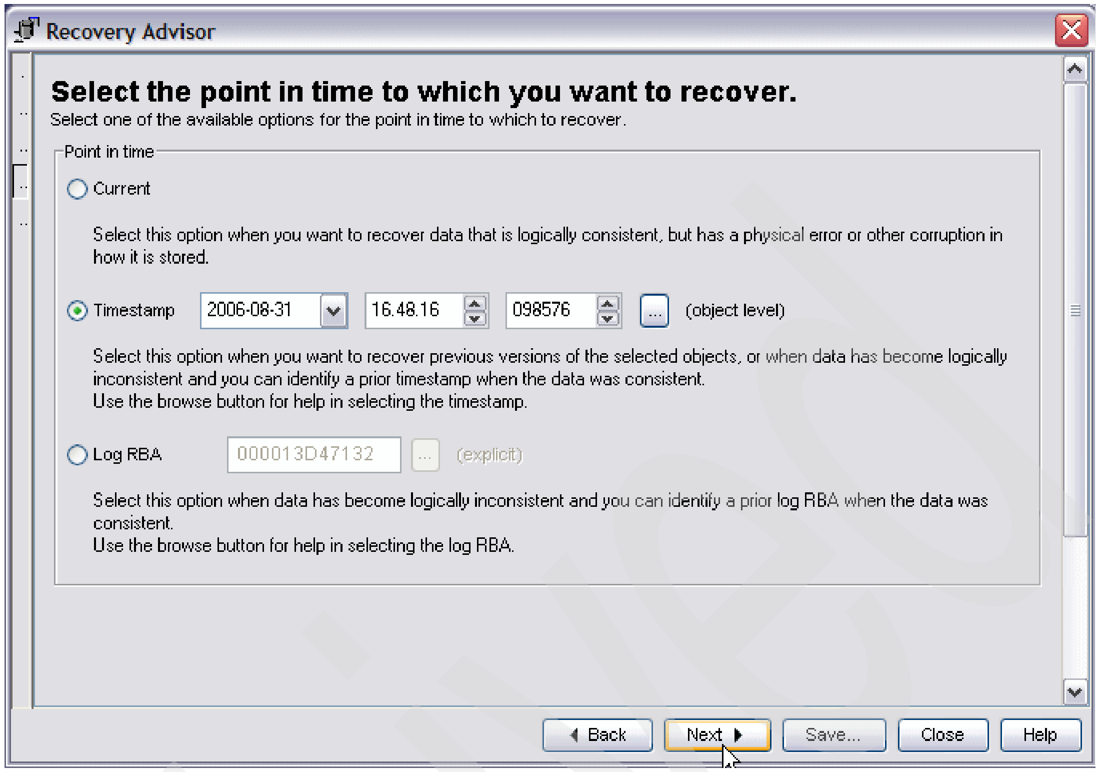

Figure 2-111 Selecting the point to recover to (3/3)

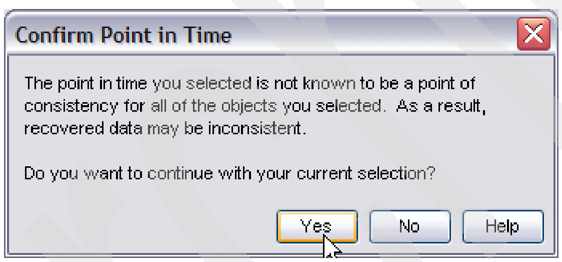

Figure 2-112 Warning message that the selected timestamp might not be a point-of-consistency

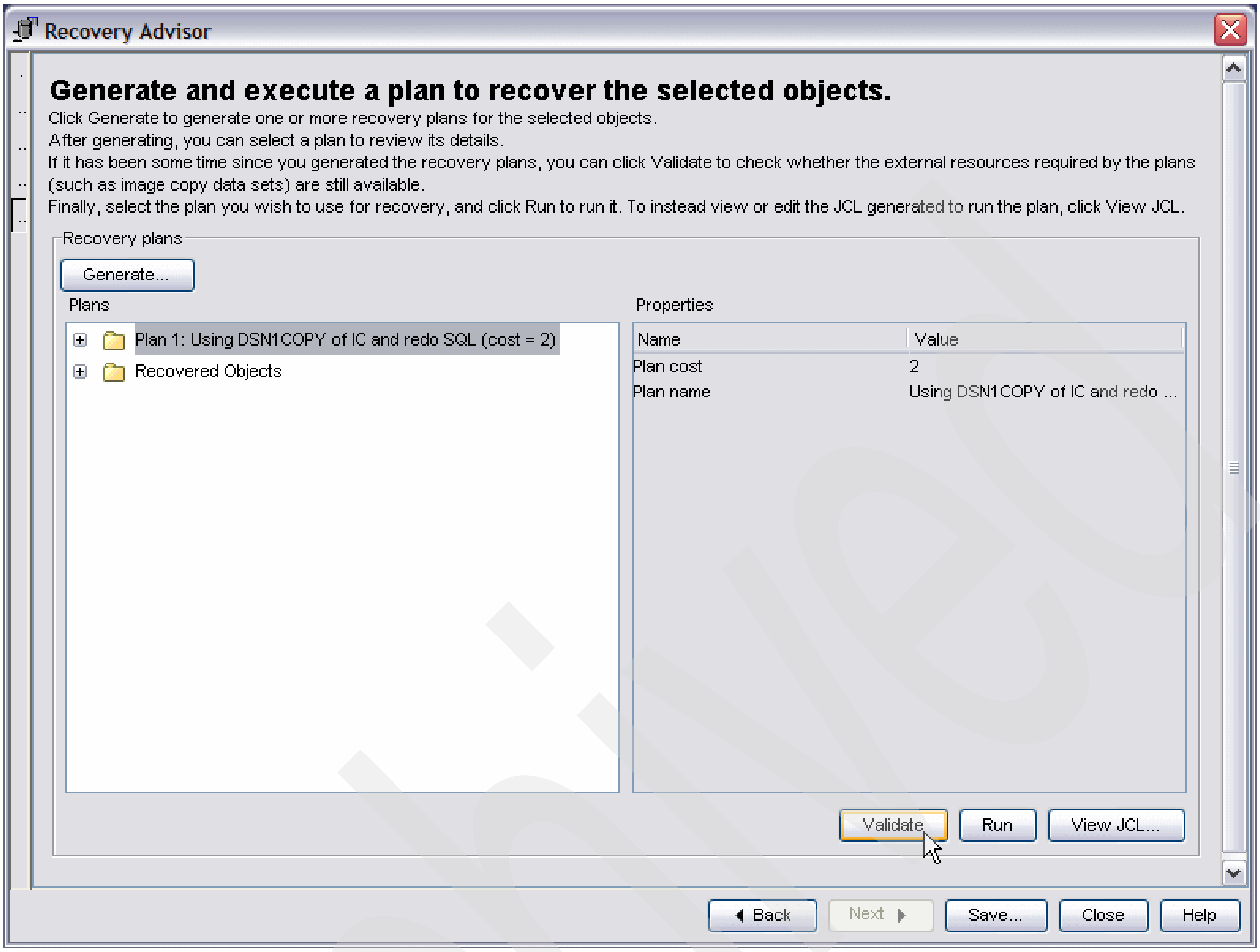

Figure 2-113 Generated recovery plans

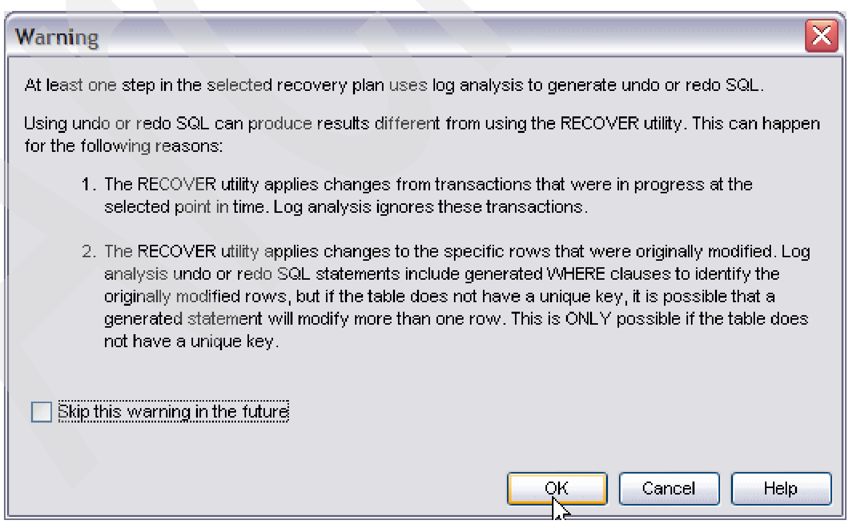

Figure 2-114 Warning message that a log analysis step is in the selected recovery plan

Example 2-19 Partial contents of the generated JCL for Plan 1

```
//W8SMPTB JOB (999,POK),'RETEST',CLASS=A,MSGCLASS=H,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
//*
....................
....................
//SQCR0002 EXEC PGM=ARY#UTIL,PARM='EXECSQL,DB8W ',COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
CONFIG DEFAULT
//SYSIN    DD   *
--#SET CONTINUE ON WARNING

  SET CURRENT SQLID = 'ALICIAW';
  CREATE TABLE "RETOOLS"."SINGLE"
    (DEPTNO               CHAR(3) NOT NULL FOR SBCS DATA
    ,DEPTNAME             VARCHAR(50) NOT NULL FOR SBCS DATA
    ,MGRNO                CHAR(6) FOR SBCS DATA WITH DEFAULT NULL
    ,ADMRDEPT             CHAR(3) NOT NULL FOR SBCS DATA
    ,LOCATION             CHAR(16) FOR SBCS DATA WITH DEFAULT NULL
    ,NEWCOL1              CHAR(1) FOR SBCS DATA WITH DEFAULT NULL
    ,PRIMARY KEY ( DEPTNO))
    IN SINGLEDB.SINGLETS
    AUDIT NONE
    CCSID EBCDIC
;
  COMMIT;

  SET CURRENT SQLID = 'RETOOLS';
  SET CURRENT PATH = "SYSIBM","SYSFUN","SYSPROC","RETOOLS";
--#SET TERMINATOR #
    CREATE VIEW RETOOLS.VSINGLE
    AS SELECT ALL DEPTNO , DEPTNAME, MGRNO , ADMRDEPT
    FROM RETOOLS.SINGLE#
--#SET TERMINATOR ;
  COMMIT;

  SET CURRENT SQLID = 'ALICIAW';
  CREATE UNIQUE INDEX "RETOOLS"."XSINGLE"
    ON "RETOOLS"."SINGLE"
    (DEPTNO ASC)
    USING STOGROUP "SMSSG"
      PRIQTY 12
      SECQTY 4
      ERASE NO
    PCTFREE 10
    PIECESIZE 2 G
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
;
```

```
  COMMIT;
/*
//*
......................
......................
//DCPX0005 EXEC PGM=ARY#UTIL,PARM='UTGENA,DB8W ',COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
CONFIG DEFAULT
//SYSXLAT  DD   DSN=NALUR.ARY06243.S170430.M123204.SYSXLAT,
//              UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//              DISP=(NEW,CATLG,DELETE)
//SYSIN    DD   *
TABLESPACE
0268,SINGLEDB
0002,SINGLETS
0003,"RETOOLS"."SINGLE"
/*
//*
//DCPY0006 EXEC PGM=DSN1COPY,COND=(4,LT),
// PARM='CHECK,PAGESIZE(4K),FULLCOPY,OBIDXLAT,RESET'
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSUT1   DD DISP=SHR,DSN=REIC.DB8W.SINGLEDB.SINGLETS.FULL.G0002V00
//SYSUT2   DD DISP=SHR,DSN=DB8WUS.DSNDBC.SINGLEDB.SINGLETS.I0001.A001
//SYSPRINT DD   SYSOUT=*
//SYSXLAT  DD DISP=SHR,DSN=NALUR.ARY06243.S170430.M123204.SYSXLAT
//*
//RBLD0007 EXEC PGM=DSNUTILB,PARM=(DB8W),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//UTPRINT  DD   SYSOUT=*
//SYSOUT   DD   UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN    DD   *

  REBUILD INDEX
    ( "RETOOLS"."XSINGLE" )
      SCOPE ALL

/*
..........................
..........................
//*********************************************************************
//* STEP 2: READ THE DB2 LOG TO GENERATE THE GENERAL REPORT           *
//*********************************************************************
//LASR0002 EXEC PGM=ARYGEN1,REGION=0M,COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
```

```
//MODEFILE DD   DSN=NALUR.ARYLAS.MODE.R0000335,
// DISP=OLD
//SYSOUT   DD   SYSOUT=*
//CFILES   DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//GENRPT   DD   SYSOUT=*
//EXTREP   DD   SYSOUT=*
//SUMREPT  DD   SYSOUT=*
//XDREPT   DD   SYSOUT=*
//QTRPT    DD   SYSOUT=*
//WARNINGS DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//DATAIN   DD   *
SSID           =DB8W
START RBA      =00001481695A
END RBA        =000014855441
/*
//FILTERS  DD *
GENERAL FILTERS...........
SHOW UPDATES   =Y
SHOW INSERTS   =X
SHOW DELETES   =Y
SHOW ROLLBACKS =N
CATALOG DATA   =N
SHOW UNCOMMITS =N
OBJECT FILTERS-BY IDS.....
00000,00000,00000,00268,00002,00003,R,00001481695A,000014855441
OBJECT FILTERS-BY NAME....
TABLE OWNER=RETOOLS
TABLE NAME =SINGLE
PARTITION  =0000
SRBA       =00001481695A
ERBA       =000014855441
INC/EXC    =I
/*
//**********************************
//* PARAMETERS FOR JOB SHOWN BELOW *
//**********************************
//* SSID           =DB8W
//* START RBA      =00001481695A
//* END RBA        =000014855441
//* /*
//* //FILTERS  DD *
//* GENERAL FILTERS...........
//* SHOW UPDATES   =Y
//* SHOW INSERTS   =X
//* SHOW DELETES   =Y
//* SHOW ROLLBACKS =N
//* CATALOG DATA   =N
//* SHOW UNCOMMITS =N
//* OBJECT FILTERS-BY IDS.....
//* 00000,00000,00000,00268,00002,00003,R,00001481695A,000014855441
//* OBJECT FILTERS-BY NAME....
//* TABLE OWNER=RETOOLS
```

```
//* TABLE NAME =SINGLE
//* PARTITION  =0000
//* SRBA       =00001481695A
//* ERBA       =000014855441
//* INC/EXC    =I
//*
.....................
.....................
//LASR0006 EXEC PGM=ARYDTL4,REGION=0M,COND=(4,LT)
//*******************************************************************
//*  STEP 6: THIS STEP DOES THE FOLLOWING:
//*         -SQL GENERATION
//*******************************************************************
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//MODEFILE DD   DSN=NALUR.ARYLAS.MODE.R0000335,
// DISP=OLD
//SQLOUT   DD   DSN=NALUR.ARY06243.S170430.M162894.EXECSQL,
//         DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE),
//         DCB=(LRECL=80,BLKSIZE=23440,RECFM=FB)
//SYSOUT   DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*                                                01870000
//SYSUDUMP DD   SYSOUT=*                                                01880000
//WARNINGS DD   SYSOUT=*                                                01890000
//CFILES   DD   SYSOUT=*
//*                                                                     01920000
//*
....................
....................
//LGNL0008 EXEC PGM=ARY#UTIL,PARM='EXECSQL,DB8W ',
//             COND=((4,LT),(0,NE,LASR0002))
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
CONFIG DEFAULT
CONTINUE ON WARNING
//SYSIN    DD   DSN=NALUR.ARY06243.S170430.M162894.EXECSQL,
//             DISP=(OLD,KEEP)
//*
.................
.................
```

2.5.2 Recovering table in single-table table space to IC at current DDL version

This scenario describes the recovery of a single table in a single-table table space to an image copy that was taken after the last DDL change that occurred on the table, that is, at the current DDL version of the object. The timeline shown in Figure 2-115 on page 178 is assumed for this scenario. This environment is a non-data sharing environment.

The following windows only represent the ones of particular interest (in our opinion) to this scenario. The windows leading up to these windows and those that follow these windows are almost identical to the windows described in 2.3.1, “Recovering multi-table table space” on page 56. We show the partial contents of the generated JCL for a given recovery plan, but not the execution job results.

Figure 2-116 on page 179 shows the dropped table RETOOLS.DATST1 selected for recovery in the navigation pane. Its red color in the navigation pane confirms that this table has been dropped. Figure 2-116 on page 179 through Figure 2-119 on page 182 show the selection of a full image copy (share level reference) corresponding to time t10 in Figure 2-115 on page 178 from a list of available entries in SYSIBM.SYSCOPY (recovery history events); not all window interactions are shown here. Figure 2-119 on page 182 shows this image copy selection translated to a timestamp corresponding to the timestamp in the image copy entry that is selected in Figure 2-118 on page 181. Figure 2-120 on page 182 shows the COPY parameters specified for the recovery plan generation. Figure 2-121 on page 183 shows a single recovery plan generated for the requested recovery action and the expanded steps included in the recovery plan, Plan 1. A portion of the generated JCL for Plan 1 is shown in Example 2-20 on page 183.

The selected recovery plan, Plan 1, has multiple steps that include the creation of the dropped table and its dependencies (such as indexes and views, if any), followed by a DSN1COPY restore of the latest image copy with an OBID translate to the new OBID of the object. This is followed by a rebuilding of the indexes to complete the recovery action requested. An image copy is then taken of this table space, as specified in Figure 2-120 on page 182.

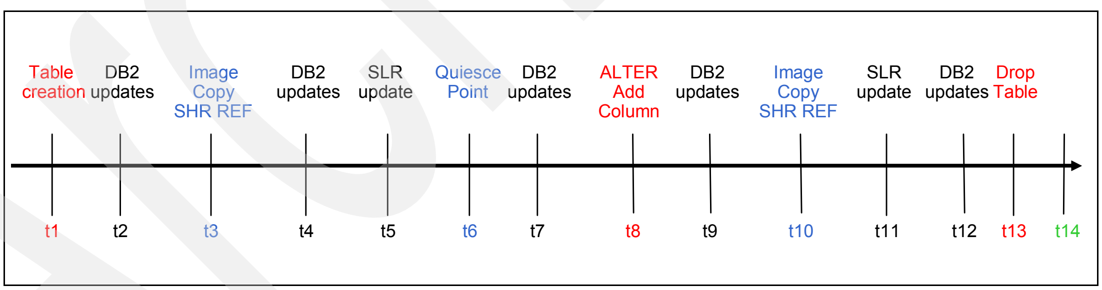

Figure 2-115 Recovering dropped table in single-table table space to IC at current DDL scenario timeline

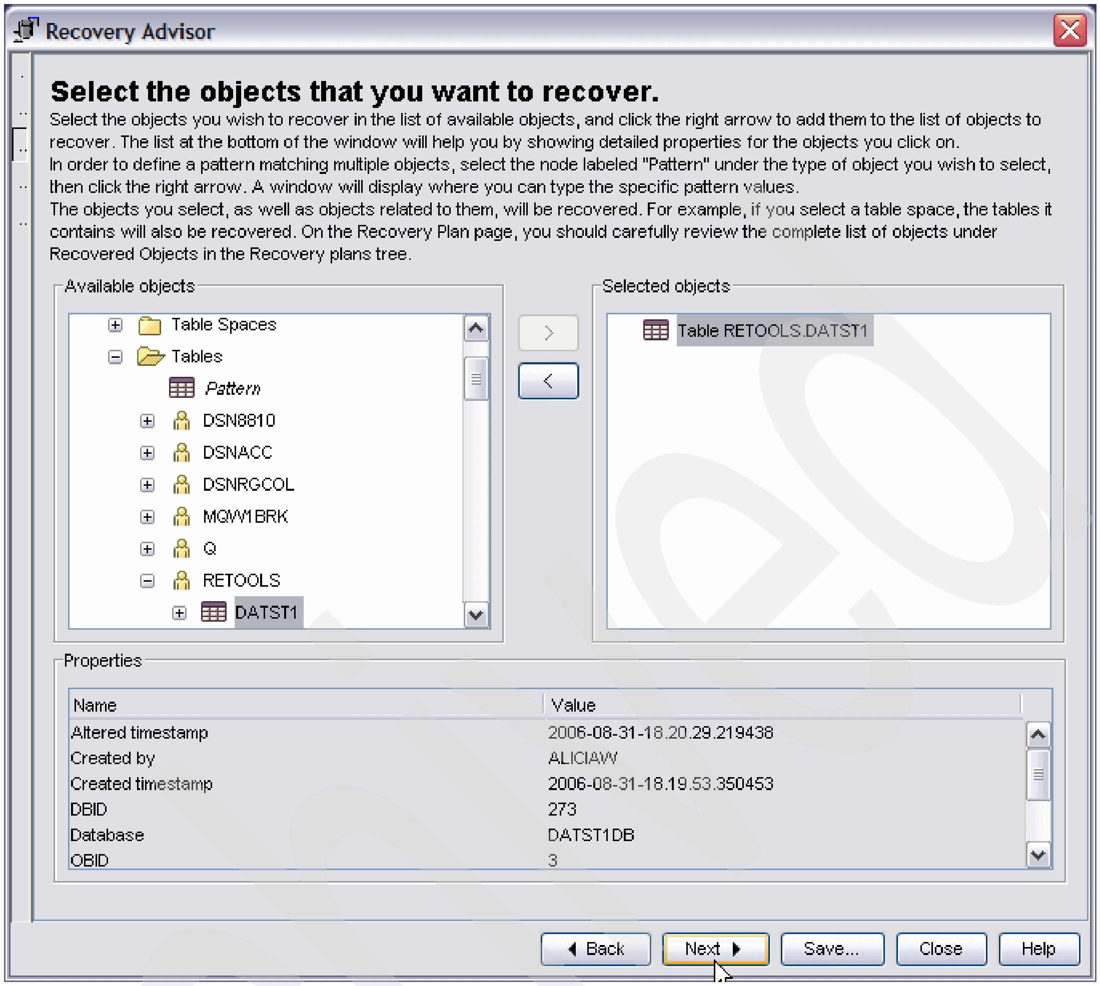

Figure 2-116 Selecting the dropped table to recover

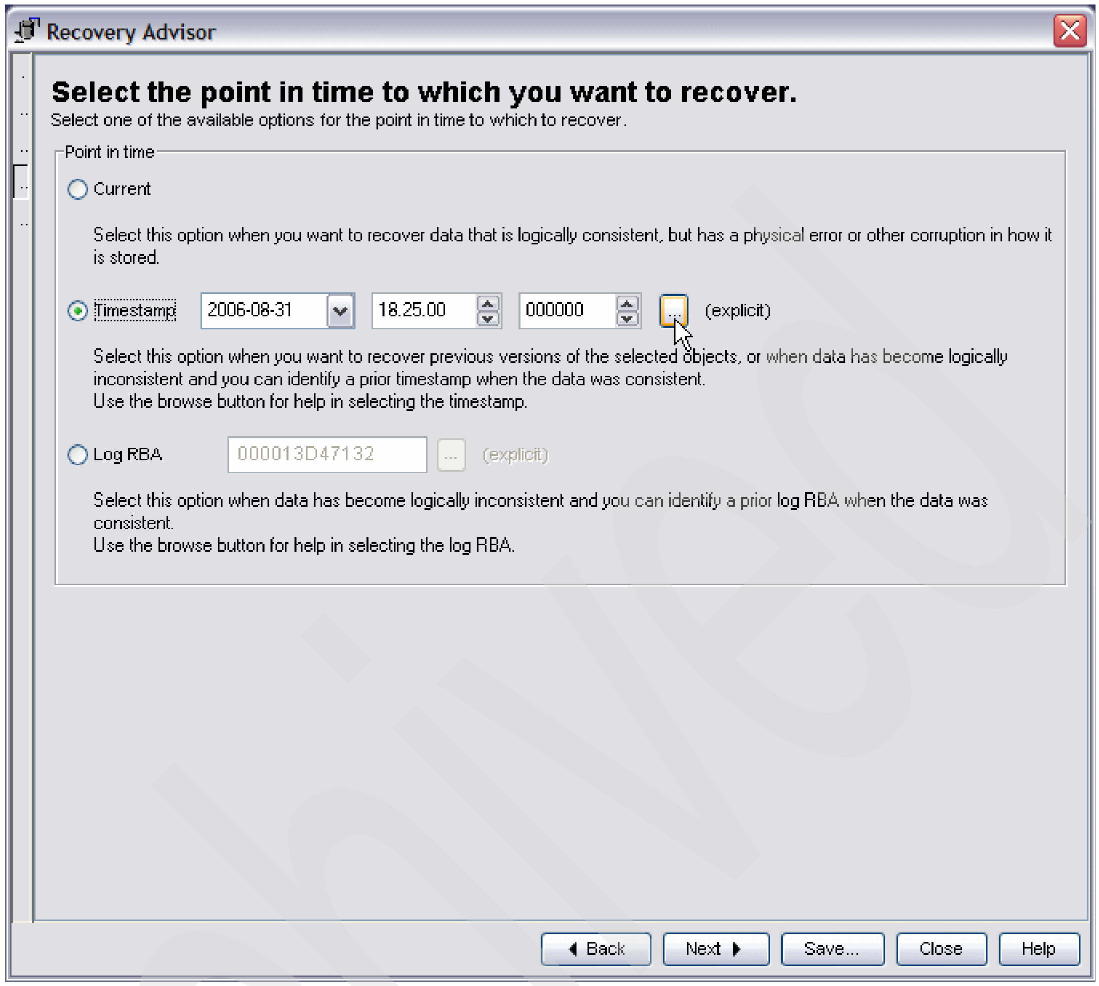

Figure 2-117 Selecting the point to recover to (1/3)

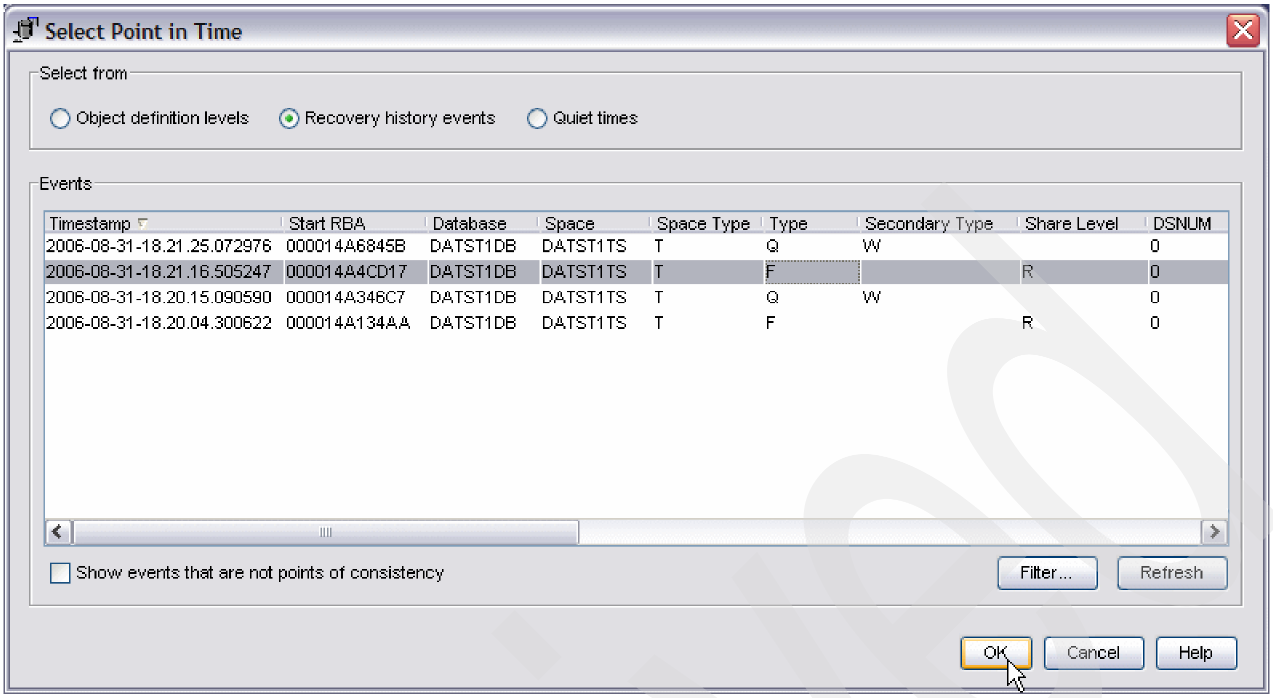

Figure 2-118 Selecting the point to recover to (2/3)

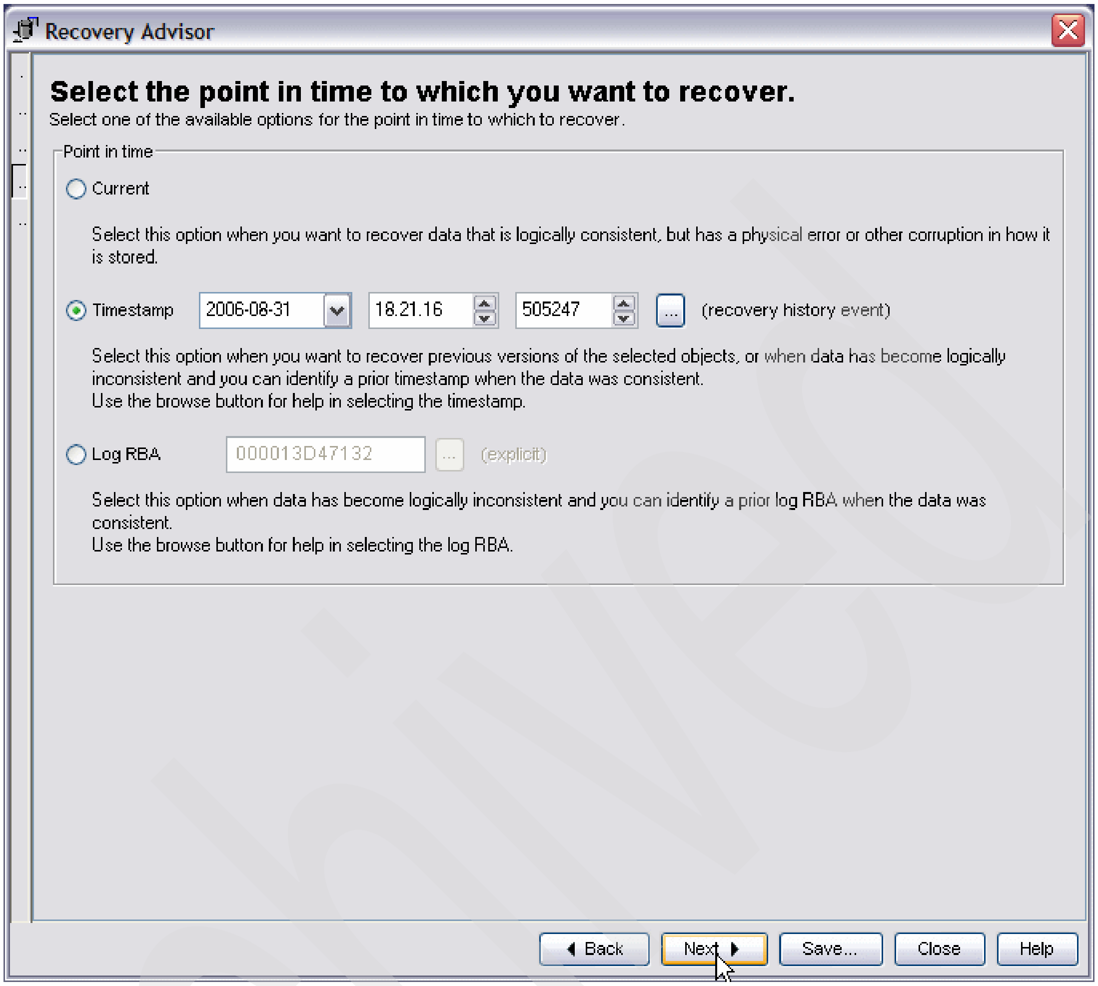

Figure 2-119   Selecting the point to recover to (3/3)

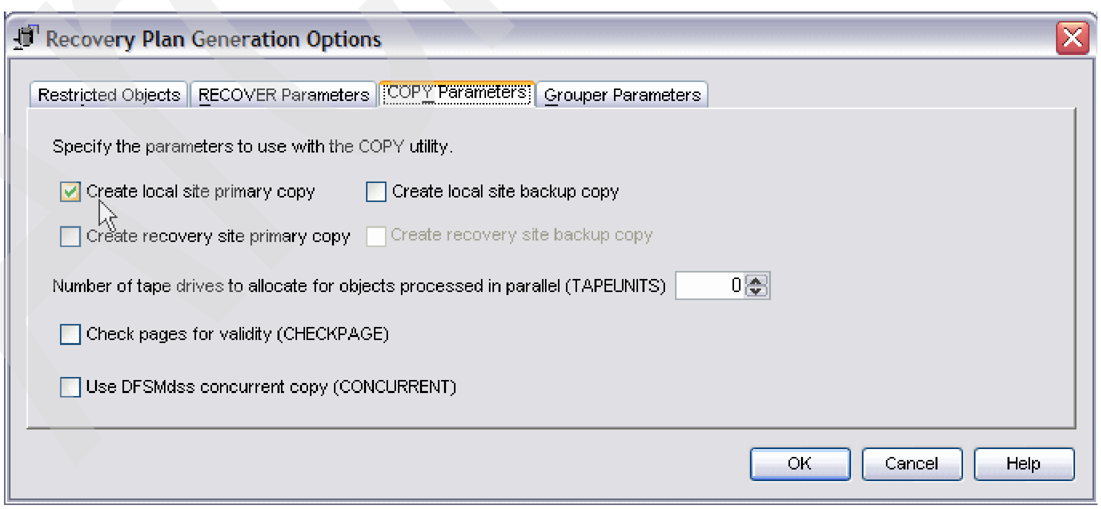

Figure 2-120   Copy parameters for recovery plan generation

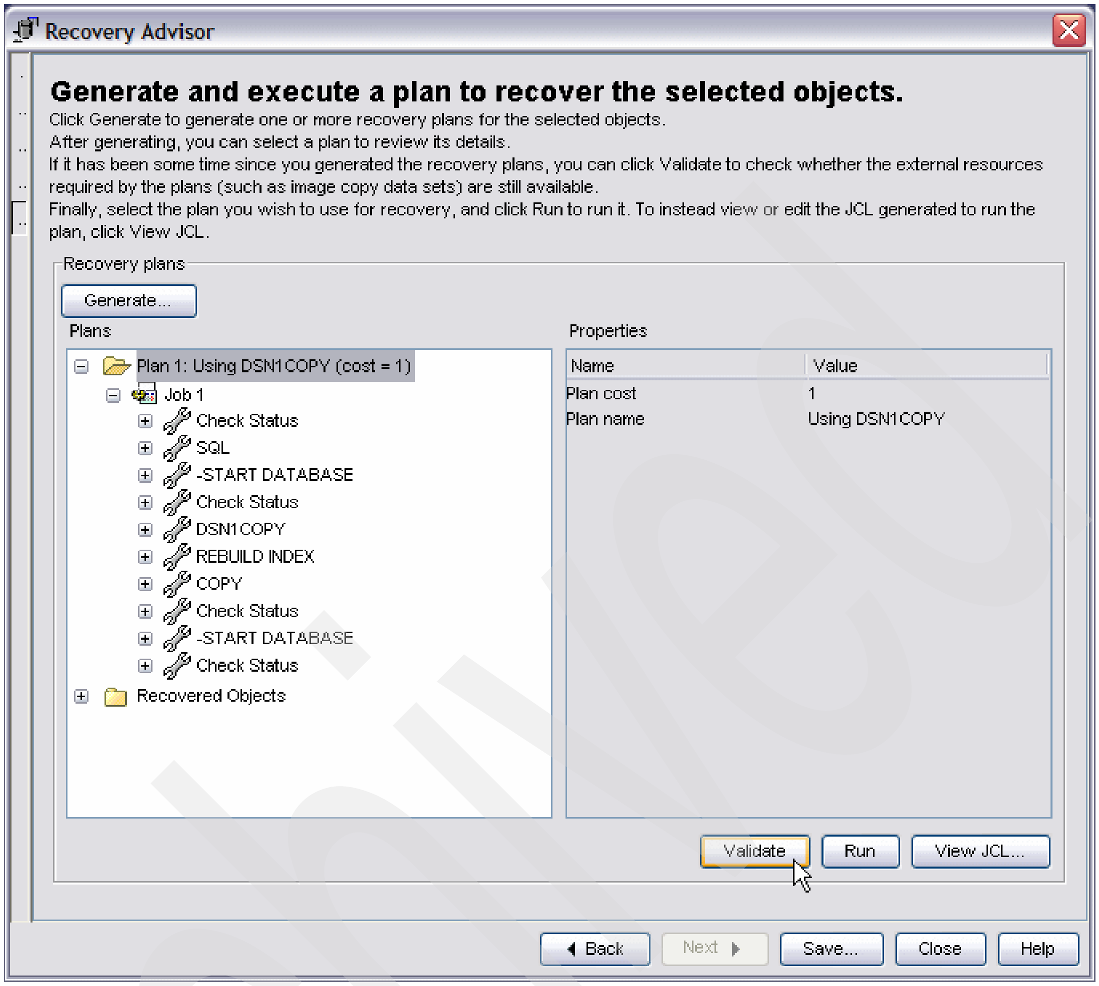

Figure 2-121 Generated recovery plans

Example 2-20 Partial contents of the generated JCL for Plan 1

```
//W8TBL JOB (999,POK),'RETEST',CLASS=A,MSGCLASS=H,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
//*
..................
..................
//SQCR0002 EXEC PGM=ARY#UTIL,PARM='EXECSQL,DB8W ',COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
CONFIG DEFAULT
//SYSIN    DD   *
--#SET CONTINUE ON WARNING

  SET CURRENT SQLID = 'ALICIAW';
  CREATE TABLE "RETOOLS"."DATST1"
    (DEPTNO              CHAR(3) NOT NULL FOR SBCS DATA
    ,DEPTNAME            VARCHAR(50) NOT NULL FOR SBCS DATA
```

```
    ,MGRNO                CHAR(6) FOR SBCS DATA WITH DEFAULT NULL
    ,ADMRDEPT             CHAR(3) NOT NULL FOR SBCS DATA
    ,LOCATION             CHAR(16) FOR SBCS DATA WITH DEFAULT NULL
    ,NEWCOL1              CHAR(1) FOR SBCS DATA WITH DEFAULT NULL
    ,PRIMARY KEY ( DEPTNO))
    IN DATST1DB.DATST1TS
    AUDIT NONE
    CCSID EBCDIC
;
  COMMIT;

  SET CURRENT SQLID = 'RETOOLS';
  SET CURRENT PATH = "SYSIBM","SYSFUN","SYSPROC","RETOOLS";
--#SET TERMINATOR #
    CREATE VIEW RETOOLS.VDATST1
    AS SELECT ALL DEPTNO , DEPTNAME, MGRNO , ADMRDEPT
    FROM RETOOLS.DATST1#
--#SET TERMINATOR ;
  COMMIT;

  SET CURRENT SQLID = 'ALICIAW';
  CREATE UNIQUE INDEX "RETOOLS"."XDATST1"
    ON "RETOOLS"."DATST1"
    (DEPTNO ASC)
    USING STOGROUP "SMSSG"
      PRIQTY 12
      SECQTY 4
      ERASE NO
    PCTFREE 10
    PIECESIZE 2 G
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
;
  COMMIT;
/*
.................
.................
//DCPX0005 EXEC PGM=ARY#UTIL,PARM='UTGENA,DB8W ',COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
CONFIG DEFAULT
//SYSXLAT  DD   DSN=NALUR.ARY06243.S182811.M912459.SYSXLAT,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(NEW,CATLG,DELETE)
//SYSIN    DD   *
TABLESPACE
0273,DATST1DB
0002,DATST1TS
0003,"RETOOLS"."DATST1"
/*
```

```
//*
//DCPY0006 EXEC PGM=DSN1COPY,COND=(4,LT),
// PARM='CHECK,PAGESIZE(4K),FULLCOPY,OBIDXLAT,RESET'
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSUT1   DD DISP=SHR,DSN=REIC.DB8W.DATST1DB.DATST1TS.FULL.G0002V00
//SYSUT2   DD DISP=SHR,DSN=DB8WUS.DSNDBC.DATST1DB.DATST1TS.I0001.A001
//SYSPRINT DD   SYSOUT=*
//SYSXLAT  DD DISP=SHR,DSN=NALUR.ARY06243.S182811.M912459.SYSXLAT
//*
//RBLD0007 EXEC PGM=DSNUTILB,PARM=(DB8W),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//UTPRINT  DD   SYSOUT=*
//SYSOUT   DD   UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN    DD   *

  REBUILD INDEX
    ( "RETOOLS"."XDATST1" )
      SCOPE ALL

/*
//*
//COPY0008 EXEC PGM=DSNUTILB,PARM=(DB8W),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//CPY00010 DD   DSN=NALUR.ARY06243.S182811.M941928.COPY,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(NEW,CATLG,DELETE)
//SYSIN    DD   *

  COPY
    TABLESPACE DATST1DB.DATST1TS
      FULL YES
      COPYDDN(CPY00010)

/*
//*
.................
.................
```

2.5.3 Recovering table in single-table table space to IC before current DDL version

This scenario describes the recovery of a single table in a single-table table space to an image copy that was taken before the last DDL change that occurred on the table, that is, before the current DDL version of the object. The timeline shown in Figure 2-107 on page 169 is assumed for this scenario. This environment is a non-data sharing environment.

The following windows only represent the ones of particular interest (in our opinion) to this scenario. The windows leading up to these windows and those that follow these windows are almost identical to the windows described in 2.3.1, “Recovering multi-table table space” on page 56. We show the partial contents of the generated JCL for a given recovery plan, but not the execution job results.

Figure 2-122 on page 187 shows the dropped table RETOOLS.DATST2 selected for recovery in the navigation pane. Its red color in the navigation pane confirms that this table has been dropped. DBTST1 is also red in color, which indicates that it is a dropped table, but it is not selected for recovery. Figure 2-123 on page 188 through Figure 2-125 on page 189 show the selection of a full image copy (share level change) corresponding to time t3 in Figure 2-107 on page 169 from a list of available entries in SYSIBM.SYSCOPY (recovery history events); not all window interactions are shown here. Select the **Show events that are not points of consistency** check box to see image copies taken with share level change, as shown in Figure 2-124 on page 188. Figure 2-125 on page 189 shows this image copy selection translated to a timestamp corresponding to the timestamp in the image copy entry that is selected in Figure 2-124 on page 188. Figure 2-126 on page 189 shows a warning message that the selected timestamp might not be a point-of-consistency. Figure 2-127 on page 190 shows a single recovery plan generated for the requested recovery action, expanded recovered objects, and the expanded steps included in the recovery plan, Plan 1. A portion of the generated JCL for Plan 1 is shown in Example 2-21 on page 190.

The selected recovery plan, Plan 1, has multiple steps that include the creation of the dropped table and its dependencies (such as indexes and views, if any), followed by a DSN1COPY restore of the latest image copy with an OBID translate to the new OBID of the object. This is followed by a rebuilding of the indexes to complete the recovery action requested.

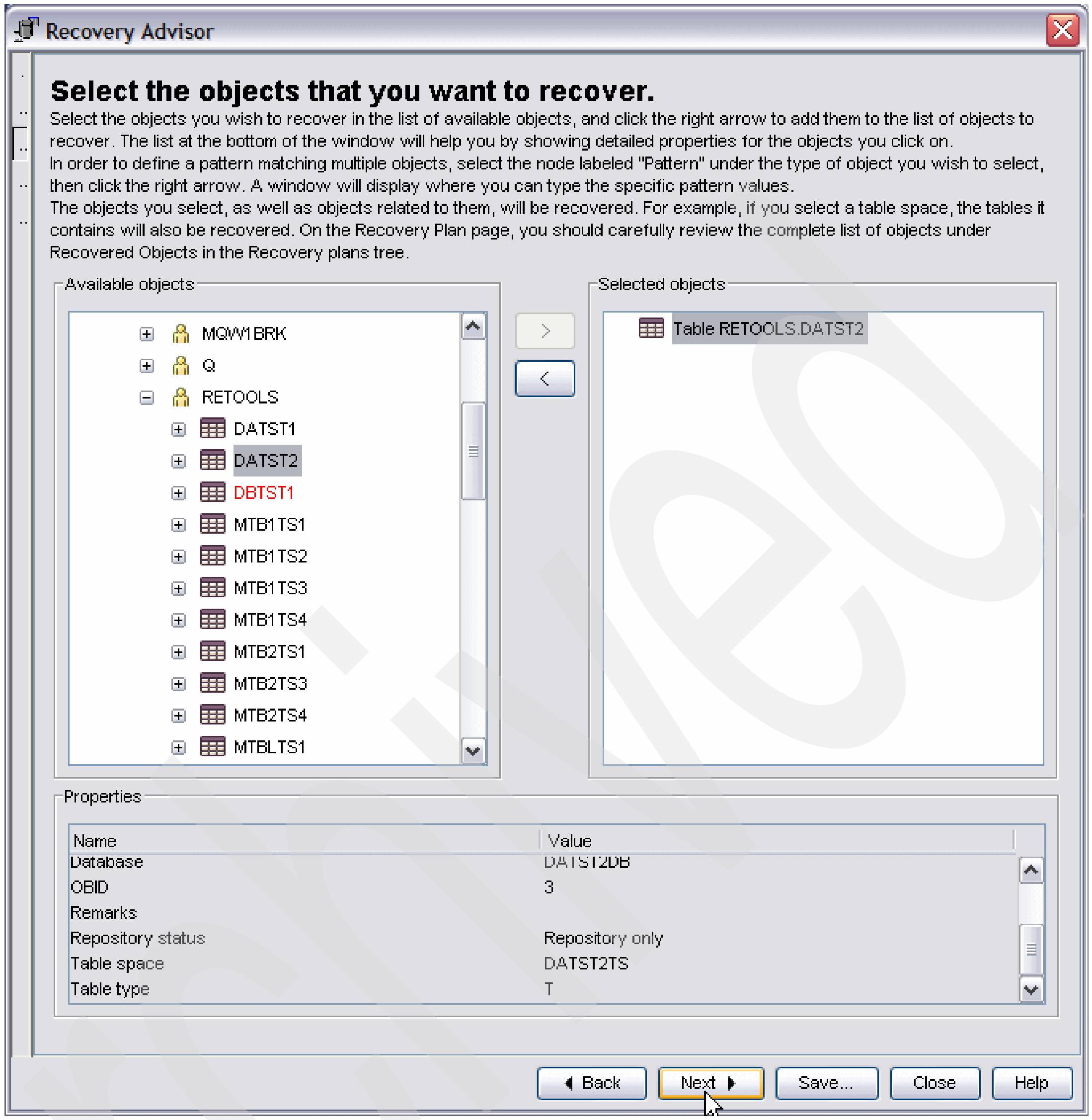

Figure 2-122 Selecting the dropped table to recover

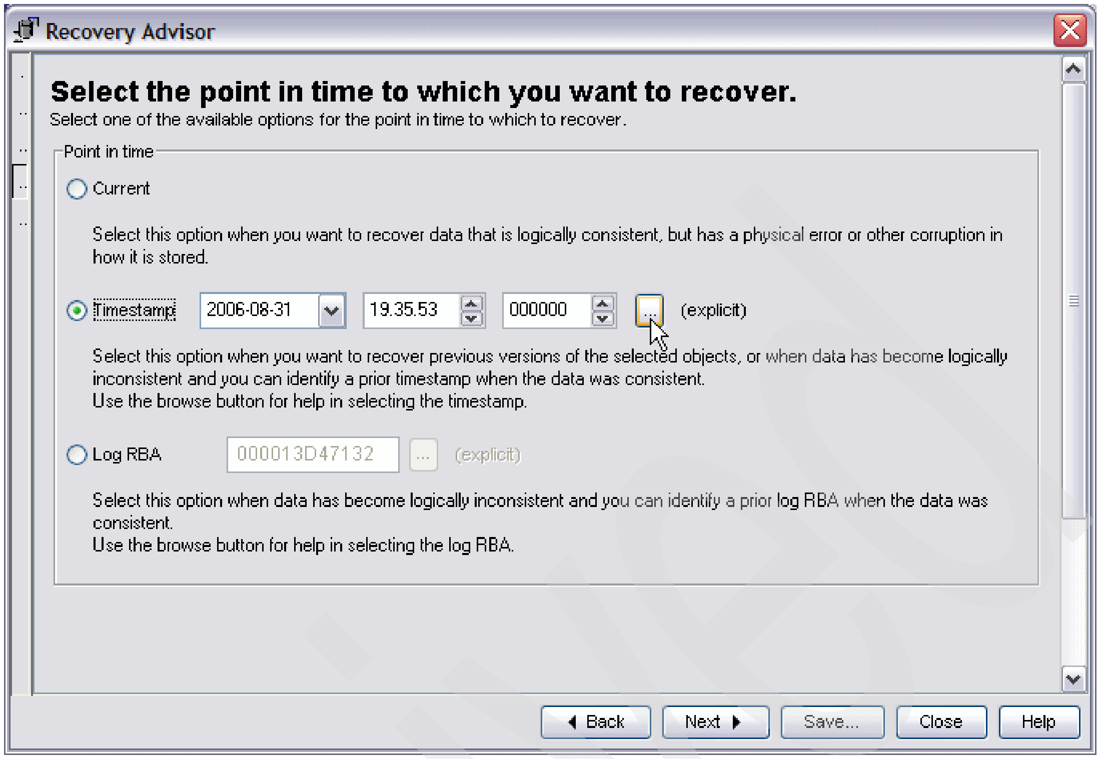

Figure 2-123 Selecting the point to recover to (1/3)

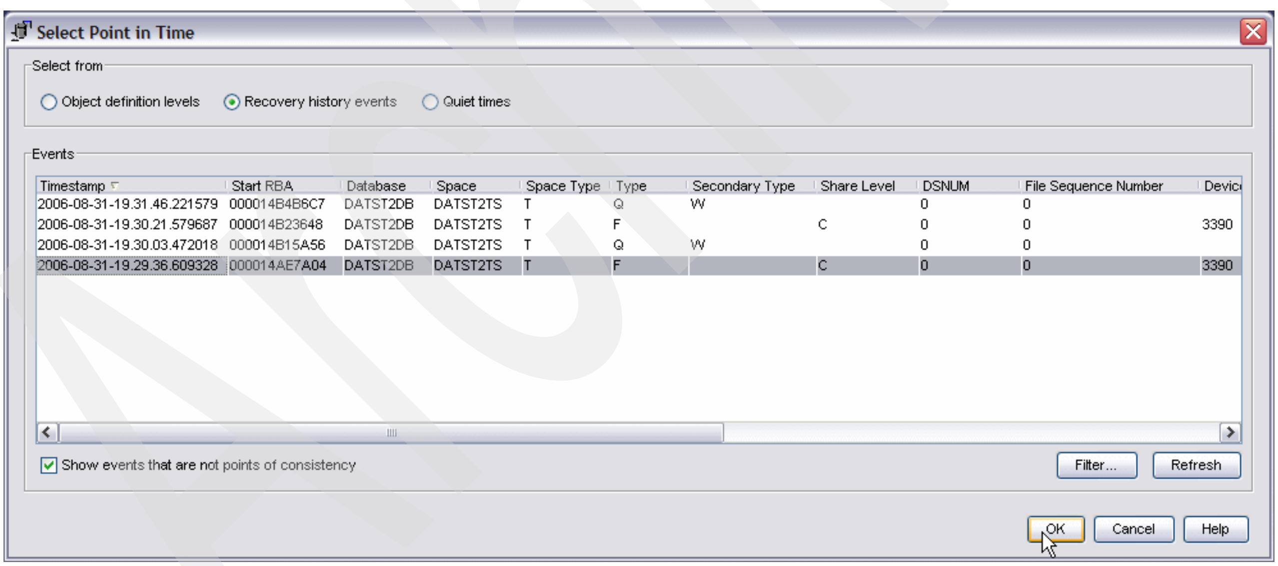

Figure 2-124 Selecting the point to recover to (2/3)

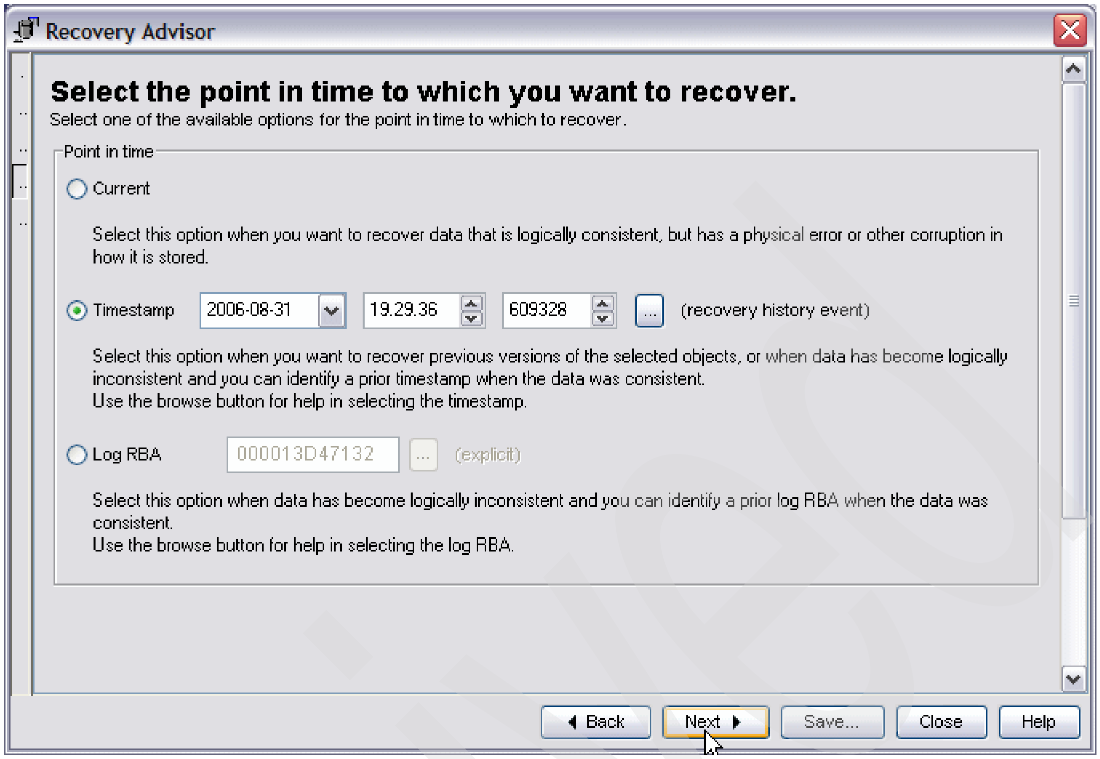

Figure 2-125 Selecting the point to recover to (3/3)

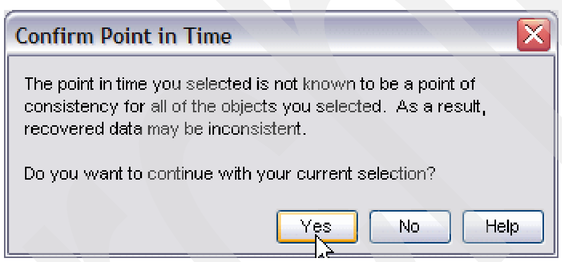

Figure 2-126 Warning message that selected timestamp might not be a point-of-consistency

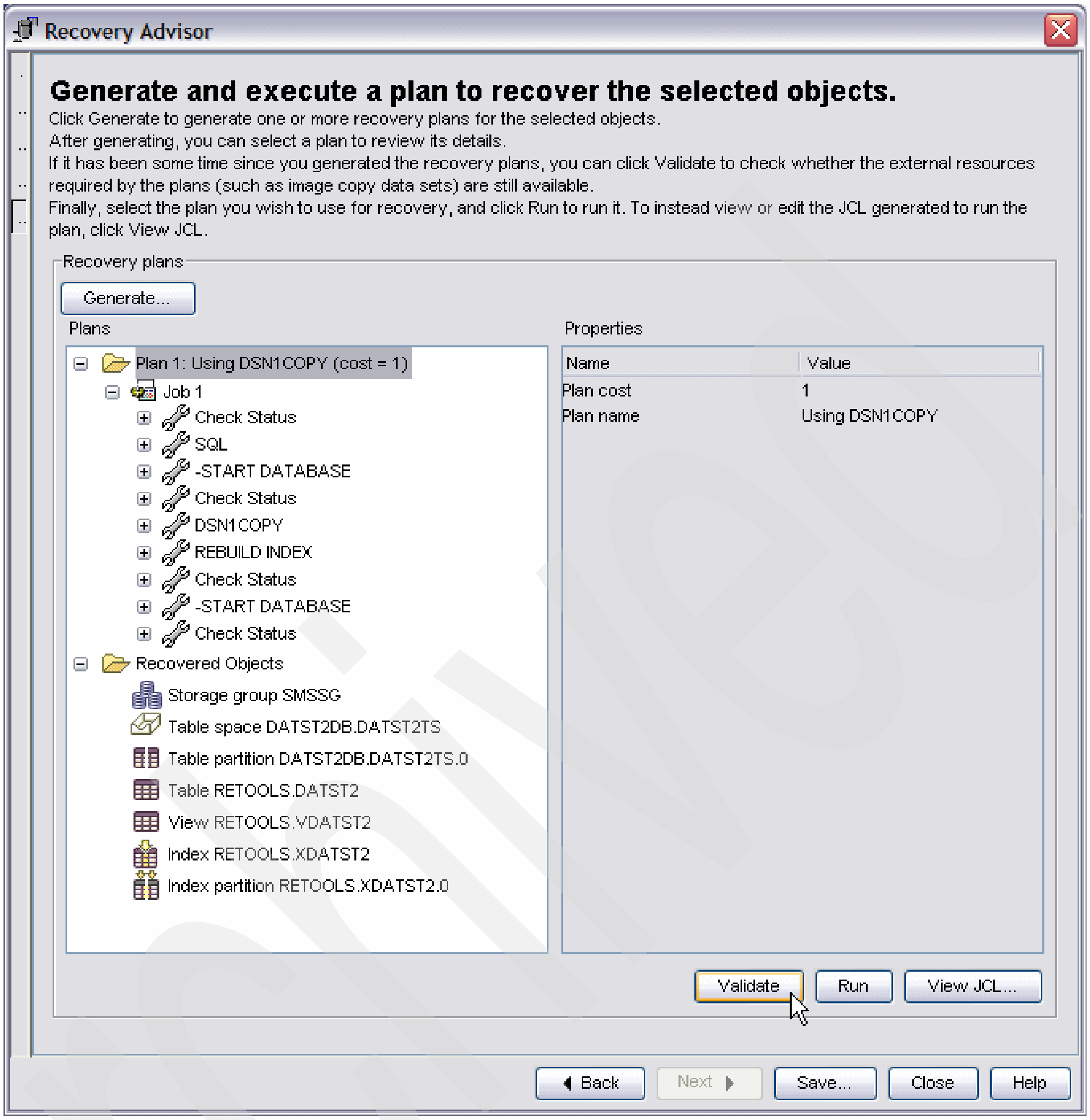

Figure 2-127 Generated recovery plans

Example 2-21 Partial contents of the generated JCL for Plan 1

```
//W8TBLB4 JOB (999,POK),'RETEST',CLASS=A,MSGCLASS=H,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
//*
.....................
.....................
//SQCR0002 EXEC PGM=ARY#UTIL,PARM='EXECSQL,DB8W ',COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
CONFIG DEFAULT
//SYSIN    DD   *
--#SET CONTINUE ON WARNING
```

```
  SET CURRENT SQLID = 'ALICIAW';
  CREATE TABLE "RETOOLS"."DATST2"
    (DEPTNO               CHAR(3) NOT NULL FOR SBCS DATA
    ,DEPTNAME             VARCHAR(36) NOT NULL FOR SBCS DATA
    ,MGRNO                CHAR(6) FOR SBCS DATA WITH DEFAULT NULL
    ,ADMRDEPT             CHAR(3) NOT NULL FOR SBCS DATA
    ,LOCATION             CHAR(16) FOR SBCS DATA WITH DEFAULT NULL
    ,PRIMARY KEY ( DEPTNO))
    IN DATST2DB.DATST2TS
    AUDIT NONE
    CCSID EBCDIC
;
  COMMIT;

  SET CURRENT SQLID = 'RETOOLS';
  SET CURRENT PATH = "SYSIBM","SYSFUN","SYSPROC","RETOOLS";
--#SET TERMINATOR #
    CREATE VIEW RETOOLS.VDATST2
    AS SELECT ALL DEPTNO , DEPTNAME, MGRNO , ADMRDEPT
    FROM RETOOLS.DATST2#
--#SET TERMINATOR ;
  COMMIT;

  SET CURRENT SQLID = 'ALICIAW';
  CREATE UNIQUE INDEX "RETOOLS"."XDATST2"
    ON "RETOOLS"."DATST2"
    (DEPTNO ASC)
    USING STOGROUP "SMSSG"
      PRIQTY 12
      SECQTY 4
      ERASE NO
    PCTFREE 10
    PIECESIZE 2 G
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
;
  COMMIT;
/*
//*
...............
...............
//DCPX0005 EXEC PGM=ARY#UTIL,PARM='UTGENA,DB8W ',COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
CONFIG DEFAULT
//SYSXLAT  DD   DSN=NALUR.ARY06243.S194125.M665542.SYSXLAT,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(NEW,CATLG,DELETE)
//SYSIN    DD   *
TABLESPACE
```

```
0274,DATST2DB
0002,DATST2TS
0003,"RETOOLS"."DATST2"
/*
//*
//DCPY0006 EXEC PGM=DSN1COPY,COND=(4,LT),
// PARM='CHECK,PAGESIZE(4K),FULLCOPY,OBIDXLAT,RESET'
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSUT1   DD DISP=SHR,DSN=REIC.DB8W.DATST2DB.DATST2TS.FULL.G0001V00
//SYSUT2   DD DISP=SHR,DSN=DB8WUS.DSNDBC.DATST2DB.DATST2TS.I0001.A001
//SYSPRINT DD   SYSOUT=*
//SYSXLAT  DD DISP=SHR,DSN=NALUR.ARY06243.S194125.M665542.SYSXLAT
//*
//RBLD0007 EXEC PGM=DSNUTILB,PARM=(DB8W),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//UTPRINT  DD   SYSOUT=*
//SYSOUT   DD   UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN    DD   *

  REBUILD INDEX
    ( "RETOOLS"."XDATST2" )
      SCOPE ALL

/*
//*
..............
..............
```

2.5.4 Recovering table in single-table table space to ODL before current DDL version

This scenario describes the recovery of a single table RETOOLS.DATST4 in a single-table table space to a previous object definition level, that is, before the current object definition level just before the drop of the table. The timeline shown in Figure 2-107 on page 169 is assumed for this scenario. This environment is a non-data sharing environment.

The following windows only represent the ones of particular interest (in our opinion) to this scenario. The windows leading up to these windows and those that follow these windows are almost identical to the windows described in 2.3.1, “Recovering multi-table table space” on page 56. We show the partial contents of the generated JCL for a given recovery plan, but not the execution job results.

Figure 2-128 on page 193 shows the dropped table RETOOLS.DATST4 selected for recovery in the navigation pane. Its red color in the navigation pane confirms that this table is dropped. DBTST1 and DATST3 are also red in color indicating them to be dropped tables, but they are not selected for recovery. Figure 2-129 on page 194 through Figure 2-131 on page 195 show the selection of object definition level corresponding to time t1 in Figure 2-107 on page 169 from a list of the available object definition level entries from the SLR tables; not all window interactions are shown here. Figure 2-131 on page 195 shows this object definition level is translated to a timestamp corresponding to the selected object definition level entry’s “Level End Timestamp”, as shown in Figure 2-130 on page 195. Figure 2-132 on page 196 shows a warning message issued by Recovery Expert for z/OS that the selected timestamp might not be a point-of-consistency. Figure 2-133 on page 196 shows a single recovery plan generated for the requested recovery action. Figure 2-134 on page 197 shows a warning that the selected recovery plan, Plan 1, has a log analysis step in it. A portion of the generated JCL for Plan 1 is shown in Example 2-22 on page 197.

The selected recovery plan, Plan 1, has multiple steps that include the creation of the dropped table and its dependencies (such as indexes and views, if any), followed by a DSN1COPY restore of the image copy taken at time t3, as shown in Figure 2-107 on page 169, with an OBID translate to the new OBID of the object. This is followed by a rebuilding of the indexes, and a log analysis that has a boundary interval corresponding to a start RBA of 000014C4E08D, which corresponds to the “Level Create Timestamp”, and an end RBA of 000014C81000, which corresponds to the “Level End Timestamp” for the object definition level entry that are selected in Figure 2-130 on page 195. The generated SQL is then applied to complete the recovery action.

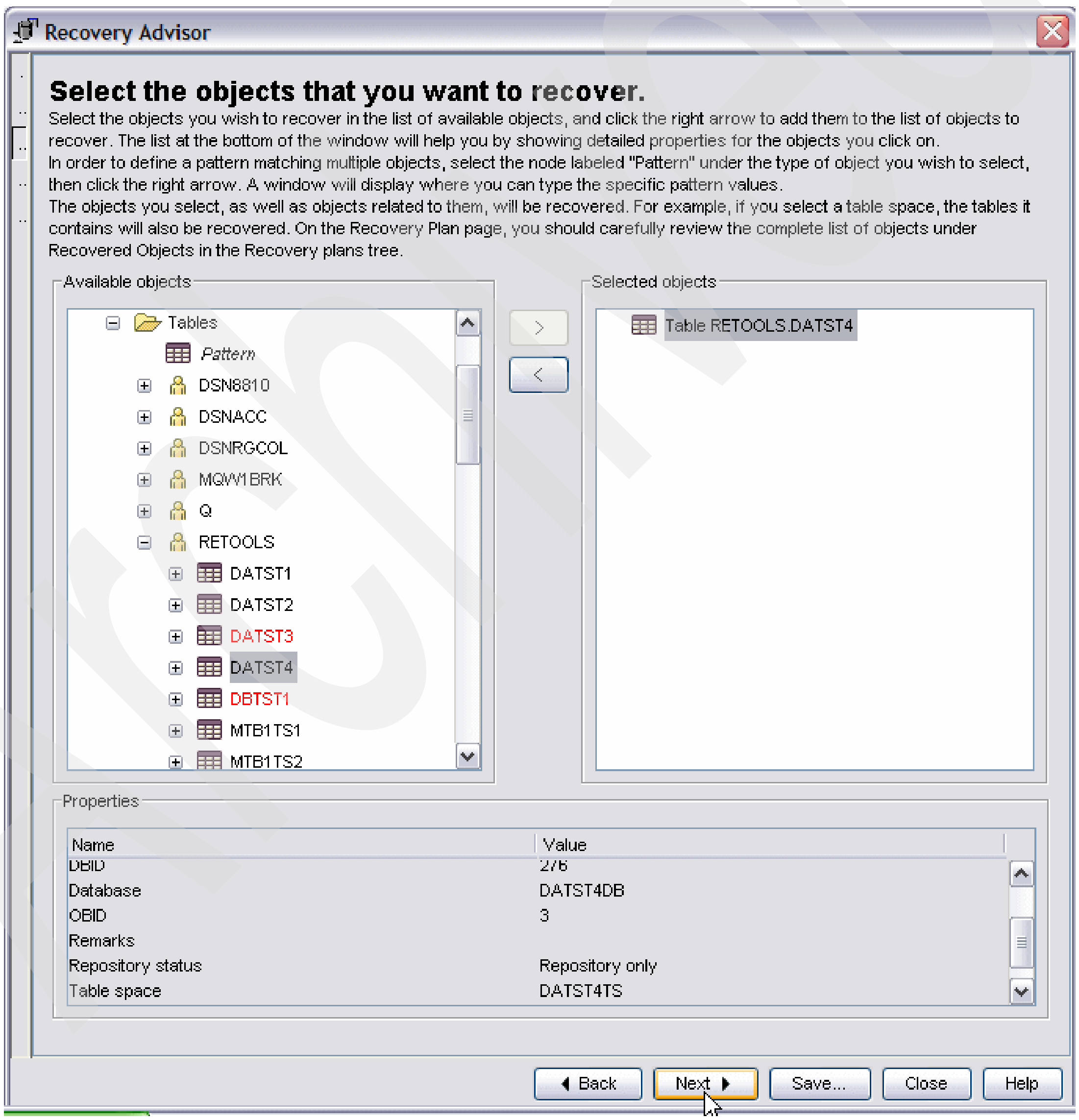

Figure 2-128 Selecting the dropped table RETOOLS.DATST4 to recover

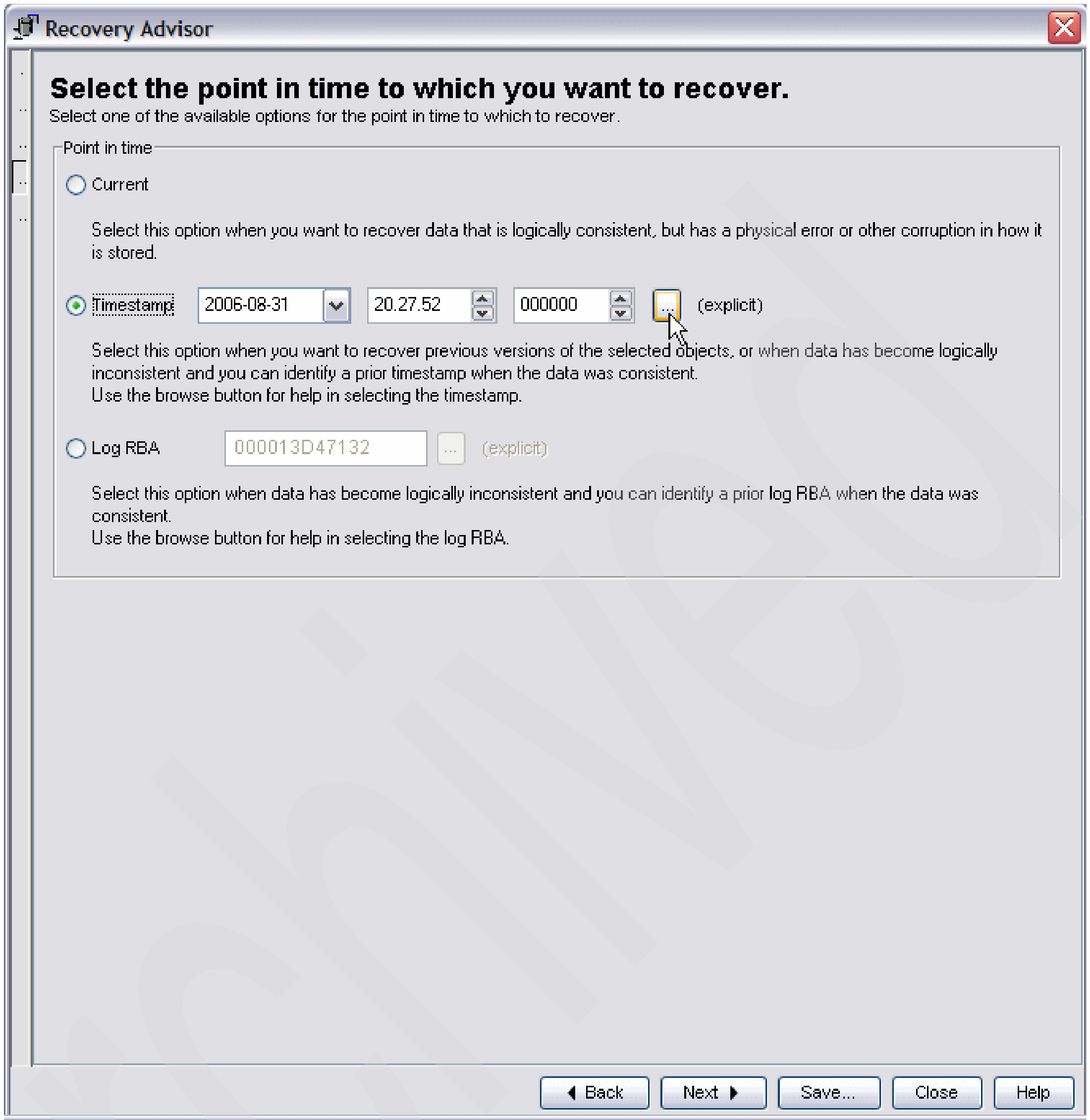

Figure 2-129 Selecting the point to recover to (1/3)

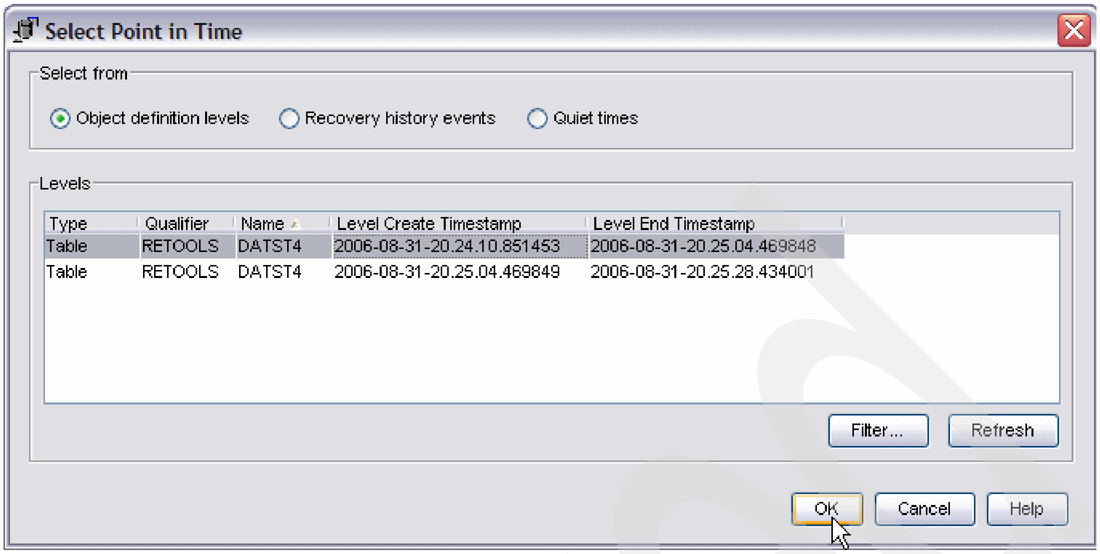

Figure 2-130 Selecting the point to recover to (2/3)

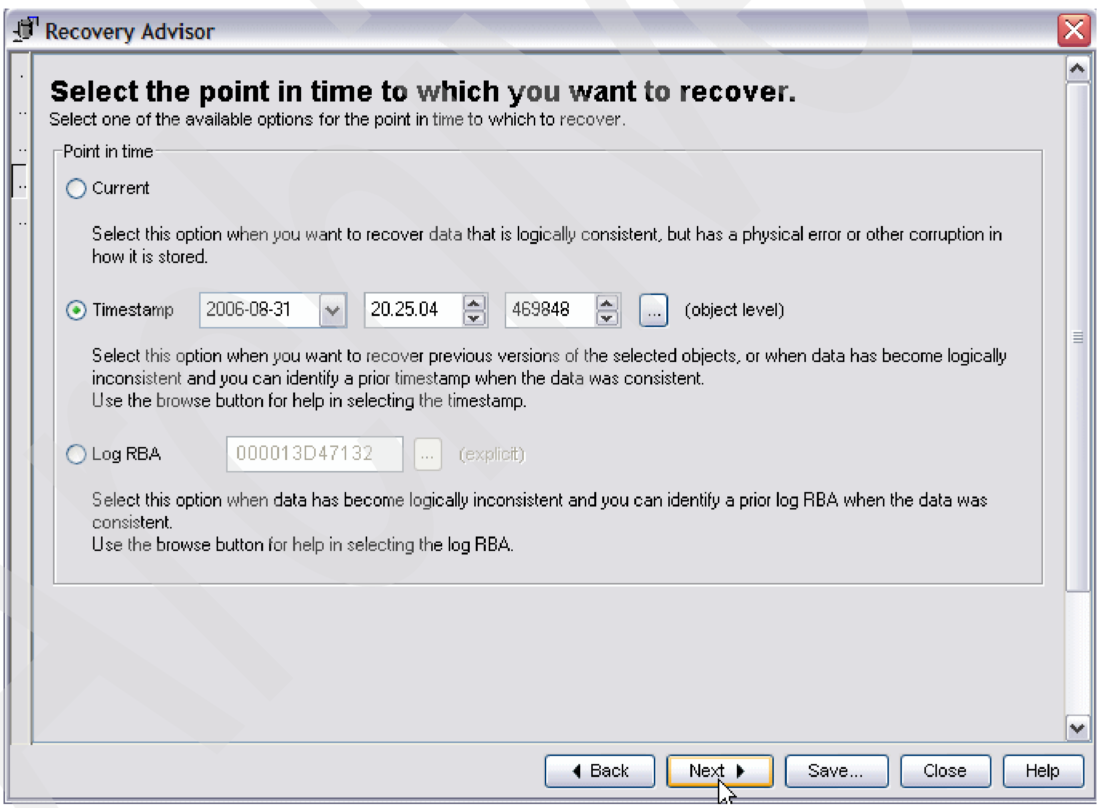

Figure 2-131 Selecting the point to recover to (3/3)

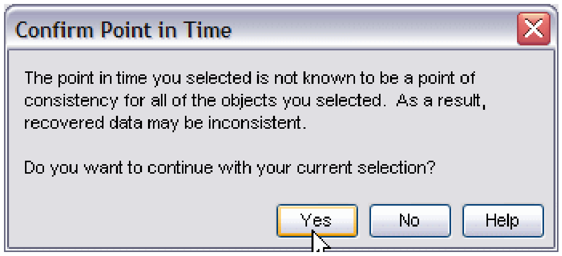

Figure 2-132 Warning message that selected timestamp might not be a point-of-consistency

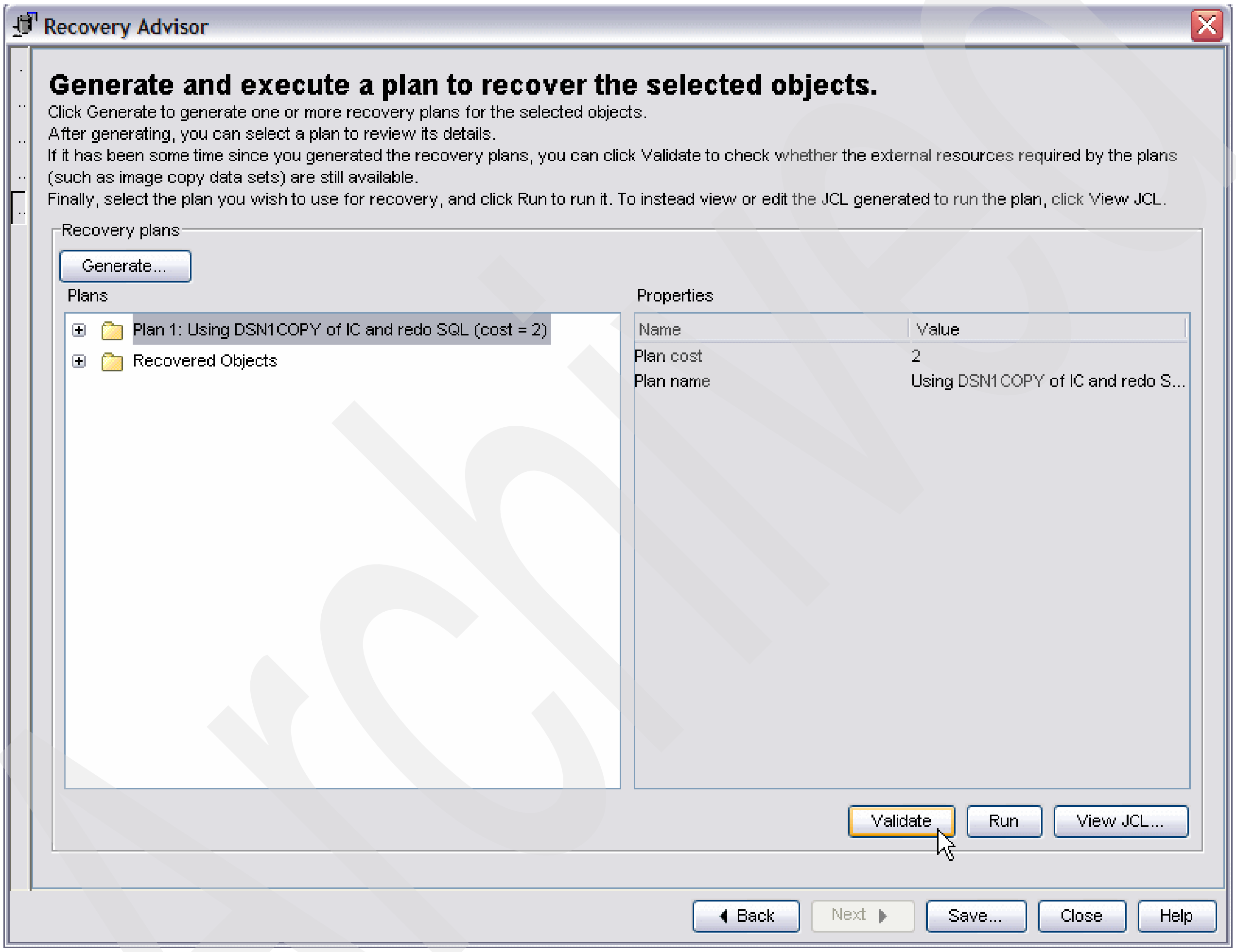

Figure 2-133 Generated recovery plans

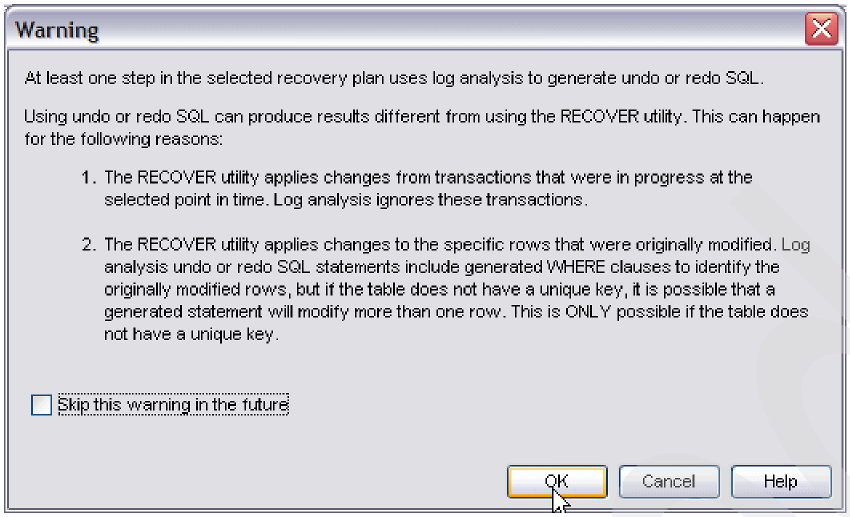

Figure 2-134 Warning message about log analysis step in selected recovery plans

Example 2-22 Partial contents of the generated JCL for Plan 1

```
//W8TBLRC JOB (999,POK),'RETEST',CLASS=A,MSGCLASS=H,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
//*
..................
..................//*
//SQCR0002 EXEC PGM=ARY#UTIL,PARM='EXECSQL,DB8W ',COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
CONFIG DEFAULT
//SYSIN    DD   *
--#SET CONTINUE ON WARNING

  SET CURRENT SQLID = 'ALICIAW';
  CREATE TABLE "RETOOLS"."DATST4"
    (DEPTNO               CHAR(3) NOT NULL FOR SBCS DATA
    ,DEPTNAME             VARCHAR(36) NOT NULL FOR SBCS DATA
    ,MGRNO                CHAR(6) FOR SBCS DATA WITH DEFAULT NULL
    ,ADMRDEPT             CHAR(3) NOT NULL FOR SBCS DATA
    ,LOCATION             CHAR(16) FOR SBCS DATA WITH DEFAULT NULL
    ,PRIMARY KEY ( DEPTNO))
    IN DATST4DB.DATST4TS
    AUDIT NONE
    CCSID EBCDIC
;
  COMMIT;

  SET CURRENT SQLID = 'RETOOLS';
  SET CURRENT PATH = "SYSIBM","SYSFUN","SYSPROC","RETOOLS";
--#SET TERMINATOR #
    CREATE VIEW RETOOLS.VDATST4
    AS SELECT ALL DEPTNO , DEPTNAME, MGRNO , ADMRDEPT
```

```
    FROM RETOOLS.DATST4#
--#SET TERMINATOR ;
  COMMIT;

  SET CURRENT SQLID = 'ALICIAW';
  CREATE UNIQUE INDEX "RETOOLS"."XDATST4"
    ON "RETOOLS"."DATST4"
    (DEPTNO ASC)
    USING STOGROUP "SMSSG"
      PRIQTY 12
      SECQTY 4
      ERASE NO
    PCTFREE 10
    PIECESIZE 2 G
    BUFFERPOOL BP0
    CLOSE NO
    COPY NO
;
  COMMIT;
/*
//*
....................
....................
//DCPX0005 EXEC PGM=ARY#UTIL,PARM='UTGENA,DB8W ',COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//SYSPRINT DD   SYSOUT=*
//CONTROLS DD   *
CONFIG DEFAULT
//SYSXLAT  DD   DSN=NALUR.ARY06243.S203505.M906570.SYSXLAT,
//             UNIT=SYSALLDA,SPACE=(CYL,(1,1),RLSE),
//             DISP=(NEW,CATLG,DELETE)
//SYSIN    DD   *
TABLESPACE
0276,DATST4DB
0002,DATST4TS
0003,"RETOOLS"."DATST4"
/*
//*
//DCPY0006 EXEC PGM=DSN1COPY,COND=(4,LT),
// PARM='CHECK,PAGESIZE(4K),FULLCOPY,OBIDXLAT,RESET'
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSUT1   DD DISP=SHR,DSN=REIC.DB8W.DATST4DB.DATST4TS.FULL.G0001V00
//SYSUT2   DD DISP=SHR,DSN=DB8WUS.DSNDBC.DATST4DB.DATST4TS.I0001.A001
//SYSPRINT DD   SYSOUT=*
//SYSXLAT  DD DISP=SHR,DSN=NALUR.ARY06243.S203505.M906570.SYSXLAT
//*
//RBLD0007 EXEC PGM=DSNUTILB,PARM=(DB8W),COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//UTPRINT  DD   SYSOUT=*
```

```
//SYSOUT   DD   UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN    DD   *

  REBUILD INDEX
    ( "RETOOLS"."XDATST4" )
      SCOPE ALL

..................
..................
//*********************************************************************
//* STEP 2: READ THE DB2 LOG TO GENERATE THE GENERAL REPORT          *
//*********************************************************************
//LASR0002 EXEC PGM=ARYGEN1,REGION=0M,COND=(4,LT)
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//MODEFILE DD   DSN=NALUR.ARYLAS.MODE.R0000523,
// DISP=OLD
//SYSOUT   DD   SYSOUT=*
//CFILES   DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//GENRPT   DD   SYSOUT=*
//EXTREP   DD   SYSOUT=*
//SUMREPT  DD   SYSOUT=*
//XDREPT   DD   SYSOUT=*
//QTRPT    DD   SYSOUT=*
//WARNINGS DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//DATAIN   DD   *
SSID           =DB8W
START RBA      =000014C4E08D
END RBA        =000014C81000
/*
//FILTERS  DD *
GENERAL FILTERS...........
SHOW UPDATES   =Y
SHOW INSERTS   =X
SHOW DELETES   =Y
SHOW ROLLBACKS =N
CATALOG DATA   =N
SHOW UNCOMMITS =N
OBJECT FILTERS-BY IDS.....
00000,00000,00000,00276,00002,00003,R,000014C4E08D,000014C81000
OBJECT FILTERS-BY NAME....
TABLE OWNER=RETOOLS
TABLE NAME =DATST4
PARTITION  =0000
SRBA       =000014C4E08D
ERBA       =000014C81000
INC/EXC    =I
/*
//**********************************
//* PARAMETERS FOR JOB SHOWN BELOW *
```

```
//**********************************
//* SSID           =DB8W
//* START RBA      =000014C4E08D
//* END RBA        =000014C81000
//* /*
//* //FILTERS  DD *
//* GENERAL FILTERS...........
//* SHOW UPDATES   =Y
//* SHOW INSERTS   =X
//* SHOW DELETES   =Y
//* SHOW ROLLBACKS =N
//* CATALOG DATA   =N
//* SHOW UNCOMMITS =N
//* OBJECT FILTERS-BY IDS.....
//* 00000,00000,00000,00276,00002,00003,R,000014C4E08D,000014C81000
//* OBJECT FILTERS-BY NAME....
//* TABLE OWNER=RETOOLS
//* TABLE NAME =DATST4
//* PARTITION  =0000
//* SRBA       =000014C4E08D
//* ERBA       =000014C81000
//* INC/EXC    =I
//*
....................
....................
//LASR0006 EXEC PGM=ARYDTL4,REGION=0M,COND=(4,LT)
//*******************************************************************
//*  STEP 6: THIS STEP DOES THE FOLLOWING:
//*         -SQL GENERATION
//*******************************************************************
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//MODEFILE DD   DSN=NALUR.ARYLAS.MODE.R0000523,
// DISP=OLD
//SQLOUT   DD   DSN=NALUR.ARY06243.S203505.M946620.EXECSQL,
//         DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,SPACE=(CYL,(50,50),RLSE),
//         DCB=(LRECL=80,BLKSIZE=23440,RECFM=FB)
//SYSOUT   DD   SYSOUT=*
//MESSAGES DD   SYSOUT=*                                                01870000
//SYSUDUMP DD   SYSOUT=*                                                01880000
//WARNINGS DD   SYSOUT=*                                                01890000
//CFILES   DD   SYSOUT=*
//*                                                                     01920000
//*
......................
......................
//LGNL0008 EXEC PGM=ARY#UTIL,PARM='EXECSQL,DB8W ',
//             COND=((4,LT),(0,NE,LASR0002))
//STEPLIB  DD DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//         DD DISP=SHR,DSN=DB8W8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
//DB2PARMS DD DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//SYSPRINT DD   SYSOUT=*
```

```
//CONTROLS DD   *
CONFIG DEFAULT
CONTINUE ON WARNING
//SYSIN    DD   DSN=NALUR.ARY06243.S203505.M946620.EXECSQL,
//             DISP=(OLD,KEEP)
//*
......................
......................
```

2.6 Recovering the entire subsystem

You can use Recovery Expert for z/OS to recover a DB2 subsystem. You can choose to recover an entire operational subsystem, including the directory, catalog, and all application spaces, to a point-in-time or to current.

> **Attention:** This support is only available when you use DB2 for z/OS version 8 or later, and you use the RESTORE SYSTEM utility. For more information about the RESTORE SYSTEM utility, see *DB2 UDB for z/OS V8 Utility Guide and Reference*, SC18-7427.

The recovery process for the entire subsystem is the same as for any DB2 object up to the Objects page in the Recovery Expert for z/OS GUI client. When on the Objects page, select the DB2 subsystem to recover. Selecting a DB2 subsystem is mutually exclusive with selecting any other DB2 objects: You cannot select it if you select any other DB2 objects, and you cannot select any other DB2 objects after selecting the DB2 subsystem. The recovery process proceeds the same as for any DB2 object. However, the generated recovery plan invokes the RESTORE SYSTEM utility to recover the entire DB2 subsystem to the selected point-in-time.

> **Note:** To use the RESTORE SYSTEM capability, ensure that one or more BACKUP SYSTEM utilities have been issued previously.

In the following sections, we describe the process of recovering a non-data sharing system and a data sharing system.

2.6.1 Recovering a non-data sharing subsystem

Before you start the recovery of a DB2 subsystem, you must have at least one instance of the execution of the BACKUP SYSTEM utility recorded in the bootstrap data set (BSDS). For this scenario, we assume that one or more BACKUP SYSTEM utilities are successfully issued and the Recovery Expert for z/OS agent is aware of the history of these utilities in the BSDS. You must meet the Data Facility Storage Management Subsystem (DFSMS) requirements for the DB2 system catalog, directory, and active logs to successfully use the BACKUP SYSTEM utility.

Figure 2-135 on page 202 shows the DB8W subsystem selected for recovery in the navigation pane. In Figure 2-136 on page 203, specify the RBA to which to recover to (we assume that you have determined what this RBA must be based on some external source of information). Figure 2-137 on page 204 shows the generated recovery plan that has two jobs in it: One that creates a conditional restart entry in the BSDS, and the second job that performs the actual restore of the backup taken with the BACKUP SYSTEM utility.
Figure 2-138 on page 205 lists the instructions that you must follow to perform a successful

recovery of the entire DB2 subsystem. Figure 2-139 on page 206 through Figure 2-144 on page 208 show the process of exporting the generated JCL to a Multiple Virtual Storage (MVS) data set for later execution. Figure 2-145 on page 208 is the Launchpad window and shows the successful recovery plan generation and validation messages in the Messages section. Example 2-23 on page 208 shows the complete execution results of the job that creates the conditional restart entry in the BSDS. Example 2-24 on page 211 shows the complete execution results of the job that restores the DB2 subsystem to the specified RBA.

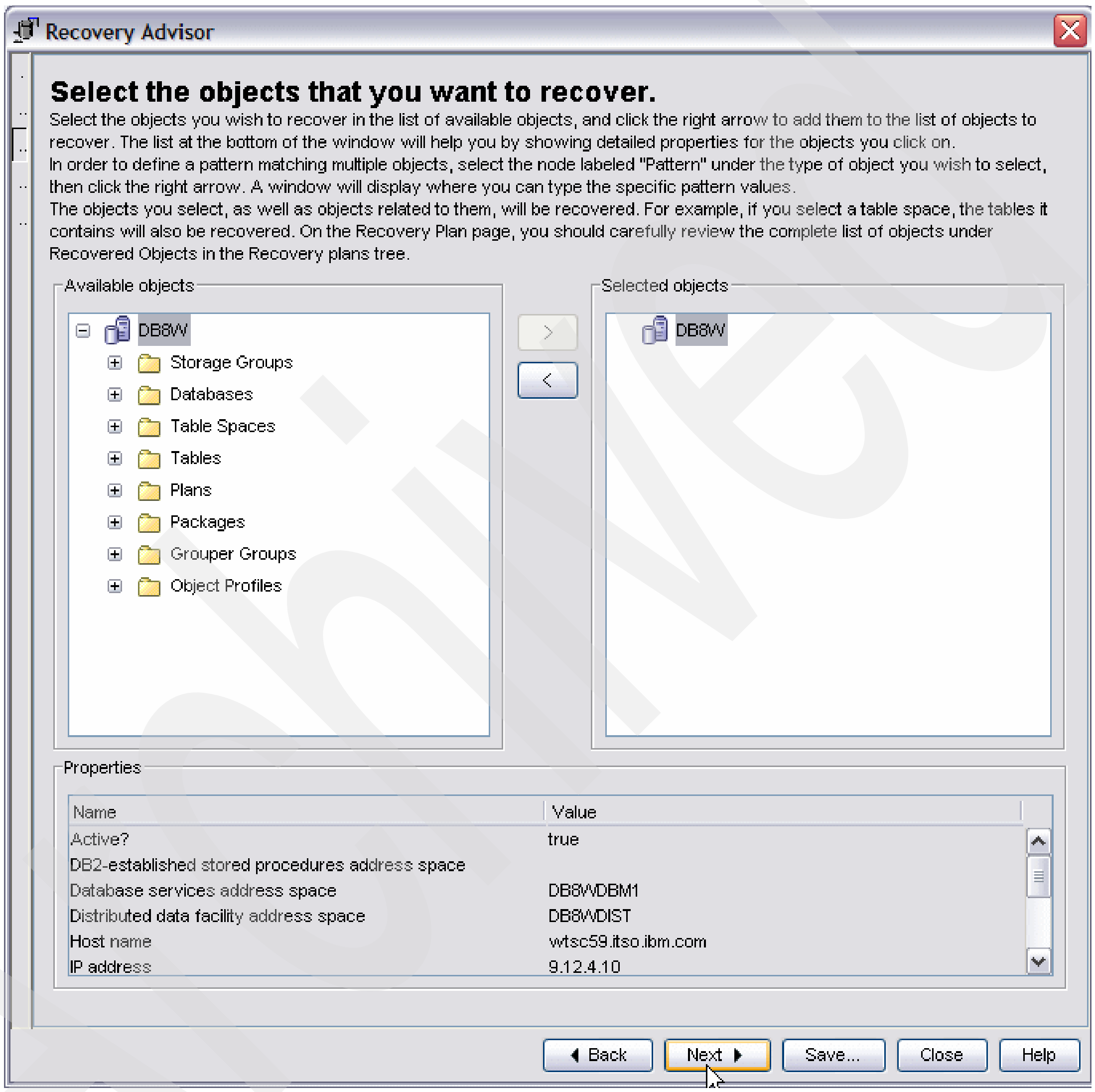

Figure 2-135 Selecting DB2 subsystem DB8W to recover

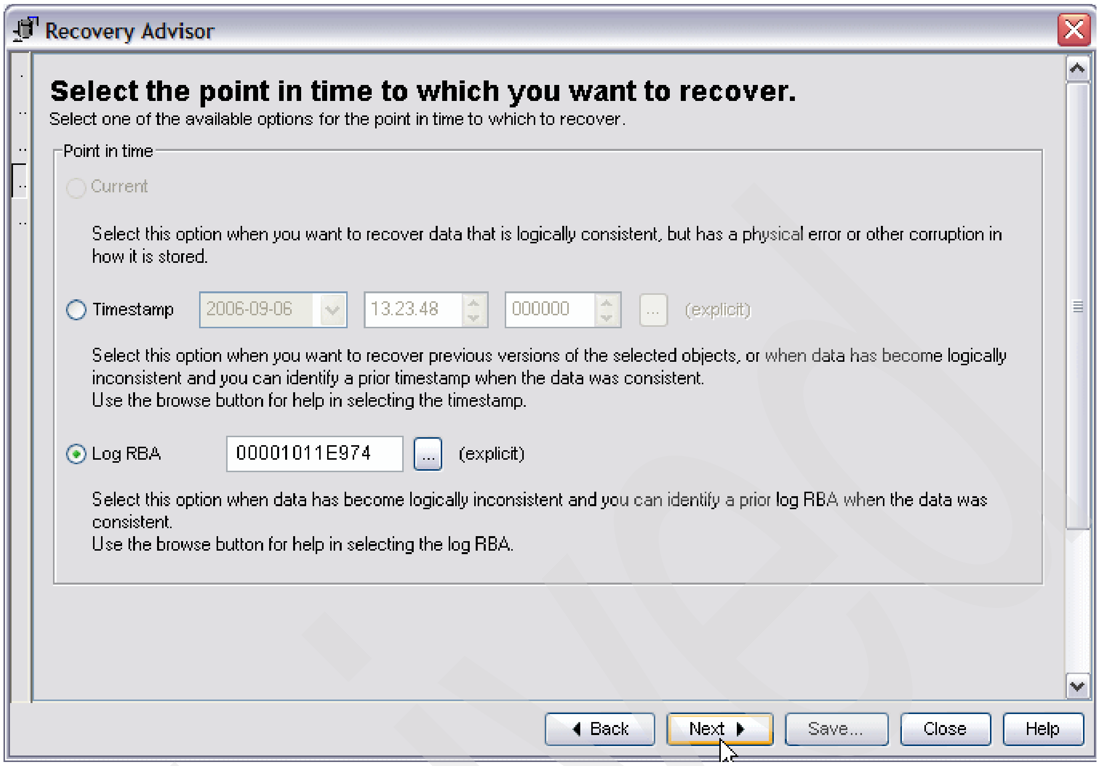

Figure 2-136 Selecting the point to recover to

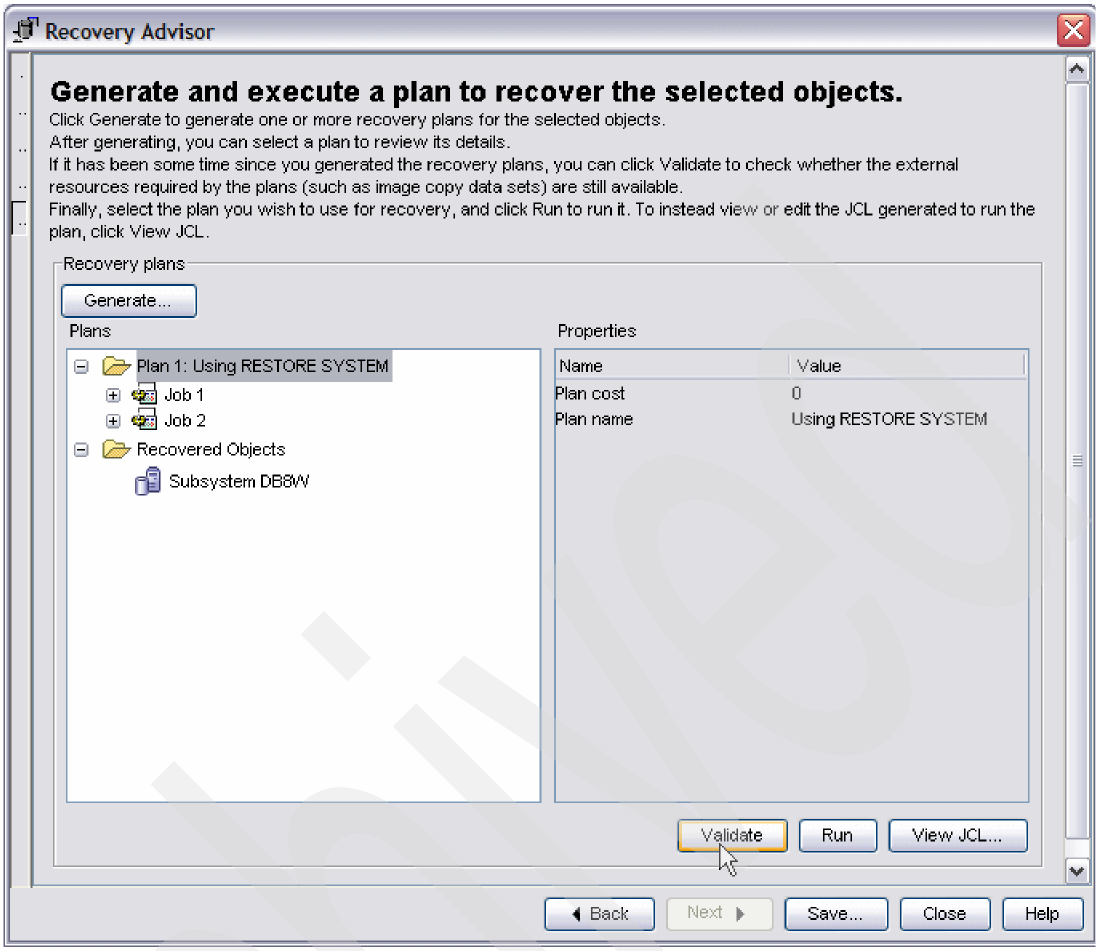

Figure 2-137 Generated recovery plans

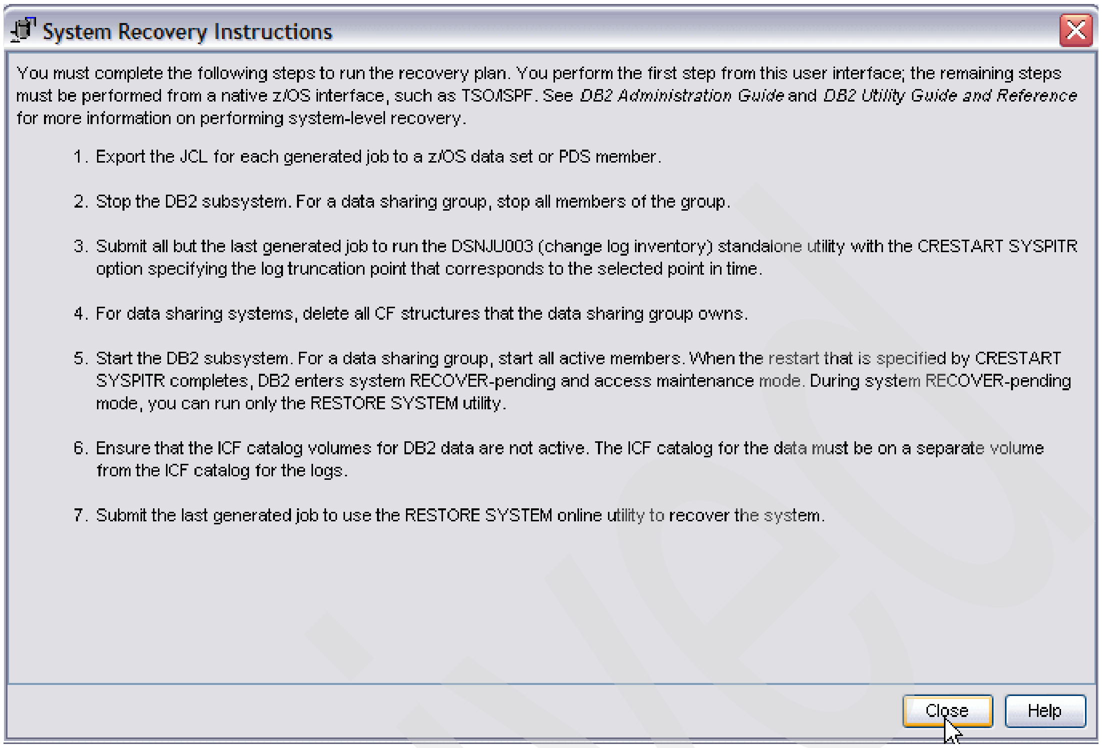

Figure 2-138 System recovery instructions

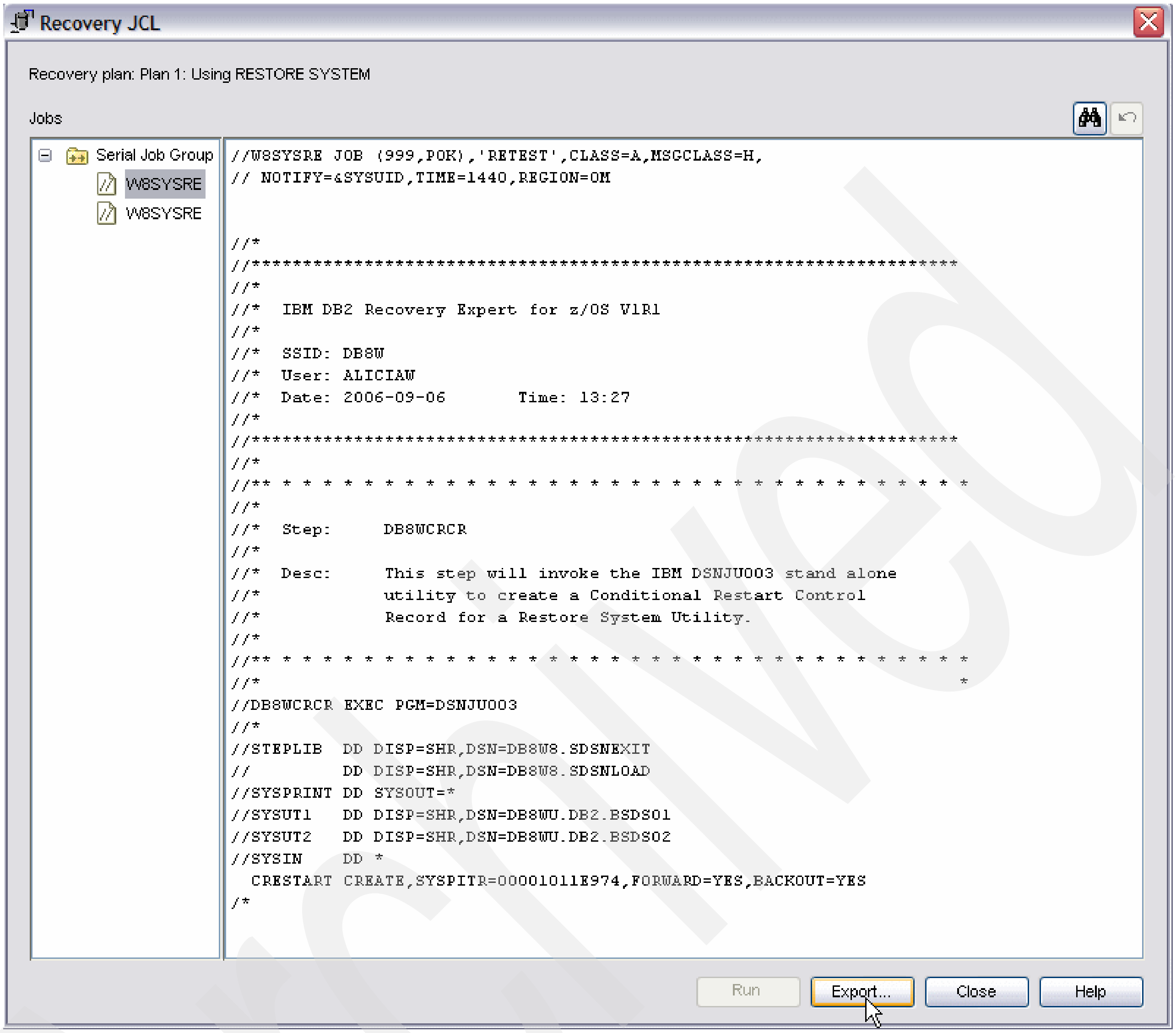

Figure 2-139   Creating conditional restart entry JCL

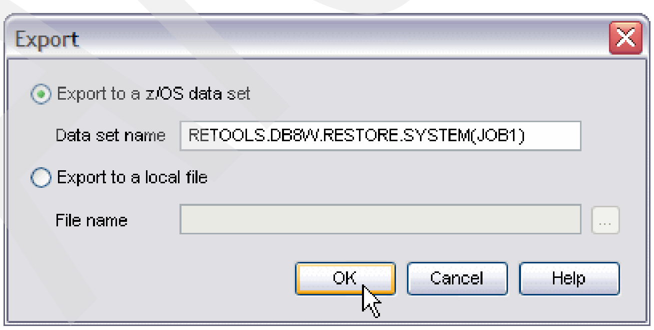

Figure 2-140   Exporting JCL to a z/OS data set

Figure 2-141 Successful export message

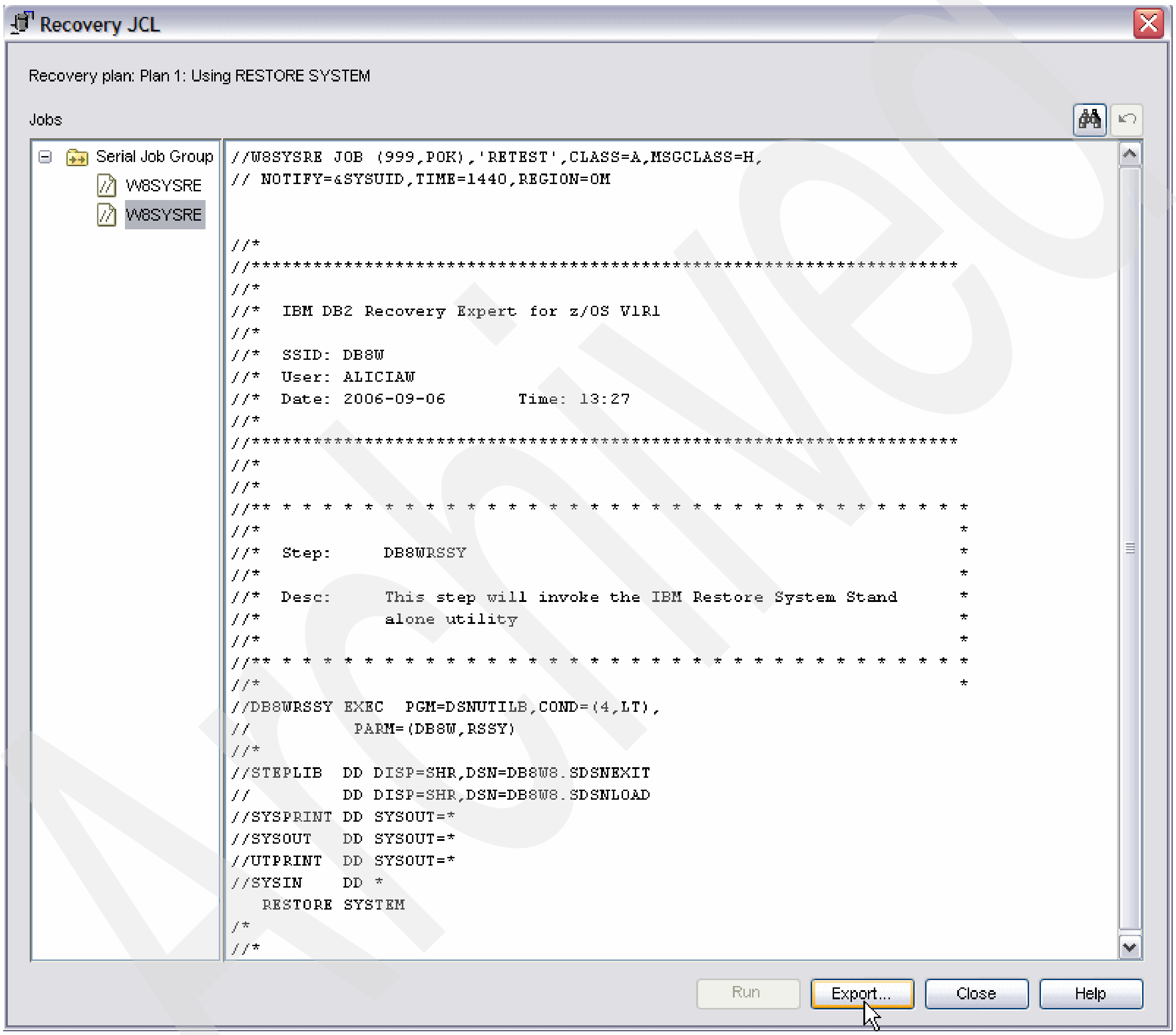

Figure 2-142 RESTORE SYSTEM JCL

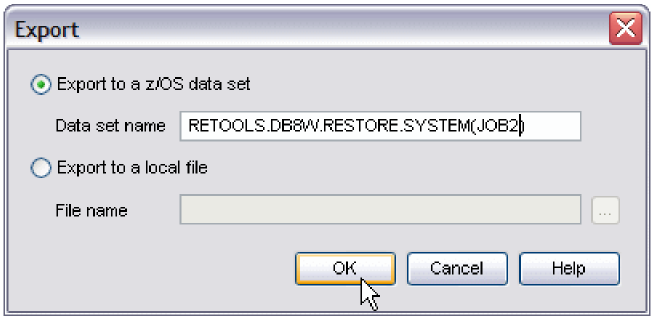

Figure 2-143 Exporting JCL to a z/OS data set

Figure 2-144 Successful export message

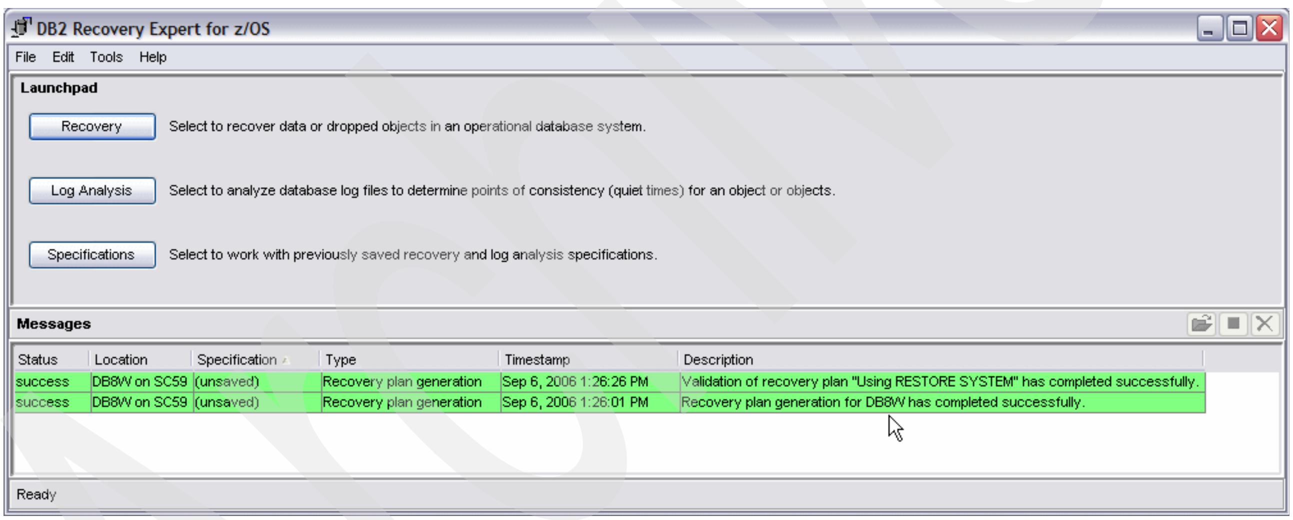

Figure 2-145 Successful recovery plan generation and validation in the messages section of launchpad

Example 2-23 Complete contents of the execution results of generated Job 1 (DSNJU003)

```
1                     J E S 2  J O B  L O G  --  S Y S T E M  I B M 2  --  N O D E  W T S C 5 9
0
 13.56.19 JOB02540 ---- WEDNESDAY, 06 SEP 2006 ----
 13.56.19 JOB02540  IRR010I  USERID ALICIAW  IS ASSIGNED TO THIS JOB.
 13.56.19 JOB02540  ICH70001I ALICIAW  LAST ACCESS AT 13:24:22 ON WEDNESDAY, SEPTEMBER 6, 2006
 13.56.19 JOB02540  $HASP373 W8SYSR1  STARTED - INIT 1    - CLASS A - SYS IBM2
 13.56.19 JOB02540  -                                        --TIMINGS (MINS.)--
----PAGING COUNTS---
 13.56.19 JOB02540  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK   SERV  PG
PAGE   SWAP    VIO SWAPS
 13.56.19 JOB02540  -W8SYSR1          DB8WCRCR    00    114    .00    .00     .0  15631   0
0      0      0     0
```

```
 13.56.19 JOB02540  -W8SYSR1  ENDED.  NAME-RETEST               TOTAL CPU TIME=   .00  TOTAL
ELAPSED TIME=    .0
 13.56.19 JOB02540  $HASP395 W8SYSR1  ENDED
0------ JES2 JOB STATISTICS ------
-  06 SEP 2006 JOB EXECUTION DATE
-           33 CARDS READ
-           78 SYSOUT PRINT RECORDS
-            0 SYSOUT PUNCH RECORDS
-            4 SYSOUT SPOOL KBYTES
-         0.00 MINUTES EXECUTION TIME
         1 //W8SYSR1 JOB (999,POK),'RETEST',CLASS=A,MSGCLASS=H,                    JOB02540
           // NOTIFY=&SYSUID,TIME=1440,REGION=0M
           //*
           //*********************************************************************
           //*
           //*  IBM DB2 Recovery Expert for z/OS V1R1
           //*
           //*  SSID: DB8W
           //*  User: ALICIAW
           //*  Date: 2006-09-06       Time: 13:27
           //*
           //*********************************************************************
           //*
           //** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
           //*
           //*  Step:     DB8WCRCR
           //*
           //*  Desc:     This step will invoke the IBM DSNJU003 stand alone
           //*            utility to create a Conditional Restart Control
           //*            Record for a Restore System Utility.
           //*
           //** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
           //*                                                                   *
           IEFC653I SUBSTITUTION JCL -
(999,POK),'RETEST',CLASS=A,MSGCLASS=H,NOTIFY=ALICIAW,TIME=1440,REGION=0M
         2 //DB8WCRCR EXEC PGM=DSNJU003
           //*
         3 //STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
         4 //         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
         5 //SYSPRINT DD SYSOUT=*
         6 //SYSUT1   DD DISP=SHR,DSN=DB8WU.DB2.BSDS01
         7 //SYSUT2   DD DISP=SHR,DSN=DB8WU.DB2.BSDS02
         8 //SYSIN    DD *
 ICH70001I ALICIAW  LAST ACCESS AT 13:24:22 ON WEDNESDAY, SEPTEMBER 6, 2006
 IEF236I ALLOC. FOR W8SYSR1 DB8WCRCR
 IGD103I SMS ALLOCATED TO DDNAME STEPLIB
 IGD103I SMS ALLOCATED TO DDNAME
 IEF237I JES2 ALLOCATED TO SYSPRINT
 IGD103I SMS ALLOCATED TO DDNAME SYSUT1
 IGD103I SMS ALLOCATED TO DDNAME SYSUT2
 IEF237I JES2 ALLOCATED TO SYSIN
 IEF142I W8SYSR1 DB8WCRCR - STEP WAS EXECUTED - COND CODE 0000
 IGD104I DB8W8.SDSNEXIT                               RETAINED,  DDNAME=STEPLIB
 IGD104I DB8W8.SDSNLOAD                               RETAINED,  DDNAME=
 IEF285I   ALICIAW.W8SYSR1.JOB02540.D0000102.?          SYSOUT
```

```
 IGD104I DB8WU.DB2.BSDS01                             RETAINED,  DDNAME=SYSUT1
 IGD104I DB8WU.DB2.BSDS02                             RETAINED,  DDNAME=SYSUT2
 IEF285I   ALICIAW.W8SYSR1.JOB02540.D0000101.?          SYSIN
 IEF373I STEP/DB8WCRCR/START 2006249.1356
 IEF374I STEP/DB8WCRCR/STOP  2006249.1356 CPU    0MIN 00.01SEC SRB    0MIN 00.00SEC VIRT   244K
SYS   260K EXT       8K SYS   10536K
 IEF375I  JOB/W8SYSR1 /START 2006249.1356
 IEF376I  JOB/W8SYSR1 /STOP  2006249.1356 CPU    0MIN 00.01SEC SRB    0MIN 00.00SEC
1DSNJCNVB CONVERSION PROGRAM HAS NOT RUN   DDNAME=SYSUT1
 DSNJCNVB CONVERSION PROGRAM HAS NOT RUN   DDNAME=SYSUT2
0  CRESTART CREATE,SYSPITR=00001011E974,FORWARD=YES,BACKOUT=YES
 DSNJ408I  DSNRJFCK CHECKPOINT RBA FOUND, RBA = 00000FDE9090, TIME = 11:07:15 SEPTEMBER 05, 2006
 DSNJ411I  DSNRJRCR CRESTART CREATE FOR CRCRID = 0001, DDNAME = SYSUT1
 DSNJ408I  DSNRJFCK CHECKPOINT RBA FOUND, RBA = 00000FDE9090, TIME = 11:07:15 SEPTEMBER 05, 2006
 DSNJ411I  DSNRJRCR CRESTART CREATE FOR CRCRID = 0001, DDNAME = SYSUT2
 DSNJ225I  CRESTART OPERATION COMPLETED SUCCESSFULLY
-DSNJ200I  DSNJU003 CHANGE LOG INVENTORY UTILITY PROCESSING COMPLETED SUCCESSFULLY
1                    J E S 2  J O B  L O G  --  S Y S T E M  I B M 2  --  N O D E  W T S C 5 9
0
 13.56.19 JOB02540 ---- WEDNESDAY, 06 SEP 2006 ----
 13.56.19 JOB02540  IRR010I  USERID ALICIAW  IS ASSIGNED TO THIS JOB.
 13.56.19 JOB02540  ICH70001I ALICIAW  LAST ACCESS AT 13:24:22 ON WEDNESDAY, SEPTEMBER 6, 2006
 13.56.19 JOB02540  $HASP373 W8SYSR1  STARTED - INIT 1    - CLASS A - SYS IBM2
 13.56.19 JOB02540  -                                        --TIMINGS (MINS.)--
----PAGING COUNTS---
 13.56.19 JOB02540  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK   SERV  PG
PAGE   SWAP    VIO SWAPS
 13.56.19 JOB02540  -W8SYSR1          DB8WCRCR    00    114    .00    .00     .0  15631   0
0      0      0     0
 13.56.19 JOB02540  -W8SYSR1  ENDED.  NAME-RETEST               TOTAL CPU TIME=   .00  TOTAL
ELAPSED TIME=    .0
 13.56.19 JOB02540  $HASP395 W8SYSR1  ENDED
0------ JES2 JOB STATISTICS ------
-  06 SEP 2006 JOB EXECUTION DATE
-           33 CARDS READ
-           78 SYSOUT PRINT RECORDS
-            0 SYSOUT PUNCH RECORDS
-            4 SYSOUT SPOOL KBYTES
-         0.00 MINUTES EXECUTION TIME
         1 //W8SYSR1 JOB (999,POK),'RETEST',CLASS=A,MSGCLASS=H,                    JOB02540
           // NOTIFY=&SYSUID,TIME=1440,REGION=0M
           //*
           //*******************************************************************
           //*
           //*  IBM DB2 Recovery Expert for z/OS V1R1
           //*
           //*  SSID: DB8W
           //*  User: ALICIAW
           //*  Date: 2006-09-06       Time: 13:27
           //*
           //*******************************************************************
           //*
           //** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
           //*
           //*  Step:     DB8WCRCR
```

```
          //*
          //*  Desc:     This step will invoke the IBM DSNJU003 stand alone
          //*            utility to create a Conditional Restart Control
          //*            Record for a Restore System Utility.
          //*
          //** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
          //*                                                               *
          IEFC653I SUBSTITUTION JCL -
(999,POK),'RETEST',CLASS=A,MSGCLASS=H,NOTIFY=ALICIAW,TIME=1440,REGION=0M
        2 //DB8WCRCR EXEC PGM=DSNJU003
          //*
        3 //STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
        4 //         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
        5 //SYSPRINT DD SYSOUT=*
        6 //SYSUT1   DD DISP=SHR,DSN=DB8WU.DB2.BSDS01
        7 //SYSUT2   DD DISP=SHR,DSN=DB8WU.DB2.BSDS02
        8 //SYSIN    DD *
 ICH70001I ALICIAW  LAST ACCESS AT 13:24:22 ON WEDNESDAY, SEPTEMBER 6, 2006
 IEF236I ALLOC. FOR W8SYSR1 DB8WCRCR
 IGD103I SMS ALLOCATED TO DDNAME STEPLIB
 IGD103I SMS ALLOCATED TO DDNAME
 IEF237I JES2 ALLOCATED TO SYSPRINT
 IGD103I SMS ALLOCATED TO DDNAME SYSUT1
 IGD103I SMS ALLOCATED TO DDNAME SYSUT2
 IEF237I JES2 ALLOCATED TO SYSIN
 IEF142I W8SYSR1 DB8WCRCR - STEP WAS EXECUTED - COND CODE 0000
 IGD104I DB8W8.SDSNEXIT                              RETAINED,  DDNAME=STEPLIB
 IGD104I DB8W8.SDSNLOAD                              RETAINED,  DDNAME=
 IEF285I   ALICIAW.W8SYSR1.JOB02540.D0000102.?         SYSOUT
 IGD104I DB8WU.DB2.BSDS01                            RETAINED,  DDNAME=SYSUT1
 IGD104I DB8WU.DB2.BSDS02                            RETAINED,  DDNAME=SYSUT2
 IEF285I   ALICIAW.W8SYSR1.JOB02540.D0000101.?         SYSIN
 IEF373I STEP/DB8WCRCR/START 2006249.1356
 IEF374I STEP/DB8WCRCR/STOP  2006249.1356 CPU    0MIN 00.01SEC SRB    0MIN 00.00SEC VIRT   244K
SYS   260K EXT       8K SYS   10536K
 IEF375I  JOB/W8SYSR1 /START 2006249.1356
 IEF376I  JOB/W8SYSR1 /STOP  2006249.1356 CPU    0MIN 00.01SEC SRB    0MIN 00.00SEC
1DSNJCNVB CONVERSION PROGRAM HAS NOT RUN   DDNAME=SYSUT1
 DSNJCNVB CONVERSION PROGRAM HAS NOT RUN   DDNAME=SYSUT2
0  CRESTART CREATE,SYSPITR=00001011E974,FORWARD=YES,BACKOUT=YES
 DSNJ408I  DSNRJFCK CHECKPOINT RBA FOUND, RBA = 00000FDE9090, TIME = 11:07:15 SEPTEMBER 05, 2006
 DSNJ411I  DSNRJRCR CRESTART CREATE FOR CRCRID = 0001, DDNAME = SYSUT1
 DSNJ408I  DSNRJFCK CHECKPOINT RBA FOUND, RBA = 00000FDE9090, TIME = 11:07:15 SEPTEMBER 05, 2006
 DSNJ411I  DSNRJRCR CRESTART CREATE FOR CRCRID = 0001, DDNAME = SYSUT2
 DSNJ225I  CRESTART OPERATION COMPLETED SUCCESSFULLY
-DSNJ200I  DSNJU003 CHANGE LOG INVENTORY UTILITY PROCESSING COMPLETED SUCCESSFULLY
```

Example 2-24   Complete contents of the execution results of generated Job 2 (RESTORE SYSTEM utility)

```
1                    J E S 2  J O B  L O G  --  S Y S T E M  I B M 2  --  N O D E  W T S C 5 9
0
 23.59.28 JOB02546 ---- WEDNESDAY, 06 SEP 2006 ----
 23.59.28 JOB02546  IRR010I  USERID WELLIE2  IS ASSIGNED TO THIS JOB.
 23.59.28 JOB02546  ICH70001I WELLIE2  LAST ACCESS AT 20:04:59 ON WEDNESDAY, SEPTEMBER 6, 2006
 23.59.28 JOB02546  $HASP373 W8SYSR2  STARTED - INIT 1    - CLASS A - SYS IBM2
```

```
 23.59.36 JOB02546  -                                         --TIMINGS (MINS.)--
----PAGING COUNTS---
 23.59.36 JOB02546  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK   SERV  PG
PAGE   SWAP   VIO SWAPS
 23.59.36 JOB02546  -W8SYSR2          DB8WRSSY    00    517    .00    .00     .1   322K   0
0      0      0     0
 23.59.36 JOB02546  -W8SYSR2  ENDED.  NAME-RETEST              TOTAL CPU TIME=   .00  TOTAL
ELAPSED TIME=    .1
 23.59.36 JOB02546  $HASP395 W8SYSR2  ENDED
0------ JES2 JOB STATISTICS ------
-  06 SEP 2006 JOB EXECUTION DATE
-           35 CARDS READ
-           86 SYSOUT PRINT RECORDS
-            0 SYSOUT PUNCH RECORDS
-            5 SYSOUT SPOOL KBYTES
-         0.13 MINUTES EXECUTION TIME
         1 //W8SYSR2 JOB (999,POK),'RETEST',CLASS=A,MSGCLASS=H,                    JOB02546
           // NOTIFY=&SYSUID,TIME=1440,REGION=0M
           //*
           //*********************************************************************
           //*
           //*  IBM DB2 Recovery Expert for z/OS V1R1
           //*
           //*  SSID: DB8W
           //*  User: ALICIAW
           //*  Date: 2006-09-06       Time: 13:27
           //*
           //*********************************************************************
           //*
           //*
           //** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
           //*                                                                    *
           //*  Step:     DB8WRSSY                                                *
           //*                                                                    *
           //*  Desc:     This step will invoke the IBM Restore System Stand      *
           //*            alone utility                                           *
           //*                                                                    *
           //** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
           //*                                                                    *
           IEFC653I SUBSTITUTION JCL -
(999,POK),'RETEST',CLASS=A,MSGCLASS=H,NOTIFY=WELLIE2,TIME=1440,REGION=0M
         2 //DB8WRSSY EXEC  PGM=DSNUTILB,COND=(4,LT),
           //           PARM=(DB8W,RSSY)
           //*
         3 //STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
         4 //         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
         5 //SYSPRINT DD SYSOUT=*
         6 //SYSOUT   DD SYSOUT=*
         7 //UTPRINT  DD SYSOUT=*
         8 //SYSIN    DD *
           //*
 ICH70001I WELLIE2  LAST ACCESS AT 20:04:59 ON WEDNESDAY, SEPTEMBER 6, 2006
 IEF236I ALLOC. FOR W8SYSR2 DB8WRSSY
 IGD103I SMS ALLOCATED TO DDNAME STEPLIB
 IGD103I SMS ALLOCATED TO DDNAME
```

```
 IEF237I JES2 ALLOCATED TO SYSPRINT
 IEF237I JES2 ALLOCATED TO SYSOUT
 IEF237I JES2 ALLOCATED TO UTPRINT
 IEF237I JES2 ALLOCATED TO SYSIN
 IEF142I W8SYSR2 DB8WRSSY - STEP WAS EXECUTED - COND CODE 0000
 IGD104I DB8W8.SDSNEXIT                               RETAINED,  DDNAME=STEPLIB
 IGD104I DB8W8.SDSNLOAD                               RETAINED,  DDNAME=
 IEF285I   WELLIE2.W8SYSR2.JOB02546.D0000102.?          SYSOUT
 IEF285I   WELLIE2.W8SYSR2.JOB02546.D0000103.?          SYSOUT
 IEF285I   WELLIE2.W8SYSR2.JOB02546.D0000104.?          SYSOUT
 IEF285I   WELLIE2.W8SYSR2.JOB02546.D0000101.?          SYSIN
 IEF373I STEP/DB8WRSSY/START 2006249.2359
 IEF374I STEP/DB8WRSSY/STOP  2006249.2359 CPU    0MIN 00.13SEC SRB    0MIN 00.00SEC VIRT  1808K
SYS   244K EXT      20K SYS   10644K
 IEF375I  JOB/W8SYSR2 /START 2006249.2359
 IEF376I  JOB/W8SYSR2 /STOP  2006249.2359 CPU    0MIN 00.13SEC SRB    0MIN 00.00SEC
1DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RSSY
 DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
0DSNU050I    DSNUGUTC -  RESTORE SYSTEM
 DSNU1606I   DSNUVBRD - RESTORE SYSTEM UTILITY STARTING,
                       COPYPOOL = DSN$DB8W$DB
                       TOKEN = X'C4C2F8E6BF5DB563EED73A8600000FDE9090'.
 DSNU1627I   DSNUVBRD - RESTORE SYSTEM PRE-LOG APPLY COMPLETED SUCCESSFULLY,
                       COPYPOOL = DSN$DB8W$DB
                       TOKEN = X'C4C2F8E6BF5DB563EED73A8600000FDE9090'
                       ELAPSED TIME = 00:00:00.
 DSNU1604I -DB8W DSNUVARL - RESTORE SYSTEM PHASE LOG APPLY STARTED AT LOG POINT =
X'00000FDE9090'.
 DSNU1635I -DB8W DSNUVARL - THE RBA RANGE FOR THE LAST CHECKPOINT ISSUED DURING THE LOGAPPLY
PHASE OF THE RESTORE SYSTEM
 UTILITY IS START_RBA = X'00001012142C' END_RBA = X'000010123578'
 DSNU1628I   DSNUVBRD - RESTORE SYSTEM PHASE LOG APPLY COMPLETED, ELAPSED TIME = 00:00:07.
 DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
1                     J E S 2  J O B  L O G  --  S Y S T E M  I B M 2  --  N O D E  W T S C 5 9
0
 23.59.28 JOB02546 ---- WEDNESDAY, 06 SEP 2006 ----
 23.59.28 JOB02546  IRR010I  USERID WELLIE2  IS ASSIGNED TO THIS JOB.
 23.59.28 JOB02546  ICH70001I WELLIE2  LAST ACCESS AT 20:04:59 ON WEDNESDAY, SEPTEMBER 6, 2006
 23.59.28 JOB02546  $HASP373 W8SYSR2  STARTED - INIT 1    - CLASS A - SYS IBM2
 23.59.36 JOB02546  -                                        --TIMINGS (MINS.)--
----PAGING COUNTS---
 23.59.36 JOB02546  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK   SERV  PG
PAGE   SWAP    VIO SWAPS
 23.59.36 JOB02546  -W8SYSR2          DB8WRSSY    00    517    .00    .00     .1   322K   0
0      0      0     0
 23.59.36 JOB02546  -W8SYSR2  ENDED.  NAME-RETEST               TOTAL CPU TIME=   .00  TOTAL
ELAPSED TIME=    .1
 23.59.36 JOB02546  $HASP395 W8SYSR2  ENDED
0------ JES2 JOB STATISTICS ------
-  06 SEP 2006 JOB EXECUTION DATE
-           35 CARDS READ
-           86 SYSOUT PRINT RECORDS
-            0 SYSOUT PUNCH RECORDS
-            5 SYSOUT SPOOL KBYTES
-         0.13 MINUTES EXECUTION TIME
```

```
        1 //W8SYSR2 JOB (999,POK),'RETEST',CLASS=A,MSGCLASS=H,                    JOB02546
          // NOTIFY=&SYSUID,TIME=1440,REGION=0M
          //*
          //*********************************************************************
          //*
          //*  IBM DB2 Recovery Expert for z/OS V1R1
          //*
          //*  SSID: DB8W
          //*  User: ALICIAW
          //*  Date: 2006-09-06       Time: 13:27
          //*
          //*********************************************************************
          //*
          //*
          //** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
          //*                                                                    *
          //*  Step:     DB8WRSSY                                                *
          //*                                                                    *
          //*  Desc:     This step will invoke the IBM Restore System Stand      *
          //*            alone utility                                           *
          //*                                                                    *
          //** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
          //*                                                                    *
          IEFC653I SUBSTITUTION JCL -
(999,POK),'RETEST',CLASS=A,MSGCLASS=H,NOTIFY=WELLIE2,TIME=1440,REGION=0M
        2 //DB8WRSSY EXEC  PGM=DSNUTILB,COND=(4,LT),
          //          PARM=(DB8W,RSSY)
          //*
        3 //STEPLIB  DD DISP=SHR,DSN=DB8W8.SDSNEXIT
        4 //         DD DISP=SHR,DSN=DB8W8.SDSNLOAD
        5 //SYSPRINT DD SYSOUT=*
        6 //SYSOUT   DD SYSOUT=*
        7 //UTPRINT  DD SYSOUT=*
        8 //SYSIN    DD *
          //*
 ICH70001I WELLIE2  LAST ACCESS AT 20:04:59 ON WEDNESDAY, SEPTEMBER 6, 2006
 IEF236I ALLOC. FOR W8SYSR2 DB8WRSSY
 IGD103I SMS ALLOCATED TO DDNAME STEPLIB
 IGD103I SMS ALLOCATED TO DDNAME
 IEF237I JES2 ALLOCATED TO SYSPRINT
 IEF237I JES2 ALLOCATED TO SYSOUT
 IEF237I JES2 ALLOCATED TO UTPRINT
 IEF237I JES2 ALLOCATED TO SYSIN
 IEF142I W8SYSR2 DB8WRSSY - STEP WAS EXECUTED - COND CODE 0000
 IGD104I DB8W8.SDSNEXIT                               RETAINED,  DDNAME=STEPLIB
 IGD104I DB8W8.SDSNLOAD                               RETAINED,  DDNAME=
 IEF285I   WELLIE2.W8SYSR2.JOB02546.D0000102.?          SYSOUT
 IEF285I   WELLIE2.W8SYSR2.JOB02546.D0000103.?          SYSOUT
 IEF285I   WELLIE2.W8SYSR2.JOB02546.D0000104.?          SYSOUT
 IEF285I   WELLIE2.W8SYSR2.JOB02546.D0000101.?          SYSIN
 IEF373I STEP/DB8WRSSY/START 2006249.2359
 IEF374I STEP/DB8WRSSY/STOP  2006249.2359 CPU    0MIN 00.13SEC SRB    0MIN 00.00SEC VIRT  1808K
SYS   244K EXT      20K SYS   10644K
 IEF375I  JOB/W8SYSR2 /START 2006249.2359
 IEF376I  JOB/W8SYSR2 /STOP  2006249.2359 CPU    0MIN 00.13SEC SRB    0MIN 00.00SEC
```

```
1DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RSSY
 DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
0DSNU050I    DSNUGUTC -  RESTORE SYSTEM
 DSNU1606I   DSNUVBRD - RESTORE SYSTEM UTILITY STARTING,
                        COPYPOOL = DSN$DB8W$DB
                        TOKEN = X'C4C2F8E6BF5DB563EED73A8600000FDE9090'.
 DSNU1627I   DSNUVBRD - RESTORE SYSTEM PRE-LOG APPLY COMPLETED SUCCESSFULLY,
                        COPYPOOL = DSN$DB8W$DB
                        TOKEN = X'C4C2F8E6BF5DB563EED73A8600000FDE9090'
                        ELAPSED TIME = 00:00:00.
 DSNU1604I -DB8W DSNUVARL - RESTORE SYSTEM PHASE LOG APPLY STARTED AT LOG POINT =
X'00000FDE9090'.
 DSNU1635I -DB8W DSNUVARL - THE RBA RANGE FOR THE LAST CHECKPOINT ISSUED DURING THE LOGAPPLY
PHASE OF THE RESTORE SYSTEM
 UTILITY IS START_RBA = X'00001012142C' END_RBA = X'000010123578'
 DSNU1628I   DSNUVBRD - RESTORE SYSTEM PHASE LOG APPLY COMPLETED, ELAPSED TIME = 00:00:07.
 DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

2.6.2 Recovering a data sharing subsystem

As in the case of recovering a non-data sharing subsystem, you require at least one instance of the execution of the BACKUP SYSTEM utility recorded in the BSDS. The data sharing environment is the one used in our scenarios, as shown in Figure 2-1 on page 48. The data sharing group D8FG has two members, D8F1 that resides on LPAR SC53, and D8F2 that resides on LPAR SC67.

Figure 2-146 lists the steps to recover a data sharing subsystem.

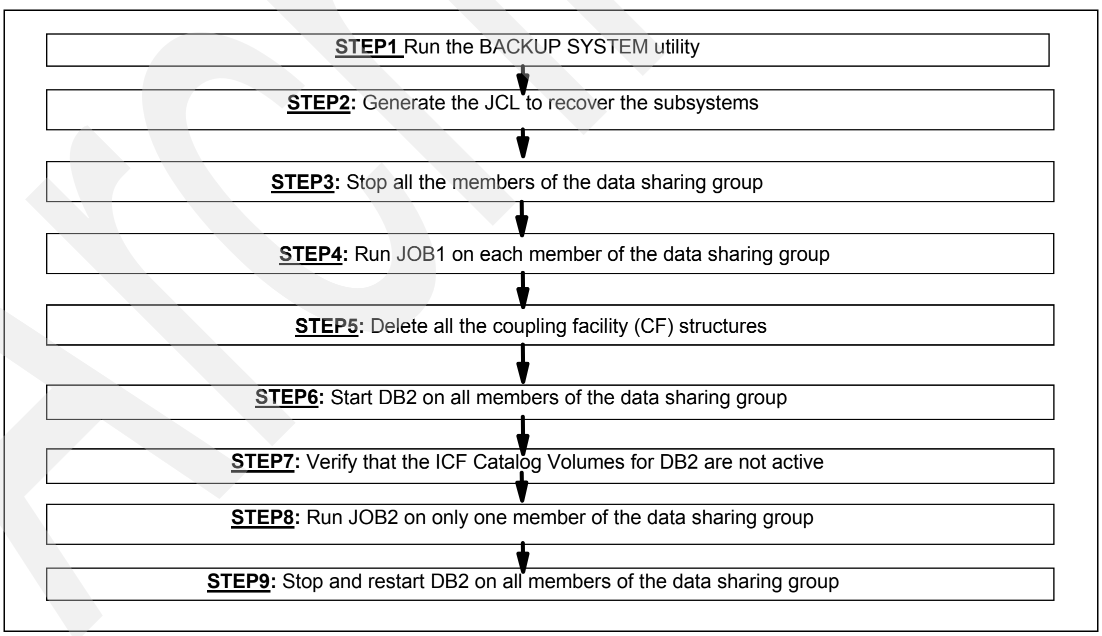

Figure 2-146 Overview of steps to recover a data sharing system

The following list provides more details about each of the steps shown in Figure 2-146.

1. Run the BACKUP SYSTEM utility.

   We used the BACKUP SYSTEM FULL syntax, which caused both the database copy pool and log copy pool to be backed up. IBM DFSMShsm™ tracks all copies made of any copy pool. Currently, all backup versions of a copy pool have to reside on DASD. The BACKUP SYSTEM executions are recorded in the BSDS of each DB2 member. The BACKUP SYSTEM JCL is shown in Example 2-25, and the output of BACKUP SYSTEM is shown in Example 2-26.

Example 2-25 BACKUP SYSTEM JCL

```
//DB8FBKUP JOB (999,POK),'BACKUP SYSTEM',CLASS=A,MSGCLASS=H,
// NOTIFY=&HENDRIK,TIME=1440,REGION=0M
/*JOBPARM SYSAFF=SC53
//JOBLIB   DD DSN=DB8F8.SDSNLOAD,DISP=SHR
//*--------------------------------------------------------------*
//*              BACKUP SYSTEM                                        -*
//*--------------------------------------------------------------*
//STEP001   EXEC DSNUPROC,UID='BKUP',SYSTEM='D8FG',
//               UTPROC=''
//*           UTPROC='RESTART'
//*           UTPROC='RESTART(PHASE)'
//STEPLIB   DD  DSN=DB8F8.SDSNLOAD,DISP=SHR
//              DD  DSN=DB8F8.SDSNEXIT,DISP=SHR
//SYSOUT    DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSIN     DD  *
  BACKUP SYSTEM FULL
/*
```

Example 2-26 BACKUP SYSTEM utility job output

```
1                   J E S 2  J O B  L O G  --  S Y S T E M  S C 5 3  --  N O D E
W T S C P L X 1
0
 16.57.49 JOB25401 ---- WEDNESDAY, 13 SEP 2006 ----
 16.57.49 JOB25401  IRR010I  USERID RC53     IS ASSIGNED TO THIS JOB.
 16.57.49 JOB25401  ICH70001I RC53     LAST ACCESS AT 16:14:55 ON WEDNESDAY,
SEPTEMBER 13, 2006
 16.57.49 JOB25401  $HASP373 DB8FBKUP STARTED - INIT A    - CLASS A - SYS SC53
 16.57.49 JOB25401  IEF403I DB8FBKUP - STARTED - TIME=16.57.49 - ASID=0024 - SC53
 16.57.55 JOB25401  -                                 --TIMINGS (MINS.)--
----PAGING COUNTS---
 16.57.55 JOB25401  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK
SERV  PG   PAGE   SWAP    VIO SWAPS
 16.57.55 JOB25401  -DB8FBKUP STEP001  DSNUPROC    00    491    .00    .00     .1
9620   0      0      0      0     0
 16.57.55 JOB25401  IEF404I DB8FBKUP - ENDED - TIME=16.57.55 - ASID=0024 - SC53
 16.57.55 JOB25401  -DB8FBKUP ENDED.  NAME-BACKUP SYSTEM        TOTAL CPU TIME=
.00  TOTAL ELAPSED TIME=    .1
 16.57.55 JOB25401  $HASP395 DB8FBKUP ENDED
...........
...........
00.00SEC VIRT  1808K SYS   280K EXT      20K SYS   11484K
 IGD104I DB8F8.SDSNLOAD                               RETAINED,  DDNAME=JOBLIB
```

```
 IEF375I  JOB/DB8FBKUP/START 2006256.1657
 IEF376I  JOB/DB8FBKUP/STOP  2006256.1657 CPU    0MIN 00.02SEC SRB    0MIN
00.00SEC
1DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = BKUP
 DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
0DSNU050I    DSNUGUTC -  BACKUP SYSTEM FULL
 DSNU1600I   DSNUVBBD - BACKUP SYSTEM UTILITY FOR DATA STARTING,
                        COPYPOOL = DSN$DB8F$DB
                        TOKEN = X'C4F8C6F1BF66C54143748F04BF668C793D8C'.
 DSNU1614I   DSNUVBBD - BACKUP SYSTEM UTILITY FOR DATA COMPLETED SUCCESSFULLY,
                        COPYPOOL = DSN$DB8F$DB
                        TOKEN = X'C4F8C6F1BF66C54143748F04BF668C793D8C'
                        ELAPSED TIME = 00:00:01.
 DSNU1600I   DSNUVBBD - BACKUP SYSTEM UTILITY FOR LOGS STARTING,
                        COPYPOOL = DSN$DB8F$LG
                        TOKEN = X'C4F8C6F1BF66C54143748F04BF668C793D8C'.
 DSNU1614I   DSNUVBBD - BACKUP SYSTEM UTILITY FOR LOGS COMPLETED SUCCESSFULLY,
                        COPYPOOL = DSN$DB8F$LG
                        TOKEN = X'C4F8C6F1BF66C54143748F04BF668C793D8C'
                        ELAPSED TIME = 00:00:00.
 DSNU1602I   DSNUVBBD - BACKUP SYSTEM UTILITY COMPLETED, ELAPSED TIME = 00:00:04.
 DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

2. The restore subsystem JCL generation is similar to the process described in 2.6.1, "Recovering a non-data sharing subsystem" on page 201 and is not repeated here. However, it does *not* generate a SYSPITR CRCR job for each member; there is only a single job created as JOB1. You must create a copy of this job for each member of the data sharing group and modify the JCL (such as BSDS names) accordingly to run on the appropriate member. Similar to the case of the non-data sharing environment, you have to choose the timestamp to recover. Export the RESTORE SYSTEM JCL (JOB1 and JOB2) to a partitioned data set (PDS) in z/OS. Ensure that there is a JOB1 for each and every member of the data sharing group. JOB1 creates a conditional restart control record (CRCR) record using DSNJU003 with the V8 SYSPITR keyword. The CRCR record is an LRSN point-in-time to which the system is to be recovered.
3. Stop all the members (D8F1 and D8F2) of the data sharing group.
4. Run the appropriately modified JOB1 on each member of the data sharing group, as shown in Example 2-27 and Example 2-28 on page 218.

Example 2-27 JOB1 output on member D8F1

```
1                   J E S 2  J O B  L O G  --  S Y S T E M  S C 5 3  --  N O D E
W T S C P L X 1
0
 17.21.08 JOB25432 ---- WEDNESDAY, 13 SEP 2006 ----
 17.21.08 JOB25432  IRR010I  USERID HENDRIK  IS ASSIGNED TO THIS JOB.
 17.21.08 JOB25432  ICH70001I HENDRIK  LAST ACCESS AT 17:10:37 ON WEDNESDAY,
SEPTEMBER 13, 2006
...................
....................
* IEFC653I SUBSTITUTION JCL - ,'DB2 RECOVERY
EXPERT',MSGCLASS=H,REGION=0M,NOTIFY=HENDRIK
         2 //D8F1CRCR EXEC PGM=DSNJU003
           //*
         3 //STEPLIB  DD DISP=SHR,DSN=DB8F8.SDSNEXIT
         4 //         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
```

```
        5 //SYSPRINT DD SYSOUT=*
        6 //SYSUT1   DD DISP=SHR,DSN=DB8FU.D8F1.BSDS01
        7 //SYSUT2   DD DISP=SHR,DSN=DB8FU.D8F1.BSDS02
        8 //SYSIN    DD *
..............
..............
0  CRESTART CREATE,SYSPITR=BF668C793D8C,FORWARD=YES,BACKOUT=YES
 DSNJ408I  DSNRJFCK CHECKPOINT RBA FOUND, RBA = 000045201EFC, TIME = 16:14:07
SEPTEMBER 13, 2006
 DSNJ411I  DSNRJRCR CRESTART CREATE FOR CRCRID = 0001, DDNAME = SYSUT1
 DSNJ408I  DSNRJFCK CHECKPOINT RBA FOUND, RBA = 000045201EFC, TIME = 16:14:07
SEPTEMBER 13, 2006
 DSNJ411I  DSNRJRCR CRESTART CREATE FOR CRCRID = 0001, DDNAME = SYSUT2
 DSNJ225I  CRESTART OPERATION COMPLETED SUCCESSFULLY
-DSNJ200I  DSNJU003 CHANGE LOG INVENTORY UTILITY PROCESSING COMPLETED SUCCESSFULLY
```

Example 2-28   JOB1 output on member D8F21

```
1                  J E S 2  J O B  L O G  --  S Y S T E M  S C 5 3  --  N O D E
W T S C P L X 1
0
 18.19.24 JOB25484 ---- WEDNESDAY, 13 SEP 2006 ----
 18.19.24 JOB25484  IRR010I  USERID HENDRIK  IS ASSIGNED TO THIS JOB.
 18.19.25 JOB25484  ICH70001I HENDRIK  LAST ACCESS AT 17:21:08 ON WEDNESDAY,
SEPTEMBER 13, 2006
...................
...................*
           IEFC653I SUBSTITUTION JCL - ,'DB2 RECOVERY
EXPERT',MSGCLASS=H,REGION=0M,NOTIFY=HENDRIK
        2 //D8F1CRCR EXEC PGM=DSNJU003
          //*
        3 //STEPLIB  DD DISP=SHR,DSN=DB8F8.SDSNEXIT
        4 //         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
        5 //SYSPRINT DD SYSOUT=*
        6 //SYSUT1   DD DISP=SHR,DSN=DB8FU.D8F2.BSDS01
        7 //SYSUT2   DD DISP=SHR,DSN=DB8FU.D8F2.BSDS02
        8 //SYSIN    DD *
 ICH70001I HENDRIK  LAST ACCESS AT 17:21:08 ON WEDNESDAY, SEPTEMBER 13, 2006
............
............
0  CRESTART CREATE,SYSPITR=BF668C793D8C,FORWARD=YES,BACKOUT=YES
 DSNJ408I  DSNRJFCK CHECKPOINT RBA FOUND, RBA = 00002B9DF5CC, TIME = 16:14:17
SEPTEMBER 13, 2006
 DSNJ411I  DSNRJRCR CRESTART CREATE FOR CRCRID = 0001, DDNAME = SYSUT1
 DSNJ408I  DSNRJFCK CHECKPOINT RBA FOUND, RBA = 00002B9DF5CC, TIME = 16:14:17
SEPTEMBER 13, 2006
 DSNJ411I  DSNRJRCR CRESTART CREATE FOR CRCRID = 0001, DDNAME = SYSUT2
 DSNJ225I  CRESTART OPERATION COMPLETED SUCCESSFULLY
-DSNJ200I  DSNJU003 CHANGE LOG INVENTORY UTILITY PROCESSING COMPLETED SUCCESSFULLY
```

5. Delete all the coupling facility (CF) structures.

   The deletion of the LOCK structure and verification that it was deleted successfully is shown in Example 2-29, and the deletion and verification for the shared communications area (SCA) structure is shown in Example 2-30.

   Note: You must also delete structures for any failed group buffer pools (GBPs).

Example 2-29 LOCK structure deletion and verification that deletion is successful

```
   "SETXCF FORCE,STRUCTURE,STRNAME=DB8FU_LOCK1
IXC579I NORMAL DEALLOCATION FOR STRUCTURE DB8FU_LOCK1 IN 709
        COUPLING FACILITY  002084.IBM.02.000000026A3A
                           PARTITION: 1E    CPCID: 00
HAS BEEN COMPLETED.
PHYSICAL STRUCTURE VERSION: BF24309C F6D093C3
INFO116: 132C2080 01 2800 00000012
TRACE THREAD: 0002E1E6.
IXC353I THE SETXCF FORCE REQUEST FOR STRUCTURE 710
DB8FU_LOCK1 WAS COMPLETED:
STRUCTURE DELETED BUT ALSO RESULTED IN DELETED CONNECTION(S)
.......
.......
   "D XCF,STR,STRNAME=DB8FU_LOCK1
IXC360I  17.29.38  DISPLAY XCF 72
STRNAME: DB8FU_LOCK1
 STATUS: NOT ALLOCATED
 POLICY INFORMATION:
  POLICY SIZE    : 16384 K
  POLICY INITSIZE: 8192 K
  POLICY MINSIZE : 0 K
  FULLTHRESHOLD  : 80
  ALLOWAUTOALT   : NO
  REBUILD PERCENT: 5
  DUPLEX         : DISABLED
  ALLOWREALLOCATE: YES
  PREFERENCE LIST: CF03     CF05
  ENFORCEORDER   : NO
  EXCLUSION LIST IS EMPTY
```

Example 2-30 SCA structure deletion and verification that deletion is successful

```
   "SETXCF FORCE,STRUCTURE,STRNAME=DB8FU_SCA
IXC579I NORMAL DEALLOCATION FOR STRUCTURE DB8FU_SCA IN 720
        COUPLING FACILITY  002084.IBM.02.000000026A3A
                           PARTITION: 1E    CPCID: 00
HAS BEEN COMPLETED.
PHYSICAL STRUCTURE VERSION: BF243098 7A1FB241
INFO116: 132C2080 01 2800 0000000F
TRACE THREAD: 0002E282.
IXC353I THE SETXCF FORCE REQUEST FOR STRUCTURE 721
DB8FU_SCA WAS COMPLETED:
STRUCTURE WAS DELETED
........................
........................
   "D XCF,STR,STRNAME=DB8FU_SCA
```

```
IXC360I  17.30.58  DISPLAY XCF 745
STRNAME: DB8FU_SCA
 STATUS: NOT ALLOCATED
 POLICY INFORMATION:
  POLICY SIZE    : 49152 K
  POLICY INITSIZE: 8192 K
  POLICY MINSIZE : 0 K
  FULLTHRESHOLD  : 80
  ALLOWAUTOALT   : NO
  REBUILD PERCENT: 5
  DUPLEX         : DISABLED
  ALLOWREALLOCATE: YES
  PREFERENCE LIST: CF03     CF05
  ENFORCEORDER   : NO
  EXCLUSION LIST IS EMPTY
```

6. Start all the DB2 members in the data sharing group. You must have SYSADM authority to start DB2 now. The CF structures are rebuilt. Reply `Y` to the following prompt:

   ```
   *034 DSNJ245I  -D8F1 CONDITIONAL RESTART RECORD INDICATES TRUNCATION AT LRSN
   BF668C793D8C.  REPLY Y TO CONTINUE, N TO CANCEL
   ```

7. Verify that the integrated catalog facility (ICF) catalog volumes for DB2 are not active by issuing modify commands such as:

   ```
   F CATALOG,CLOSE(UCAT.DB8FL)
   F CATALOG,CLOSE(UCAT.TOTDDJ)
   F CATALOG,UNALLOCATE(UCAT.DB8FL)
   .............
   ```

   This is critical for the RESTORE SYSTEM job (JOB2 as shown in Example 2-31) to complete.

8. The system is in System Recovery Pending mode. Therefore, you require install SYSADM authority to run the JOB2 JCL that is generated by Recovery Expert for z/OS. Run it on *only* one of the members of the data sharing group. JOB2 runs the RESTORE SYSTEM utility. The output of this job is shown in Example 2-32 on page 221.

Example 2-31   JOB2 JCL

```
//F8SYSRE2 JOB ,'DB2 RECOVERY EXPERT',MSGCLASS=H,
//            REGION=0M,NOTIFY=&SYSUID
/*JOBPARM S=SC53
//*
//*********************************************************************
//*
//*  IBM DB2 Recovery Expert for z/OS V1R1
//*
//*  SSID: D8F1
//*  User: HENDRIK
//*  Date: 2006-09-13       Time: 17:12
//*
//*********************************************************************
//*
//*
//** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*                                                                    *
//*  Step:     D8F1RSSY                                                *
//*                                                                    *
```

```
//*  Desc:     This step will invoke the IBM Restore System Stand      *
//*            alone utility                                           *
//*                                                                    *
//** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*                                                                    *
//D8F1RSSY EXEC  PGM=DSNUTILB,COND=(4,LT),
//          PARM=(D8F1,RSSY)
//*
//STEPLIB  DD DISP=SHR,DSN=DB8F8.SDSNEXIT
//         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSIN    DD *
   RESTORE SYSTEM
/*
//*
```

Example 2-32   JOB2 output

```
1                   J E S 2  J O B  L O G  --  S Y S T E M  S C 5 3  --  N O D E
W T S C P L X 1
0
 12.08.07 JOB26666 ---- FRIDAY,    15 SEP 2006 ----
 12.08.07 JOB26666  IRR010I  USERID RC53     IS ASSIGNED TO THIS JOB.
 12.08.08 JOB26666  ICH70001I RC53     LAST ACCESS AT 12:07:23 ON FRIDAY,
SEPTEMBER 15, 2006
...............
...............
           IEFC653I SUBSTITUTION JCL - ,'DB2 RECOVERY
EXPERT',MSGCLASS=H,REGION=0M,NOTIFY=RC53
         2 //D8F1RSSY EXEC  PGM=DSNUTILB,COND=(4,LT),
           //           PARM=(D8F1,RSSY)
           //*
         3 //STEPLIB  DD DISP=SHR,DSN=DB8F8.SDSNEXIT
         4 //         DD DISP=SHR,DSN=DB8F8.SDSNLOAD
         5 //SYSPRINT DD SYSOUT=*
         6 //SYSOUT   DD SYSOUT=*
         7 //UTPRINT  DD SYSOUT=*
         8 //SYSIN    DD *
           //*
...........
...........
1DSNU000I    DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = RSSY
 DSNU1044I   DSNUGTIS - PROCESSING SYSIN AS EBCDIC
0DSNU050I    DSNUGUTC -  RESTORE SYSTEM
 DSNU1606I   DSNUVBRD - RESTORE SYSTEM UTILITY STARTING,
                        COPYPOOL = DSN$DB8F$DB
                        TOKEN = X'C4F8C6F1BF66C54143748F04BF668C793D8C'.
 DSNU1627I   DSNUVBRD - RESTORE SYSTEM PRE-LOG APPLY COMPLETED SUCCESSFULLY,
                        COPYPOOL = DSN$DB8F$DB
                        TOKEN = X'C4F8C6F1BF66C54143748F04BF668C793D8C'
                        ELAPSED TIME = 00:00:00.
 DSNU1604I -D8F1 DSNUVARL - RESTORE SYSTEM PHASE LOG APPLY STARTED AT LOG POINT =
X'BF668C793D8C'.
```

```
 DSNU1635I -D8F1 DSNUVARL - THE RBA RANGE FOR THE LAST CHECKPOINT ISSUED DURING
THE LOGAPPLY PHASE OF THE RESTORE SYSTEM
 UTILITY IS START_RBA = X'BF69083B3560' END_RBA = X'BF69083B5198' FOR MEMBER D8F1
 DSNU1628I   DSNUVBRD - RESTORE SYSTEM PHASE LOG APPLY COMPLETED, ELAPSED TIME =
00:00:00.
 DSNU010I    DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
```

The hierarchical storage management (HSM) address space has similar information when JOB2 completes successfully, as shown in Example 2-33.

Example 2-33 HSM address space messages on JOB2 completion

```
.............
ARC1801I FAST REPLICATION RECOVERY IS STARTING FOR  636
ARC1801I (CONT.) COPY POOL DSN$DB8F$DB, AT 12:08:08 ON 2006/09/15
ARC1805I THE FOLLOWING 00003 VOLUME(S) WERE  637
ARC1805I (CONT.) SUCCESSFULLY PROCESSED BY FAST REPLICATION RECOVERY
ARC1805I (CONT.) OF COPY POOL DSN$DB8F$DB
ARC1805I (CONT.) DB8FS1
ARC1805I (CONT.) TOTDDJ
ARC1805I (CONT.) TOTDDK
ARC1802I FAST REPLICATION RECOVERY HAS COMPLETED FOR  641
ARC1802I (CONT.) COPY POOL DSN$DB8F$DB, AT 12:08:08 ON 2006/09/15,
ARC1802I (CONT.) FUNCTION RC=0000, MAXIMUM VOLUME RC=0000
...................
```

9. Stop and restart DB2 on all the members of the data sharing group to resume usual processing.

A

Introduction to DB2 for z/OS backup and recovery

In this appendix, we provide an introduction to DB2 for z/OS backup and recovery.

A.1 Introduction

DB2 for z/OS provides the means to recover data to its current state or to an earlier state (point-in-time). The units of data that can be recovered are table spaces, indexes or index spaces, partitions, and data sets. It also supports the recovery of the entire DB2 subsystem to any point-in-time.

To avoid loss of critical data and minimize delays in recovering data, you must develop a comprehensive backup and recovery strategy that includes well-tested procedures to perform the following functions:

- Create a point-of-consistency for the system when there are no outstanding units-of-work and there are no objects in a restricted state
- Restore the system to a point-of-consistency
- Create a point-of-consistency for a set of related data objects
- Restore the set of data objects to a point-of-consistency
- Back up the DB2 catalog and directory
- Recover the DB2 catalog and directory
- Back up your data
- Recover your data to current
- Recover from out-of-space conditions
- Recover from a hardware or power failure
- Recover from a z/OS component failure

In addition, you must also develop a procedure for off-site recovery in case of a disaster. The principal tools for backup and recovery are the QUIESCE, REPORT, BACKUP SYSTEM, COPY, RESTORE SYSTEM, RECOVER, MERGECOPY, DSN1COPY, and REBUILD INDEX utilities.

In the following sections, we briefly discuss:

- Overview of backup and recovery utilities
- Creating a point-of-consistency
- DB2 catalog and directory recovery
- Recovery using stand-alone backup utilities

For further information about backup and recovery, see *DB2 UDB for z/OS V8 Administration Guide*, SC18-7413, and *DB2 UDB for z/OS V8 Utility Guide and Reference*, SC18-7427.

A.2 Overview of backup and recovery utilities

In this section, we briefly describe the BACKUP SYSTEM, COPY, DSN1COPY, MERGECOPY, QUIESCE, REPORT, RESTORE SYSTEM, and RECOVER utilities.

A.2.1 BACKUP SYSTEM

The online BACKUP SYSTEM utility invokes z/OS DFSMShsm (Version 1 Release 5 or later) to copy the volumes on which the DB2 data and log information resides for either a DB2 subsystem or data sharing group. You can use BACKUP SYSTEM to copy all data for a single application, for example, when DB2 is the database server for a resource planning solution. All data sets that you want to copy must be Storage Management Subsystem (SMS) managed data sets. Subsequently, you can run the RESTORE SYSTEM utility to recover the data.

In a data sharing environment, if any failed or abnormally quiesced members exist, the BACKUP SYSTEM request fails. The BACKUP SYSTEM utility uses copy pools[1], which are new constructions in z/OS DFSMShsm V1R5. Each DB2 subsystem can have up to two copy pools, one for databases and one for logs.

BACKUP SYSTEM copies the volumes that are associated with these copy pools at the time of the copy. The output for BACKUP SYSTEM is the copy of the volumes on which the DB2 data and log information resides. The BACKUP SYSTEM history is recorded in the bootstrap data sets (BSDSs).

A.2.2 COPY

The online COPY utility creates up to four image copies of any of the following objects:

- Table space
- Table space partition
- Data set of a linear table space or all data sets of a multi-piece non-partitioned object
- Index space
- Index space partition

The two types of image copies are:

- A full image copy, which is a copy of all pages in a table space, partition, data set, or index space
- An incremental image copy, which is a copy of only those data pages that have been modified since the last use of the COPY utility and optionally system pages

Information about the image copies are recorded in the SYSIBM.SYSCOPY catalog table.

> **Note:** The CONCURRENT option of COPY specifies that the utility must use IBM DFSMSdss™ concurrent copy to make the full image copy; incremental image copy is not supported with this option. The image copy is recorded in the SYSIBM.SYSCOPY catalog table with ICTYPE=F and STYPE=C or STYPE=J.

The RECOVER utility uses these copies when recovering an object to current or a point-in-time. Copies can also be used by the MERGECOPY, RECOVER, COPYTOCOPY, and UNLOAD utilities.

You can copy a list of objects in parallel to improve performance. Specifying a list of objects along with the SHRLEVEL REFERENCE option creates a single recovery point for that list of objects. When you specify the PARALLEL keyword, this enables you to copy a list of objects in parallel, rather than serially.

[1] A copy pool is a defined set of storage groups, which contains data that DFSMShsm can back up and recover collectively. For more information about copy pools, see *z/OS V1R5.0 DFSMSdfp Storage Administration Reference*, SC26-7402-02.

A.2.3 DSN1COPY

DSN1COPY is a stand-alone utility that you can use to copy:

- DB2 Virtual Storage Access Method (VSAM) data sets to sequential data sets
- DSN1COPY sequential data sets to DB2 VSAM data sets
- DB2 VSAM data sets to other DB2 VSAM data sets
- DSN1COPY sequential data sets to other sequential data sets

Because DSN1COPY is a stand-alone utility, the copies taken by it are *not* recorded in the SYSIBM.SYSCOPY catalog table.

> **Note:** A DB2 VSAM data set is a single piece of a non-partitioned table space or index, or a single partition of a partitioned table space or index. The input must be a single z/OS sequential or VSAM data set. Concatenation of input data sets is not supported.

Using DSN1COPY, you can also print hexadecimal dumps of DB2 data sets and databases, check the validity of data or index pages (including dictionary pages for compressed data), translate database object identifiers (OBIDs) to enable moving data sets between different systems, and reset to 0 the log relative byte address (RBA) that is recorded in each index page or data page. You cannot run DSN1COPY on concurrent copies. You can use the DSN1COPY utility on large object (LOB) table spaces by specifying the LOB keyword and omitting the SEGMENT and INLCOPY keywords.

A.2.4 MERGECOPY

The online MERGECOPY utility merges image copies that the COPY utility produces, image copies that the COPYTOCOPY utility produces, or inline copies that the LOAD or REORG utilities produce. It can merge several incremental copies to make one incremental copy. It can also merge incremental copies with a full image copy to make a new full image copy. You cannot run MERGECOPY on concurrent copies.

MERGECOPY operates on the image copy data sets of a table space or index, and not on the table space or index itself. The output from the MERGECOPY utility consists of one of the following types of copies:

- A new single incremental image copy
- A new full image copy

You can create the new image copy for the local or recovery site.

A.2.5 QUIESCE

The online QUIESCE utility establishes a quiesce point for a table space, partition, table space set, or list of table spaces and table space sets. A quiesce point is the current log RBA or log record sequence number (LRSN). QUIESCE then records the quiesce point in the SYSIBM.SYSCOPY catalog table. A successful QUIESCE improves the probability of a successful RECOVER. You must run QUIESCE frequently between regular executions of COPY to establish regular recovery points for future possible point-in-time recovery.

With the WRITE(YES) option, QUIESCE writes changed pages for the table spaces and their indexes from the DB2 buffer pool to disk. The catalog table SYSCOPY records the current RBA and the timestamp of the quiesce point. A row with ICTYPE='Q' is inserted into SYSIBM.SYSCOPY for each table space that is quiesced. DB2 also inserts a SYSCOPY row with ICTYPE='Q' for any indexes (defined with the COPY YES attribute) over a table space that is being quiesced. (Table spaces DSNDB06.SYSCOPY, DSNDB01.DBD01, and DSNDB01.SYSUTILX are exceptions; their information is written to the log.)

A.2.6 REBUILD INDEX

The REBUILD INDEX utility reconstructs indexes or index partitions from the table that they reference. Because the data that DB2 has to build an index is in the table space on which the index is based, you are not required to create image copies of indexes. To rebuild the index, you do not have to recover the table space, unless it is also damaged. You do not have to rebuild an index merely because you have recovered the table space on which it is based if the table space is recovered to a point that is consistent with its indexes. If you recover a table space to a prior point-in-time and do not recover all the indexes to the same point-in-time, you must rebuild or recover all of the indexes.

Some logging might occur if both of the following conditions are true:

- The index is a non-partitioning index
- The index is being concurrently accessed either by Structured Query Language (SQL) on a different partition of the same table space or by a utility that is run on a different partition of the same table space

A.2.7 REPORT

The online REPORT utility provides information about table spaces. Use REPORT TABLESPACESET to find the names of all the table spaces and tables in a referential structure, including LOB table spaces. The REPORT utility also provides the LOB table spaces that are associated with a base table space. Use REPORT RECOVERY to find information that is necessary for recovering a table space, index, or a table space and all of its indexes.

The output from REPORT TABLESPACESET consists of the names of all table spaces in the table space set that you specify. It also lists all tables in the table spaces and all tables that are dependent on those tables. The output from REPORT RECOVERY consists of the recovery history from the SYSIBM.SYSCOPY catalog table, log ranges from the SYSIBM.SYSLGRNX directory table, and volume serial numbers where archive log data sets from the BSDS reside. In addition, REPORT RECOVERY output includes information about any indexes on the table space that are in the informational COPY-pending status because this information affects the recoverability of an index.

In a data sharing environment, the REPORT output provides:

- The RBA of when DB2 was migrated to Version 8
- The high and low RBA values of the migrated member
- A list of any SYSLGRNX records from the time before data sharing was enabled that cannot be used to recover to any point-in-time after data sharing was enabled
- For SYSCOPY, the member from which the image copy was deleted

A.2.8 RESTORE SYSTEM

The online RESTORE SYSTEM utility invokes z/OS DFSMShsm (Version 1 Release 5 or later) to recover a DB2 subsystem or a data sharing group to a previous point-in-time. To perform the recovery, the utility uses data that is copied by the BACKUP SYSTEM utility. All data sets that you want to recover must be SMS-managed data sets.

You can run the RESTORE SYSTEM utility from any member in a data sharing group, even one that is usually quiesced when any backups are taken. Any member in the data sharing group that is active at or beyond the log truncation point must be restarted, and its logs are truncated to the SYSPITR LRSN point. You can specify the SYSPITR LRSN point in the CRESTART control statement of the DSNJU003 (change log inventory) utility. Any data sharing group member that is usually quiesced at the time the backups are taken and is not active at or beyond the log truncation point does not have to be restarted.

Restrictions: DFSMShsm V1R5 can maintain multiple backup versions of copy pools. However, you cannot specify a particular backup version to be used by the RESTORE SYSTEM utility. RESTORE SYSTEM uses the latest version before the log truncation point. You can specify the log truncation point with the CRESTART SYSPITR option of the DSNJU003 (change log inventory) stand-alone utility.

RESTORE SYSTEM does not restore logs; the utility only applies the logs. If you specify BACKUP SYSTEM FULL to create copies of both the data and the logs, you can restore the logs using DFSMShsm. The output of RESTORE SYSTEM is the recovered copy of the data volume or volumes.

A.2.9 RECOVER

The online RECOVER utility recovers data to the current state or to a previous point-in-time by possibly restoring a copy and then applying log records. The largest unit of data recovery is the table space or index space; the smallest is the page. You can recover a single object or a list of objects. The RECOVER utility recovers an entire table space, index space, a partition or data set, pages within an error range, or a single page. You recover data from image copies of an object, from log records that contain changes to the object, or from both. If the most recent full image copy data set is unusable, and previous image copy data sets exist in the system, RECOVER uses the previous image copy data sets.

The output from RECOVER consists of recovered data (a table space, index, partition or data set, error range, or page within a table space). If you specify the TOLOGPOINT, TORBA, TOCOPY, TOLASTCOPY, or TOLASTFULLCOPY option to recover data to a point-in-time, RECOVER puts any associated index spaces in REBUILD-pending status. You must run REBUILD INDEX to remove the REBUILD-pending status from the index space or recover the indexes.

If you use the RECOVER utility to recover a referentially related table space set or a base table space and LOB table space set, you must ensure that you recover the entire set of table spaces. This task includes rebuilding or recovering all indexes (including indexes on auxiliary tables for a base table space and LOB table spaces) to a consistent point. If you do not include every member of the set, or if you do not recover the entire set to the same point-in-time, RECOVER sets the CHECK-pending status on for all dependent table spaces, base table spaces, or LOB table spaces in the set.

Tip: If you use the RECOVER utility to partially recover data and all indexes on the data, recover these objects to a consistent point (which can be a quiesce point or a SHRLEVEL CHANGE copy). Otherwise, RECOVER places all indexes in the CHECK-pending status.

If you restore a page set using stand-alone utilities such as DSN1COPY or DFSMSdss concurrent copy, you can use RECOVER to only apply the changes in the DB2 log by using the LOGONLY option. This option specifies that the target objects are to be recovered from their existing data sets by applying only log records to the data sets. DB2 applies all log records that are written after a point that is recorded in the data set itself.

A.3 Creating a point-of-consistency

A point-of-consistency is when there are no outstanding units-of-work and none of the objects in the identified set are in a restricted state. A point-of-consistency can be established at a system-wide level (for a DB2 subsystem), or for a set of table spaces and implicitly its associated indexes. The table spaces might or might not be related through referential integrity constraints. A point-of-consistency is typically established for a set of table spaces using the QUIESCE utility.

A.4 DB2 catalog and directory recovery

You must recover DB2 catalog and directory objects in a particular order. Because the recovery of some catalog and directory objects depends on information that is derived from other catalog and directory objects, you must recover these objects in separate RECOVER utility control statements.

However, you can use the same RECOVER control statement to recover a catalog or directory table space along with its corresponding IBM-defined indexes. After these logically dependent objects are restored to an undamaged state, you can recover the remaining catalog and directory objects in a single RECOVER utility control statement. These restrictions apply regardless of the type of recovery that you perform on the catalog. You can use the REPORT utility to report on recovery information about the catalog and directory.

Attention: Recovering the DB2 catalog and directory to a prior point-in-time is strongly discouraged.

After a DB2 conditional restart in which a log record range is specified, such as with a cold start, a portion of the DB2 recovery log is no longer available. If the unavailable portion includes information that is required for internal DB2 processing, an attempt to use the RECOVER utility to restore directory table spaces DSNDBD01 or SYSUTILX, or catalog table space SYSCOPY, will fail with abend 00E40119. Instead of using the RECOVER utility, use a different procedure using DSN1COPY to recover those table spaces and their indexes.

A.5 Recovery using stand-alone backup utilities

Many organizations use tools such as DFSMSdss to take volume backups of their total environment that includes both DB2 and non-DB2 data. Other organizations use tools such as DSN1COPY to back up the underlying VSAM data sets of individual DB2 page sets[2]. Speed and convenience are typically the main reasons for using such backup mechanisms.

Organizations require the recovery of DB2 objects to use these stand-alone utility backups. DB2 provides the LOGONLY option in the RECOVER utility to support backups taken with stand-alone backups. When the backup of the data sets is restored to disk using the stand-alone backups, you can run the RECOVER utility with the LOGONLY option to apply all the changes that occurred since the backup was taken of the underlying restored data set. The RECOVER utility knows the point in the DB2 log from which to apply the changes to the data because the restored data set has a record of the starting point for log apply.

Attention: When you use stand-alone backup and restore utilities with DB2 objects, take certain precautions to ensure that the backup images can be used correctly in point-in-time recovery situations.

[2] A page set is a term used to refer to either a table space or an index space, or both.

B

Recovery Expert for z/OS GUI client tutorial

In this appendix, we provide a brief tutorial of the Recovery Expert for z/OS graphical user interface (GUI) client.

B.1 Introduction

The Recovery Expert for z/OS GUI client is installed on a Microsoft Windows platform and is the primary vehicle for performing the various recovery actions on objects supported by the Recovery Expert for z/OS tool. This includes recovery to current, timestamps, image copies, relative byte address (RBA) or log record sequence numbers (LRSNs), and quiet times (with the help of log analysis) for existing objects. It also enables you to recover dropped objects. In case you might have to interrupt your Recovery Expert for z/OS GUI client activity to attend to more pressing matters, Recovery Expert for z/OS GUI client provides an ability to save specifications to continue at a later time. You can also share specifications with your colleagues, if required.

Note: The objective of this appendix is to provide a flavor of this tool, and therefore it does not include all the window interactions or an explanation of the various options provided. We encourage you to use the context Help buttons and the General Help tab to understand the various elements described in any particular window.

This brief tutorial can provide the basis for extrapolating to all the functionalities available in the Recovery Expert for z/OS GUI client. The main topics covered in this appendix are:

- Connecting to Recovery Expert for z/OS server
- Setting Recovery Expert for z/OS GUI client options
- Recovery to current
- Recovery to RBA/LRSN
- Log analysis
- Specifications

B.2 Connecting to Recovery Expert for z/OS server

To launch the Recovery Expert for z/OS GUI client from the Microsoft Windows platform, click **Start** → **Programs** → **IBM DB2 Recovery Expert** → **IBM DB2 Recovery Expert**. This opens the prompt to connect to the Recovery Expert for z/OS server, as shown in Figure B-1.

Type the host name (or IP address) and port number if known, or click the ellipsis to obtain a list of available servers[1] (Figure B-2 on page 233) to select from. The prompt window reflects the last server that you connected to. The drop-down list provides the list of servers that you connected to previously, as shown in Figure B-3 on page 233. Click **Connect** to initiate the connection, as shown in Figure B-4 on page 233. This takes you to the Launchpad window shown in Figure B-5 on page 234.

Figure B-1 Prompt to connect to Recovery Expert for z/OS server

[1] To use this feature, enable automatic discovery, as described in “Automatic discovery” on page 24.

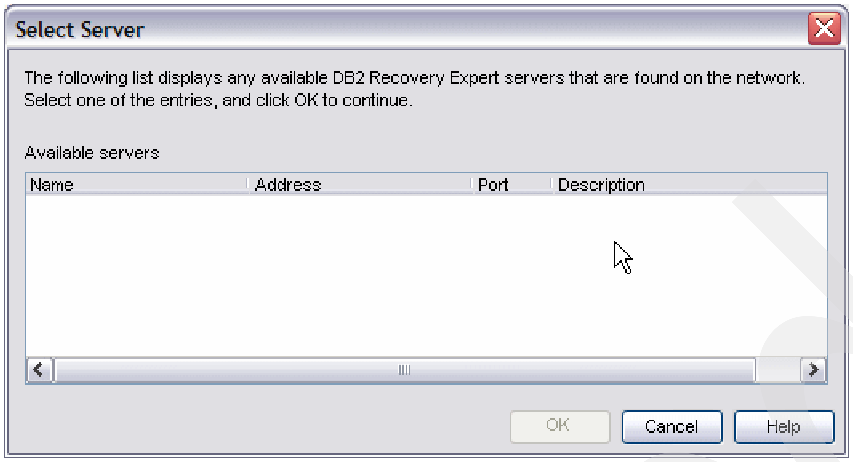

Figure B-2 List of available Recovery Expert for z/OS servers

Figure B-3 Typing or selecting Recovery Expert for z/OS server to connect to (1/2)

Figure B-4 Typing or selecting Recovery Expert for z/OS server to connect to (2/2)

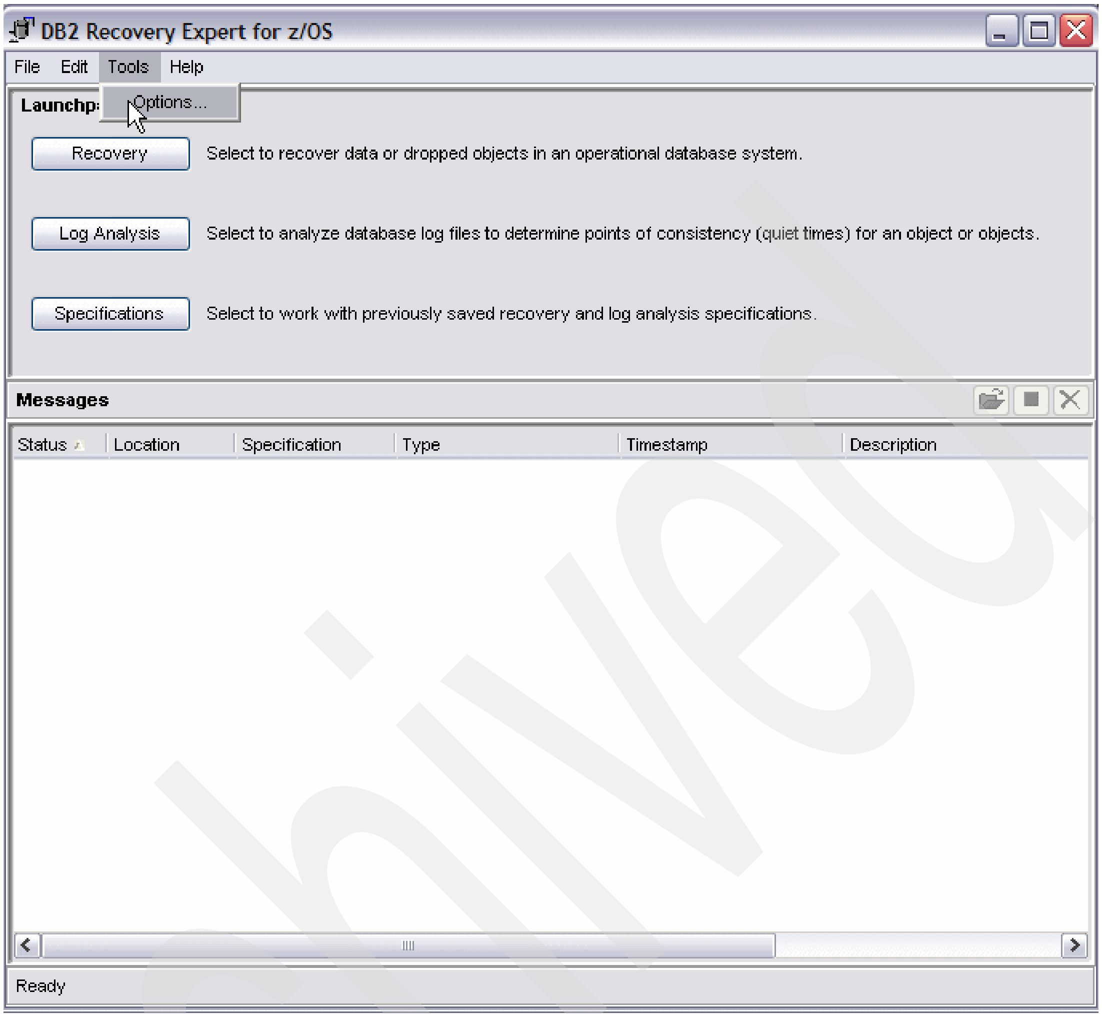

Figure B-5 Launchpad

B.3 Setting Recovery Expert for z/OS GUI client options

Figure B-5 shows the selection Tools → Options. This allows you to specify parameters for the execution of the Recovery Expert for z/OS GUI client session. A number of categories of options are available to set:

- General options (shown in Figure B-6 on page 235) mostly relate to limiting the presentation of certain windows during navigation. You can use the Reset Passwords button to reset all passwords that you previously had Recovery Expert for z/OS GUI client remember; when you click this button, as shown in Figure B-7 on page 235, you get the success message shown in Figure B-8 on page 236.
- Limits options (shown in Figure B-9 on page 236) enable you to limit the amount of data returned to the client.
- Job Cards options (shown in Figure B-10 on page 236) enable you to create the job card to be used when Recovery Expert for z/OS GUI client generates Multiple Virtual Storage (MVS) jobs for execution.
- Quiet Time Tables options (shown in Figure B-11 on page 237) enable you to specify the schema level repository (SLR) tables to which quiet times detected by log analysis must be written.

- Logging options (shown in Figure B-12 on page 237) enable you to specify the components that must produce log output and the destination of the log messages.
- Figure B-13 on page 238 and Figure B-14 on page 238 show examples of how to use context Help. Figure B-15 on page 239 and Figure B-16 on page 240 show examples of how to use General Help.

> **Important:** As mentioned previously, you are strongly advised to avail of these help facilities provided.

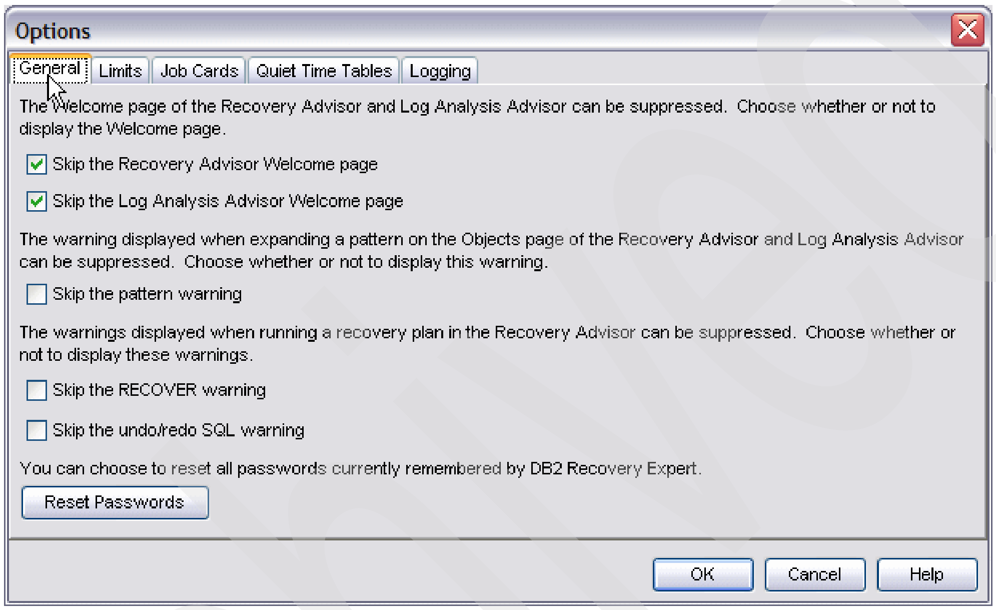

Figure B-6 General options

Figure B-7 Resetting passwords

Figure B-8 Passwords reset message

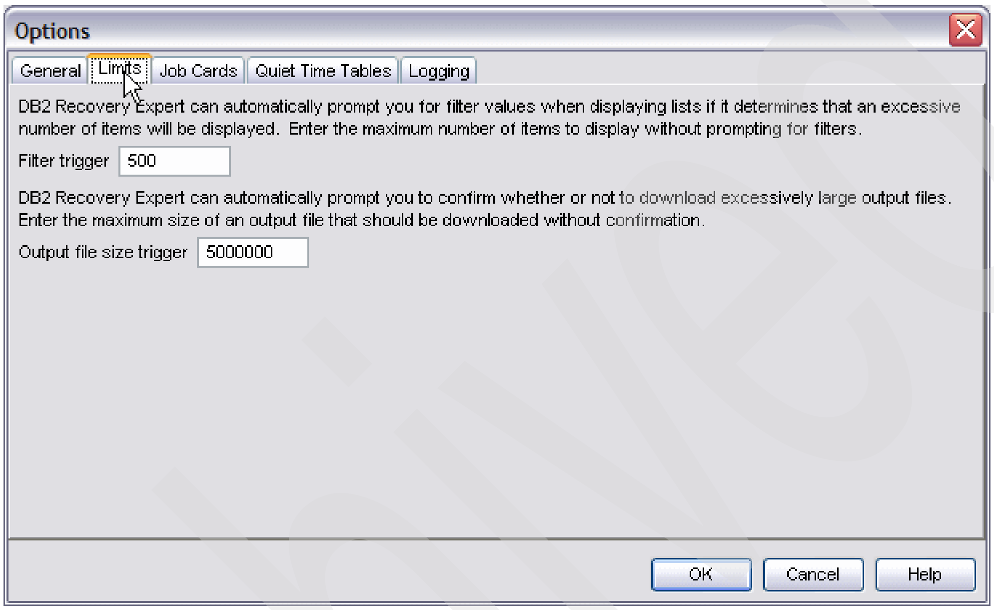

Figure B-9 Limits options

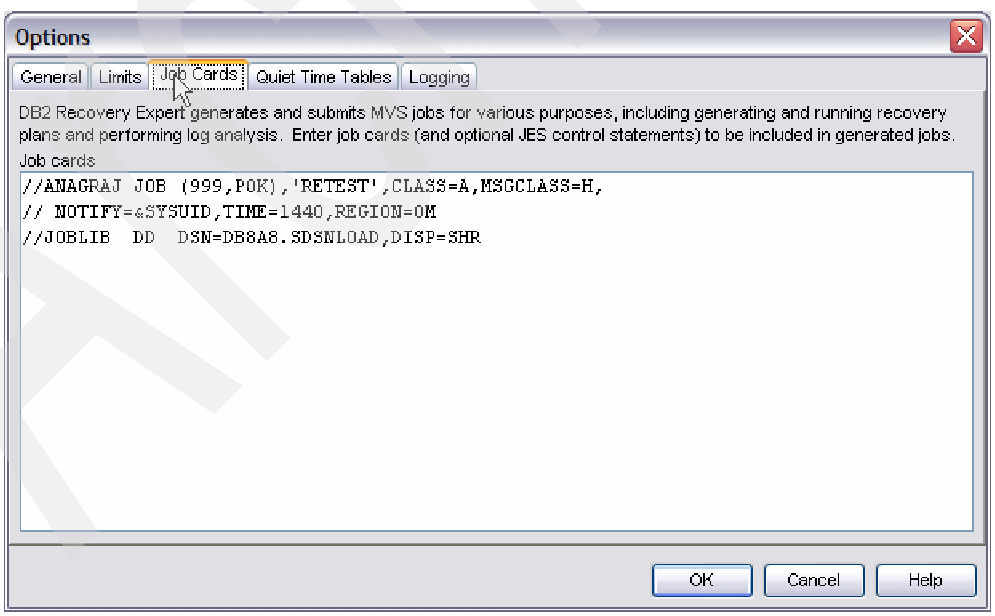

Figure B-10 Job cards options

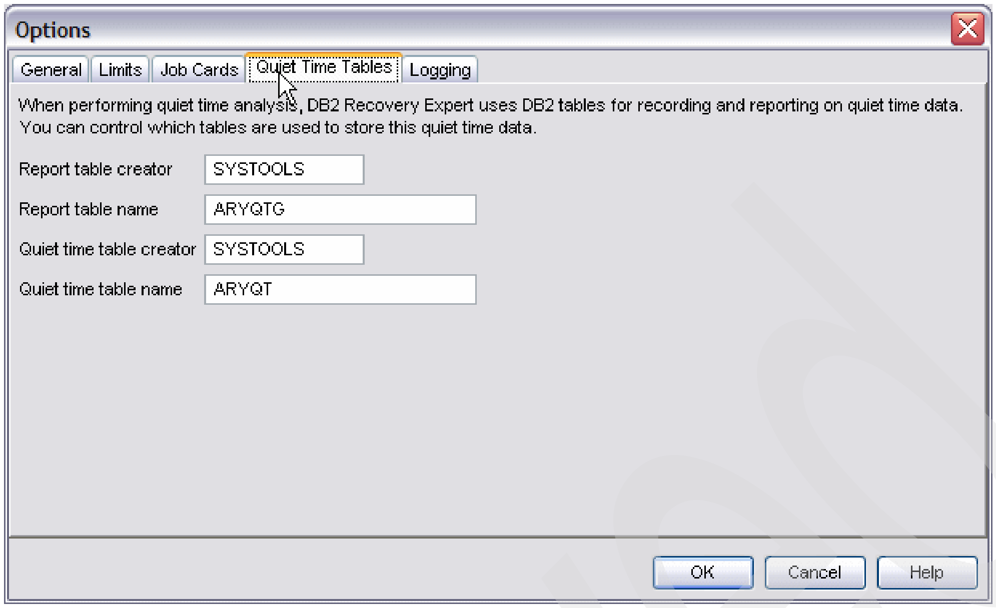

Figure B-11 Quiet time tables options

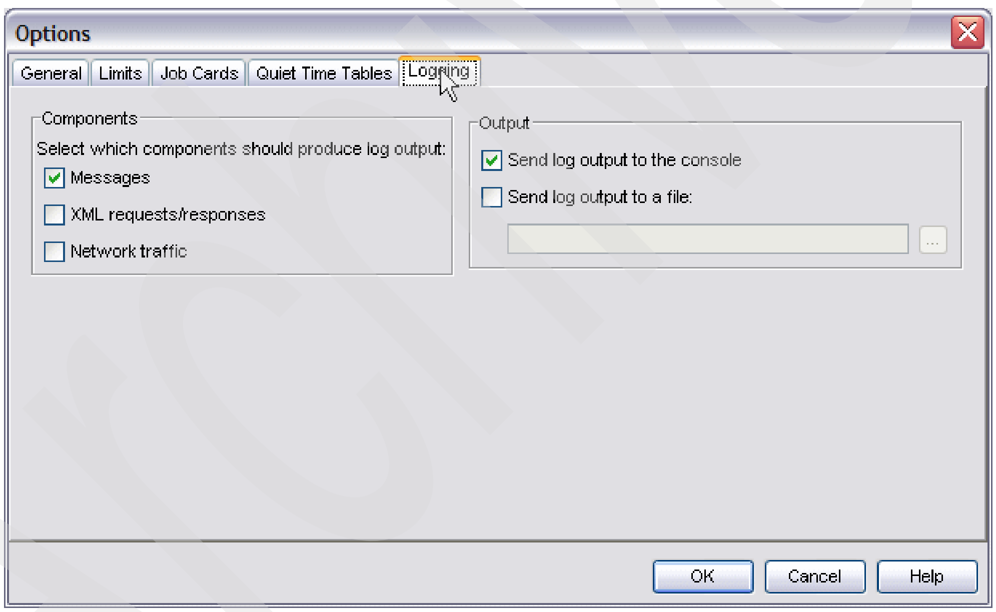

Figure B-12 Logging options

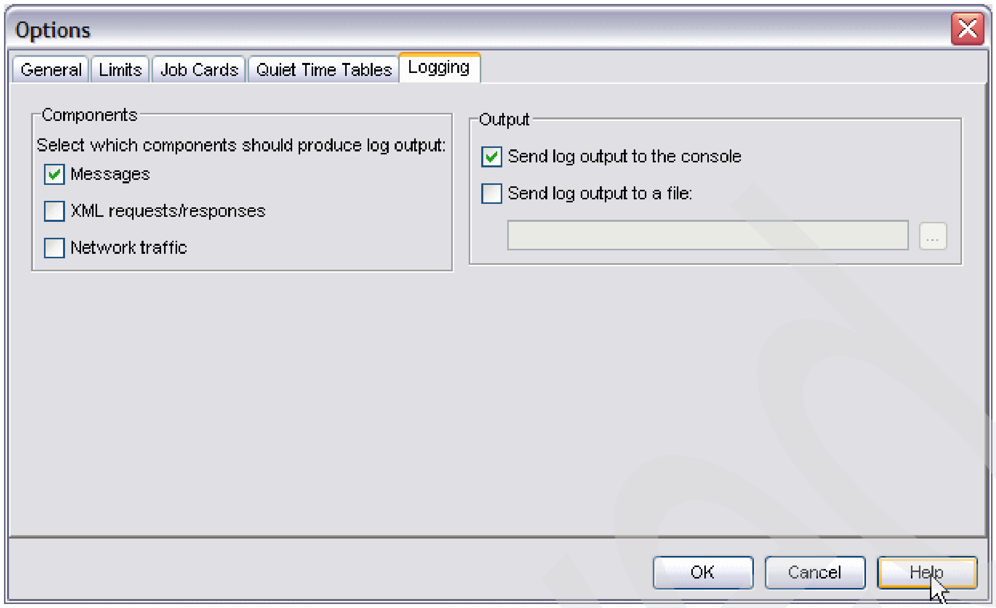

Figure B-13 Context help (1/2)

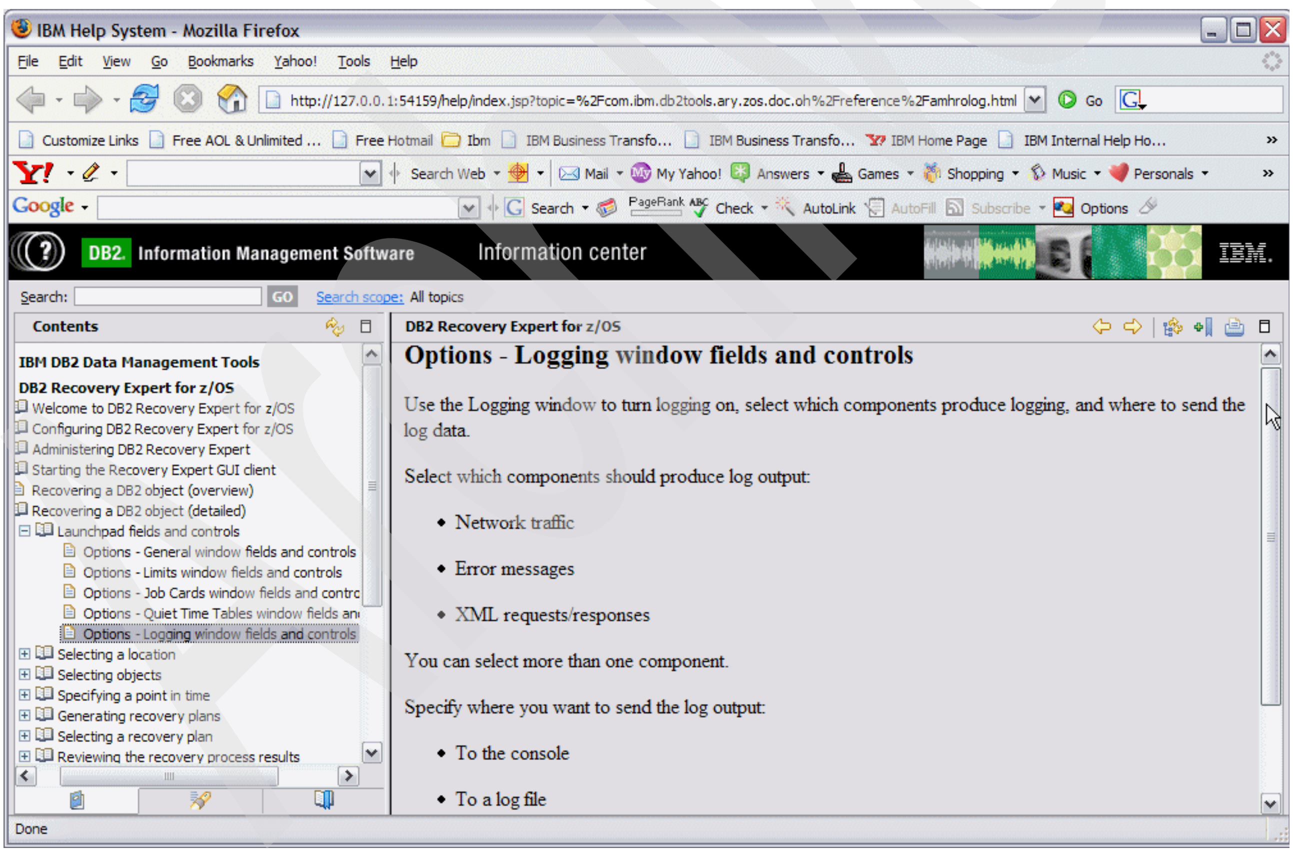

Figure B-14 Context help (2/2)

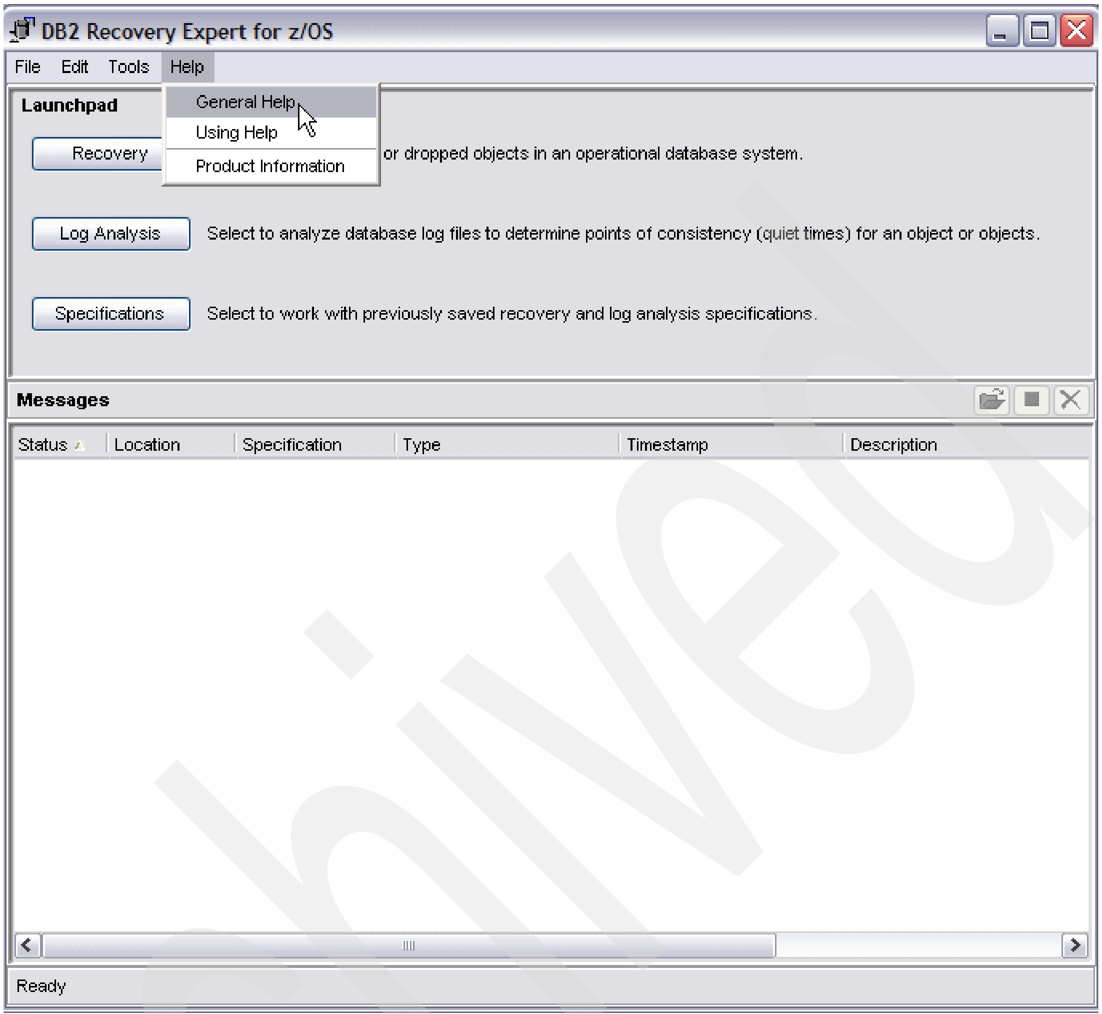

Figure B-15 General help (1/2)

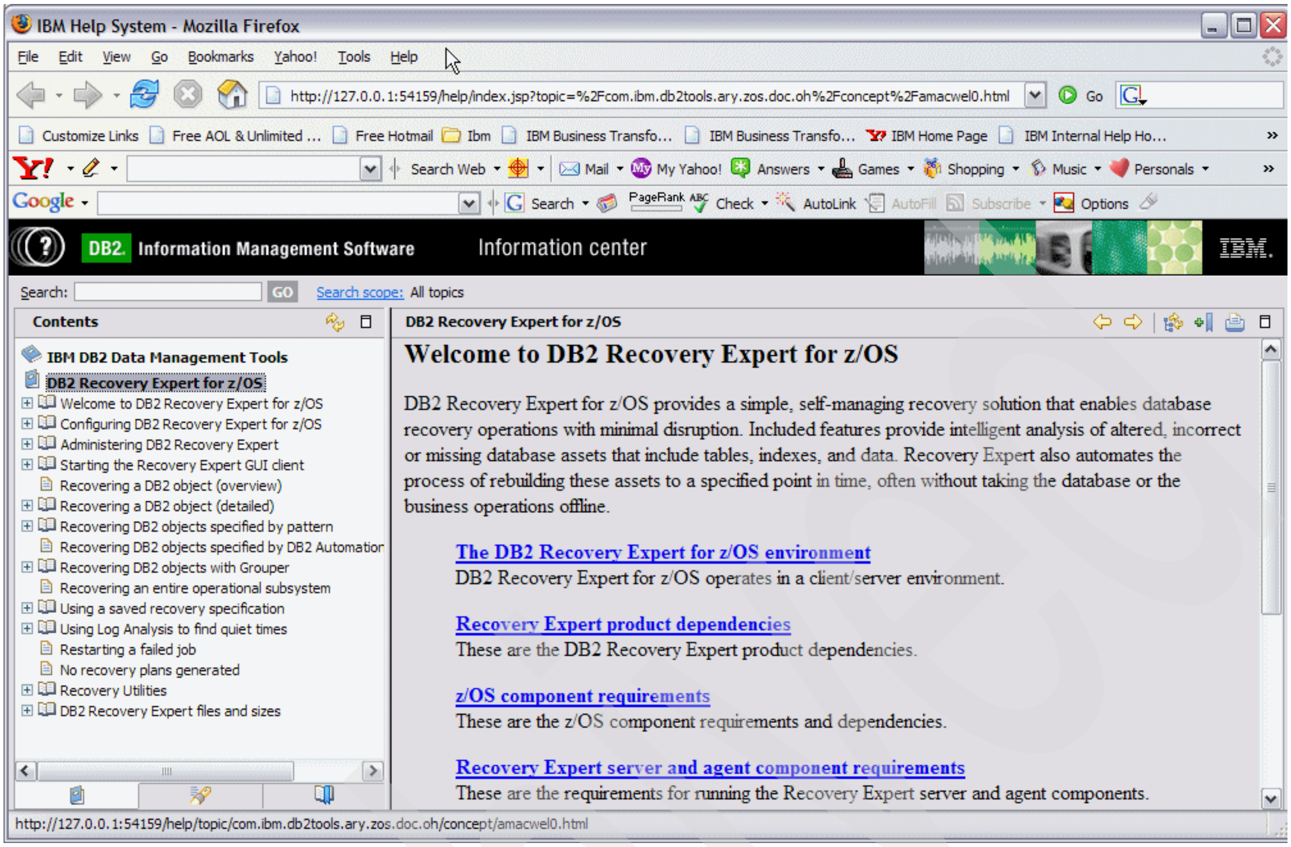

Figure B-16 General help (2/2)

B.4 Recovery to current

The triggering event for recovering an object or set of objects to current is usually media failure, that is, an input/output (I/O) error or volume failure. Typically, when a single object such as a table space is recovered to current, its dependent objects such as indexes or other table spaces in the same referential set are not recovered because they are considered to be unaffected by the error. However, Recovery Expert for z/OS assumes that all dependent objects such as other table spaces in the referential set, and all indexes[2] have to be recovered as well, and therefore generates appropriate control statements to do so. It also automatically generates CHECK DATA control statements for table spaces that are part of a referential set.

The following windows guide you through a typical recovery of an object to current. To minimize clutter, we do not show all the window interactions here.

1. After you log on to the Recovery Expert for z/OS, click **Recovery** in the Launchpad window shown in Figure B-17 on page 242. This takes you to the location selection window.
2. In the location selection window in Figure B-18 on page 243, click **DB8W**.

[2] Indexes are actually rebuilt using the REBUILD INDEX statement.

3. This opens the login prompt to supply the user ID and password, as shown in Figure B-19 on page 243. In addition, you can use this window to specify whether the password must be saved in which case this prompt is bypassed the next time you select this location. Click **OK** in Figure B-19 on page 243 to view details of the location selected, as shown in Figure B-20 on page 244. Click **Next** to proceed to specify the objects that you want to recover.
4. Figure B-21 on page 245 through Figure B-23 on page 247 show the selection of the SNGLT1TS table space as the object to recover. Click **Next** to specify the point-in-time to recover this table space to.
5. Select the **Current** radio button in Figure B-24 on page 248 and click **Next** to generate the potential recovery plans.
6. Click **Generate** button in Figure B-25 on page 249, which presents the recovery plan generation options.
7. Select the required options in Figure B-26 on page 249 through Figure B-30 on page 251. You are then presented with four possible recovery plans shown in the ascending order of efficiency, as shown in Figure B-31 on page 252.
8. Highlight **Plan 1** and click the **Validate** button to verify that the required resources exist. However, a successful validation (Figure B-32 on page 252) does not guarantee successful execution of the selected recovery plan.
9. You can view the generated job control language (JCL), as shown in Figure B-33 on page 253 through Figure B-36 on page 254, and export it to a file, as shown in Figure B-37 on page 255 and Figure B-38 on page 255. Click **Run** to run the generated JCL, as shown in Figure B-39 on page 255. You can view the results of the execution, as shown in Figure B-40 on page 256 through Figure B-42 on page 258.
10. The status of the various operations performed are shown in the Messages section of the Launchpad window, as shown in Figure B-43 on page 259. You can delete all these messages by highlighting the appropriate entry and clicking the icon to delete it, as shown in Figure B-44 on page 260 and Figure B-45 on page 261.

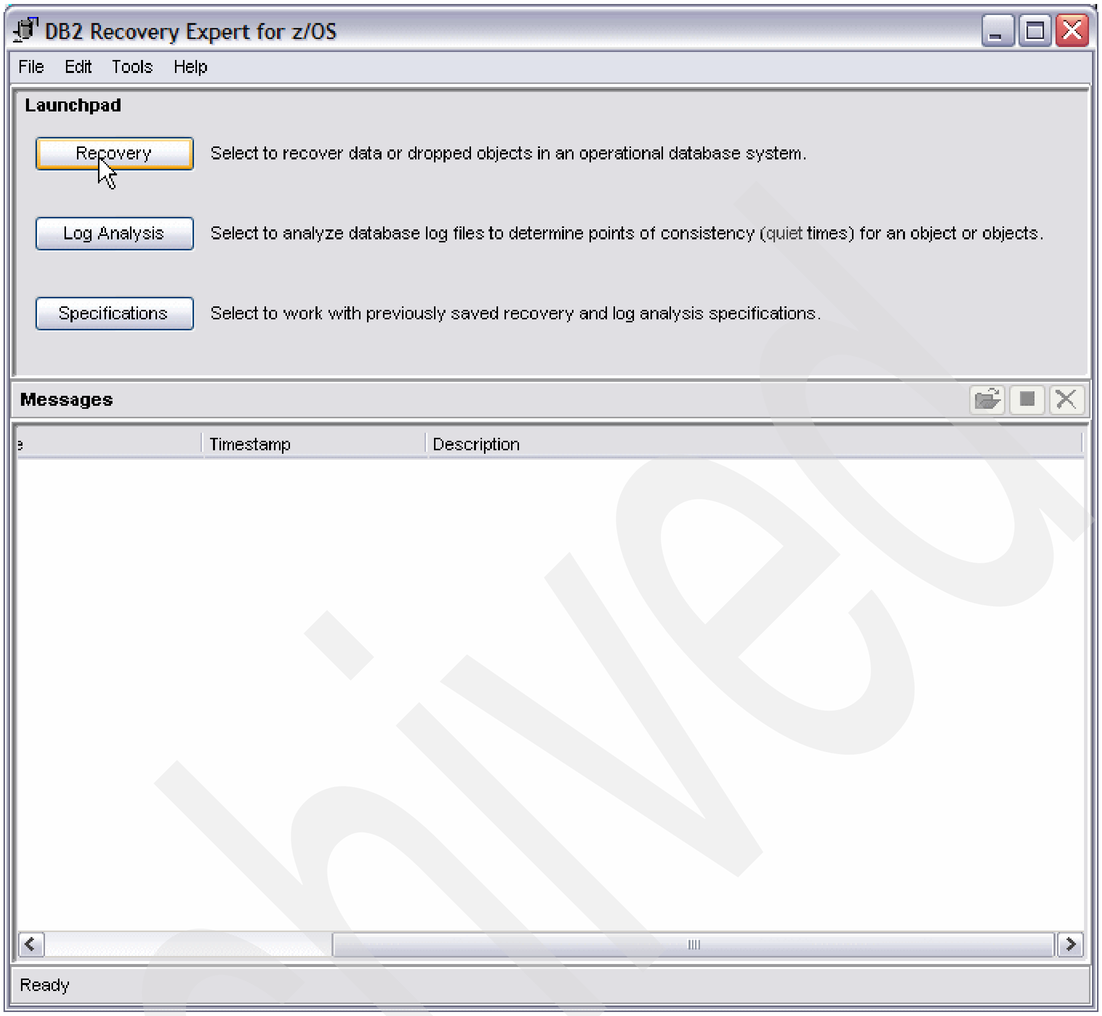

Figure B-17 Launchpad

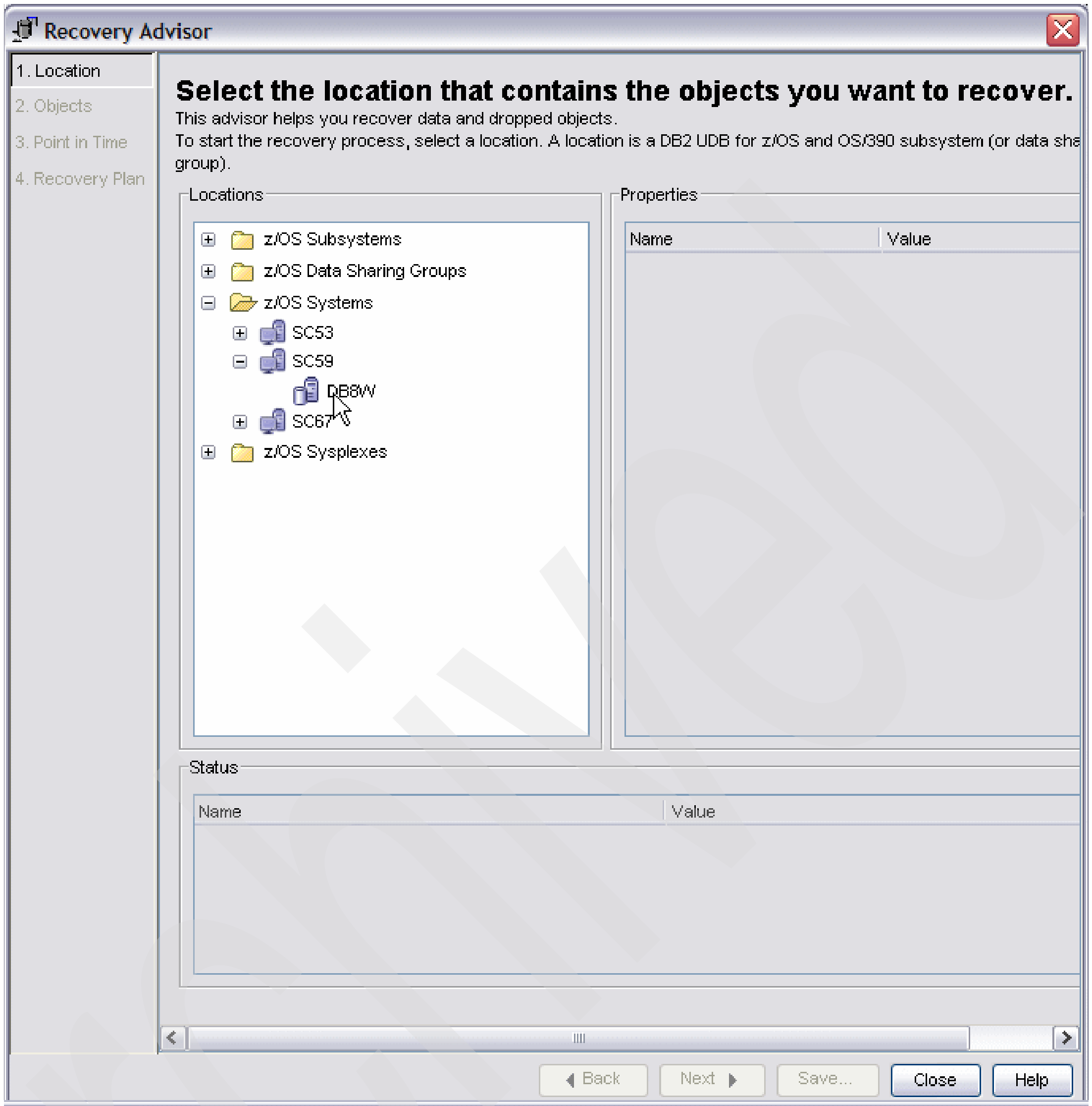

Figure B-18 Selecting the location that contains the objects you want to recover

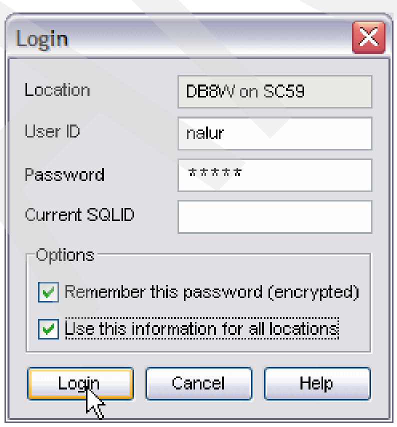

Figure B-19 Login information

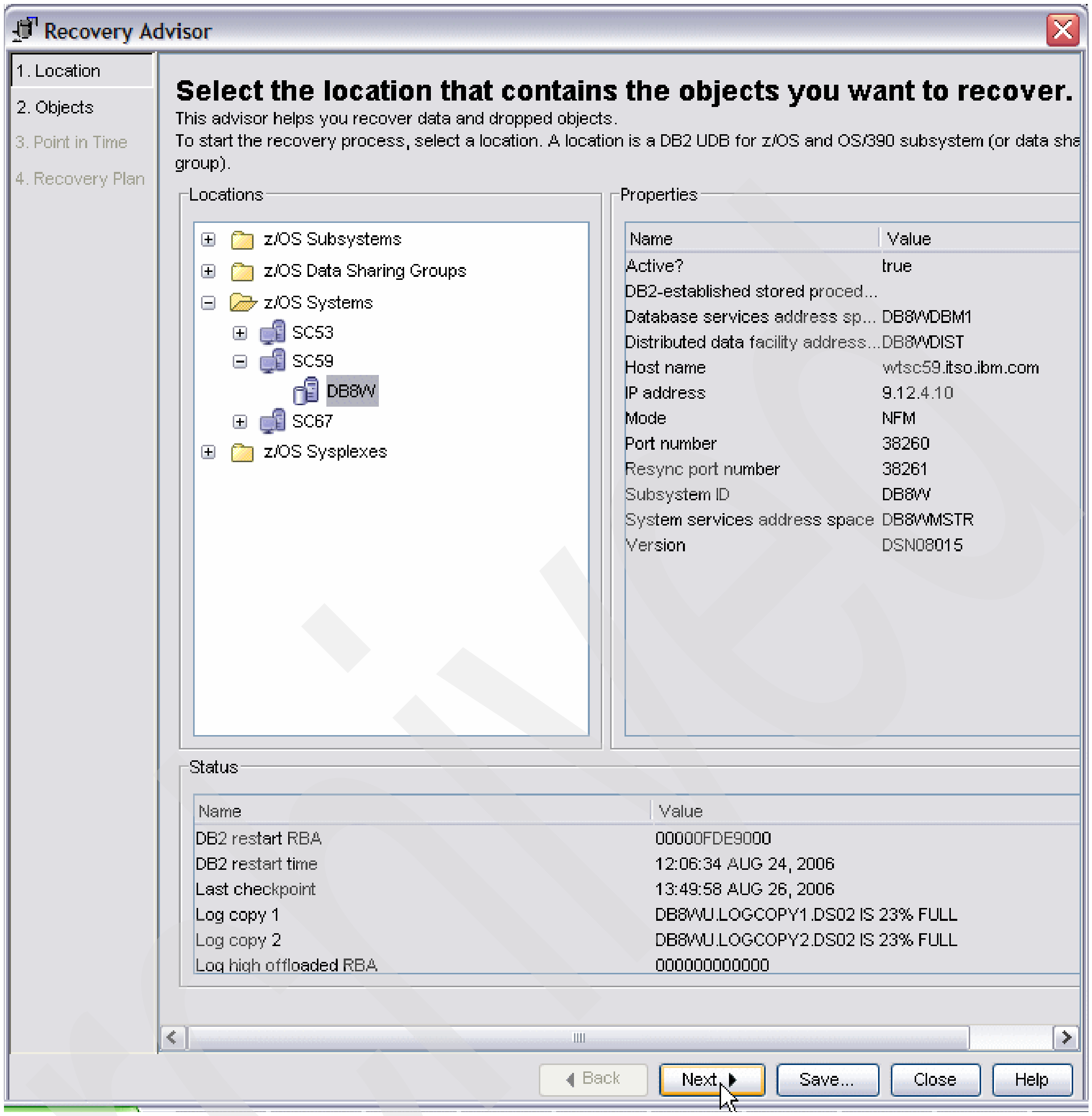

Figure B-20 Selected location details

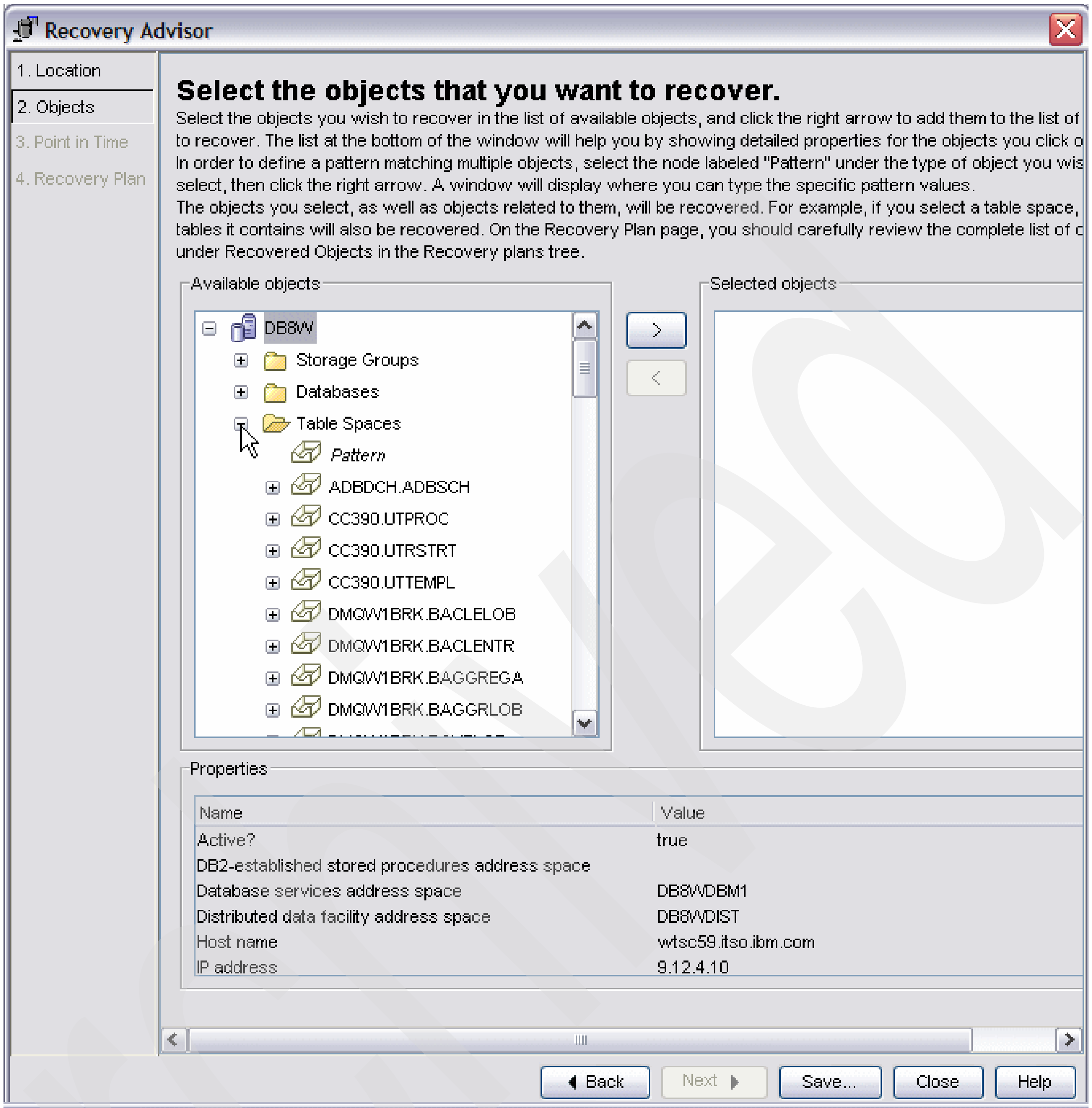

Figure B-21 Selecting the objects to recover (1/3)

> **Note:** For your convenience, you can display, start, and stop certain objects (such as a database and table space) in the navigation tree shown in Figure B-21. You can do this by selecting the object and right-clicking it to open the actions possible. However, this is not shown here.

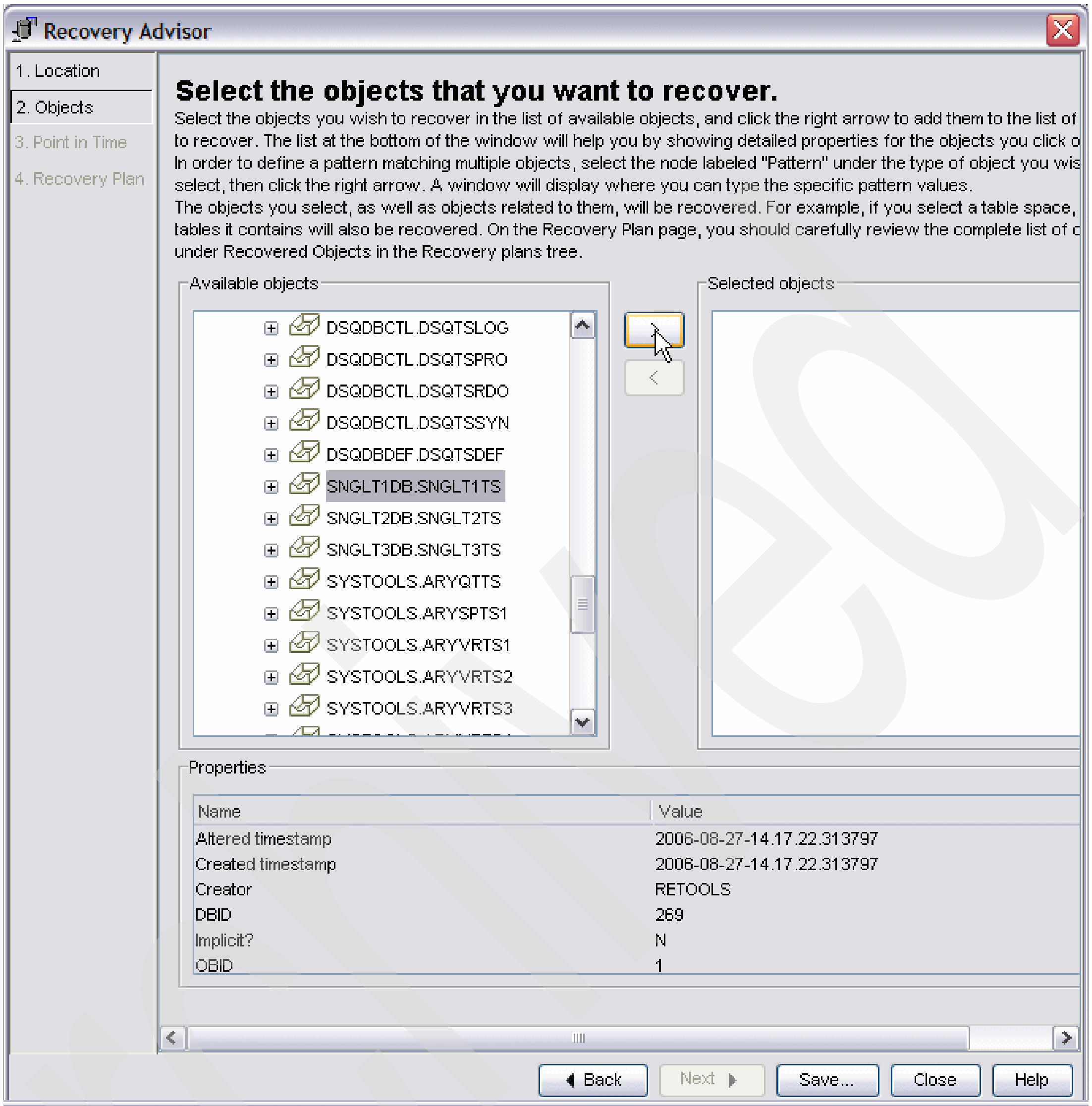

Figure B-22 Selecting the objects to recover (2/3)

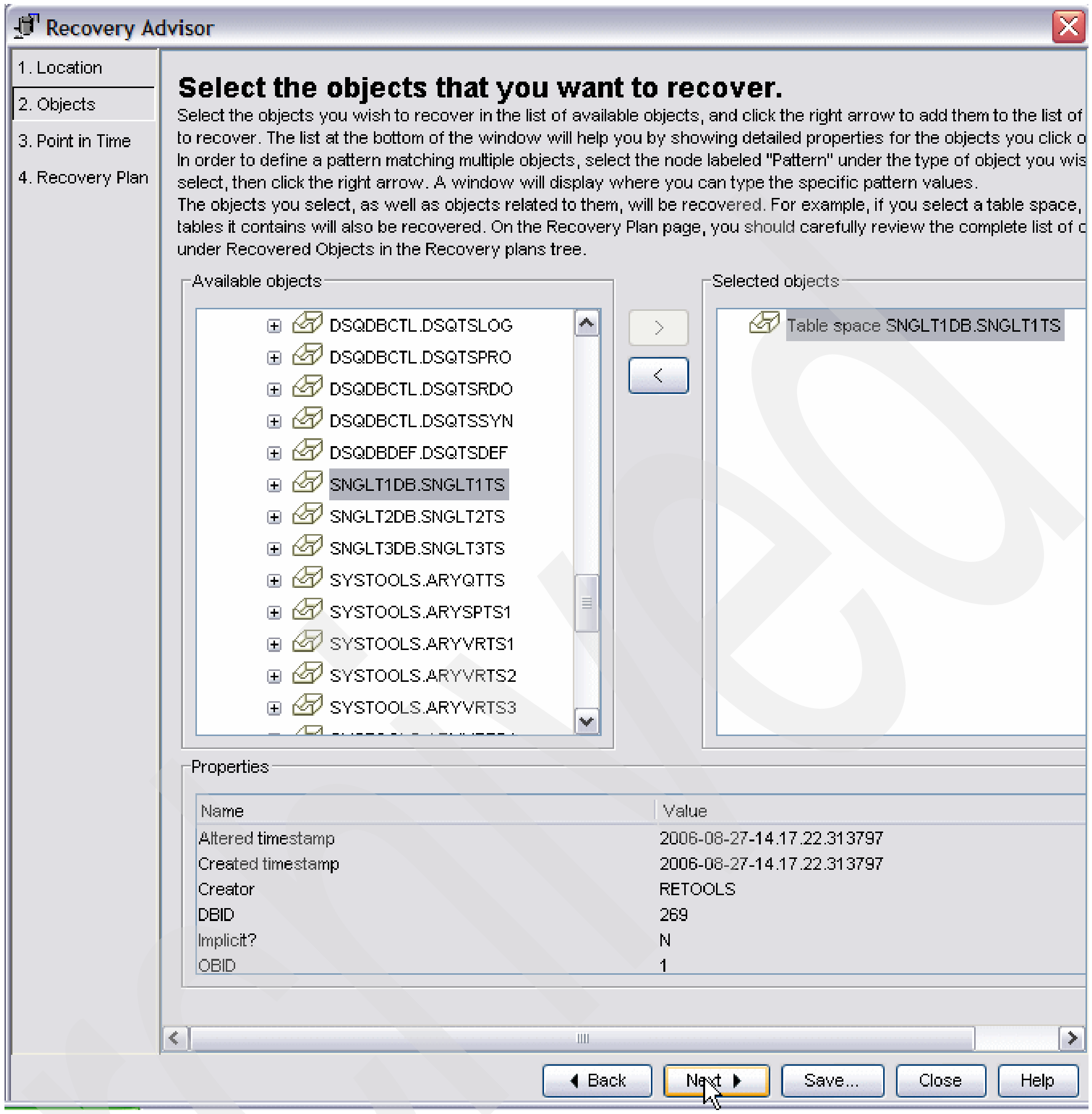

Figure B-23 Selecting the objects to recover (3/3)

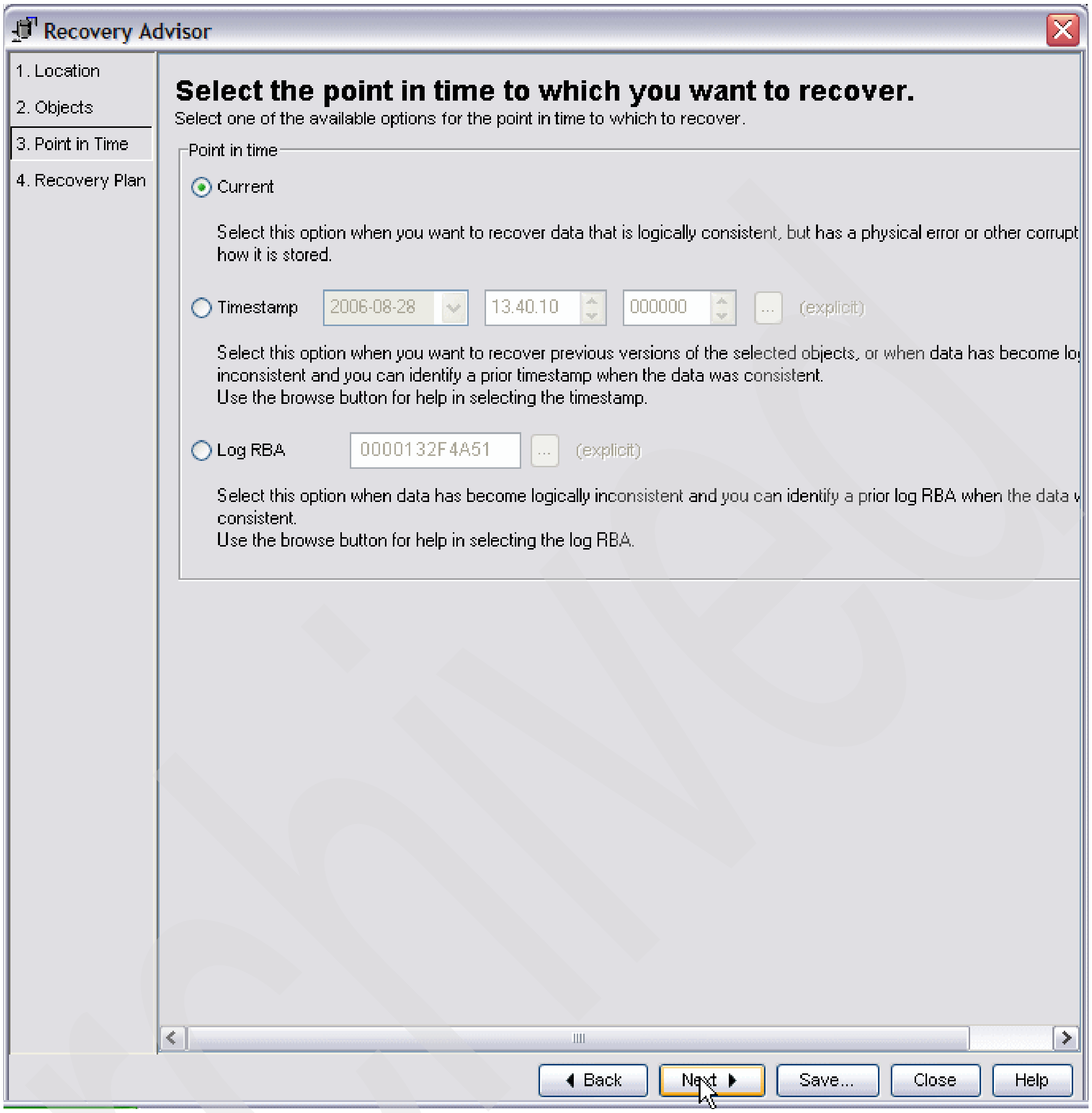

Figure B-24 Selecting the point to recover to

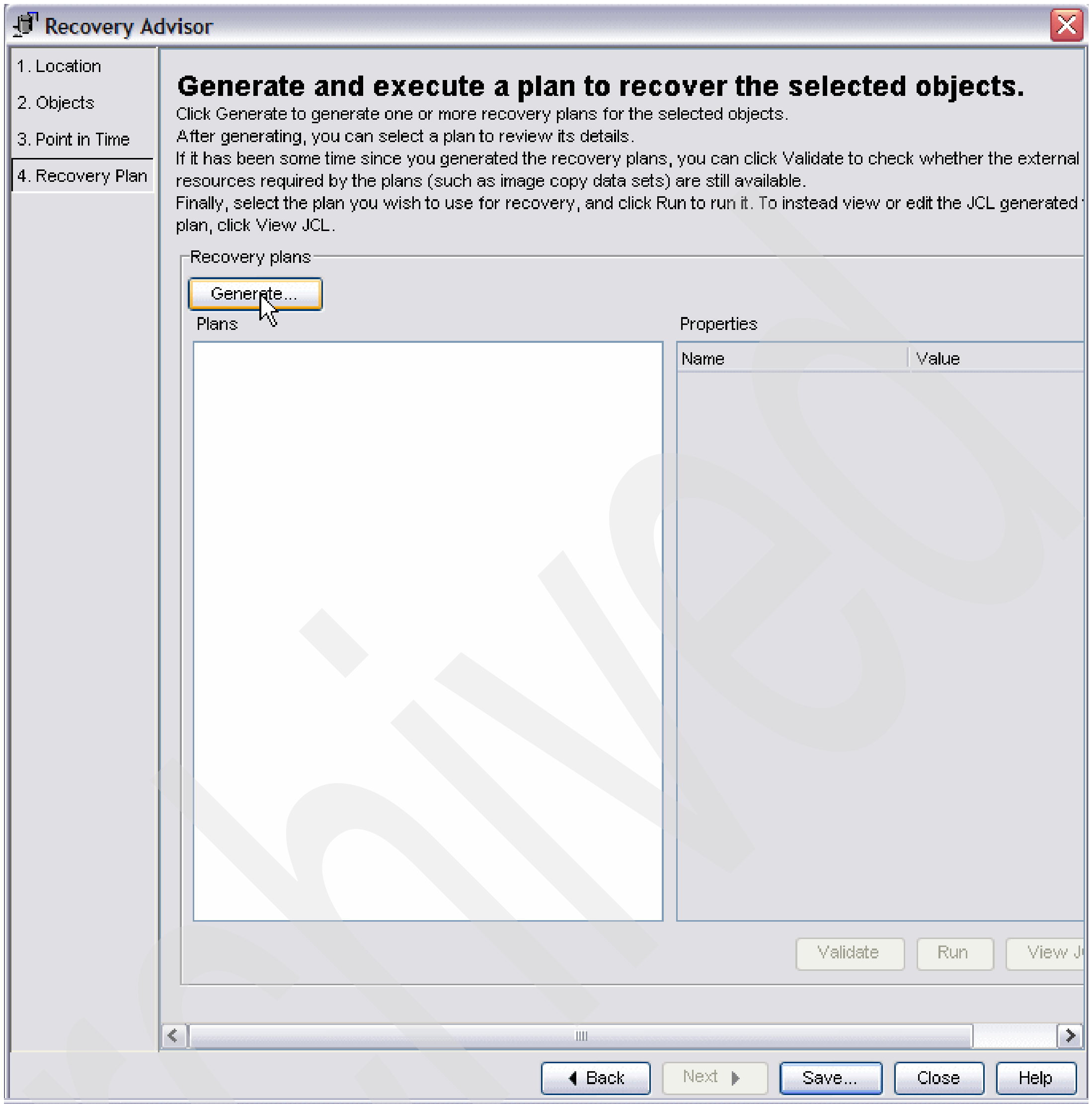

Figure B-25 Generating recovery plans

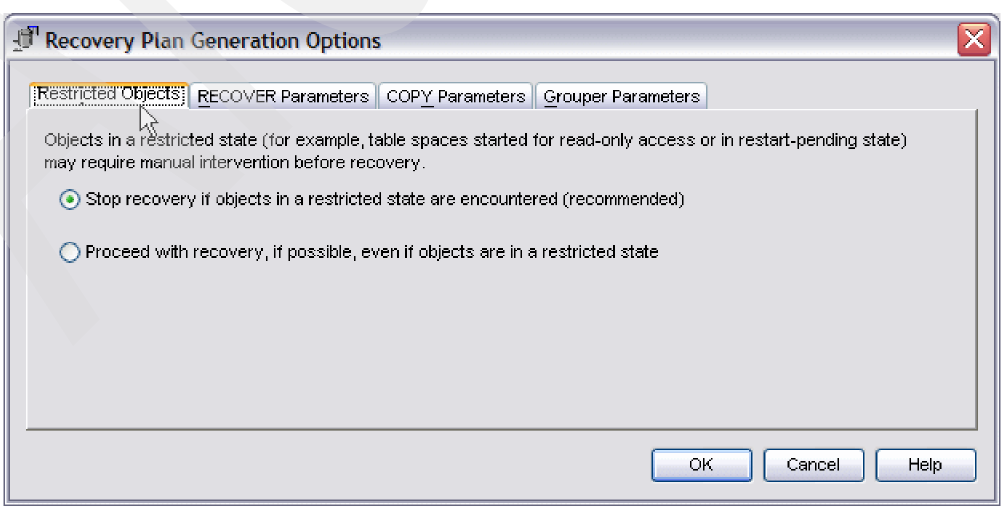

Figure B-26 Recovery plan generation options: Restricted objects

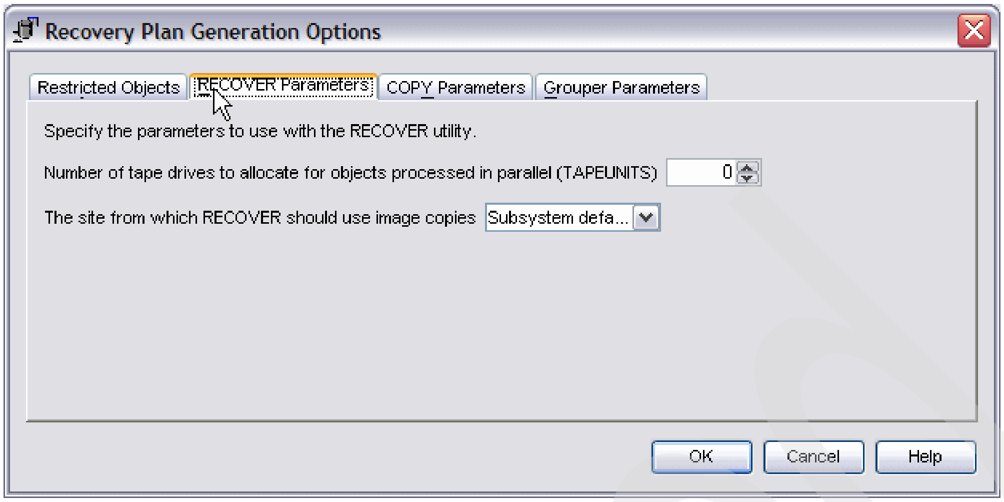

Figure B-27 Recovery plan generation options: Recover parameters

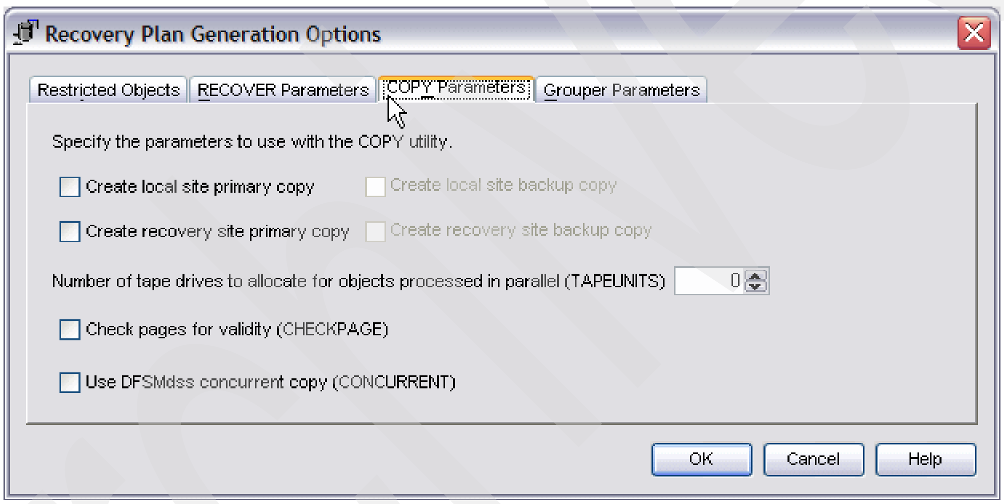

Figure B-28 Recovery plan generation options: Copy parameters

Note: We recommend that you create at least one image copy after recovery by selecting the Create local site primary copy check box.

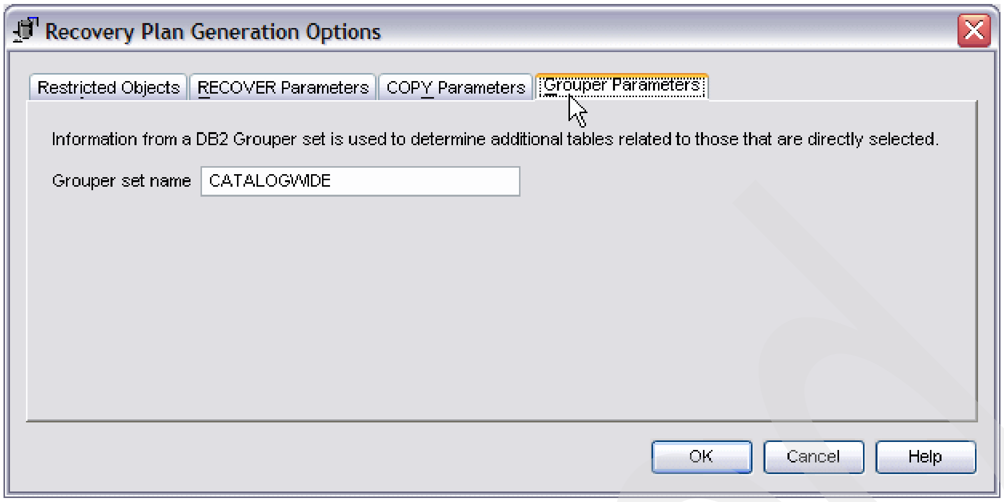

Figure B-29 Recovery plan generation options: Grouper parameters

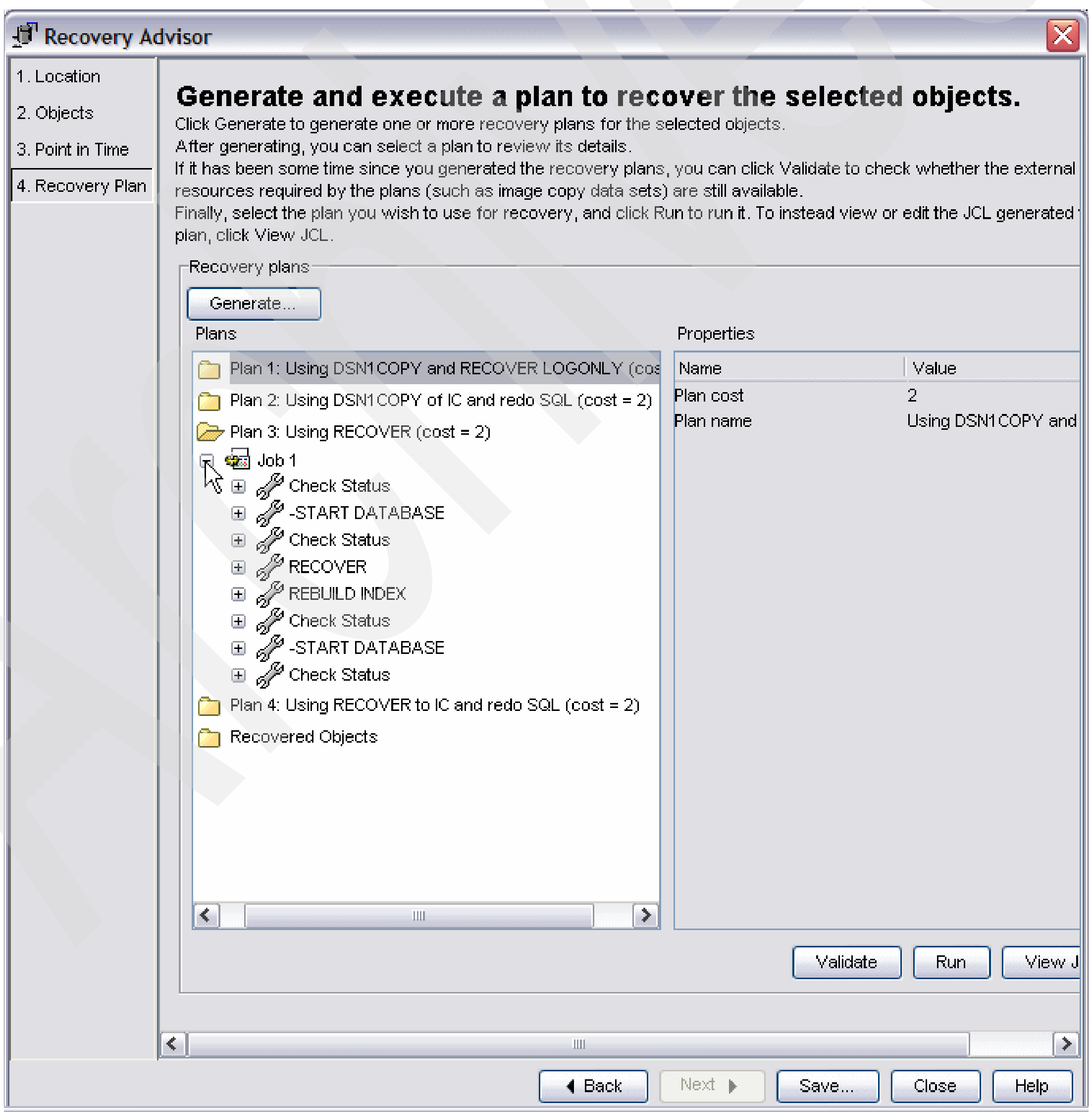

Figure B-30 Generated recovery plans with expanded steps of Plan 3

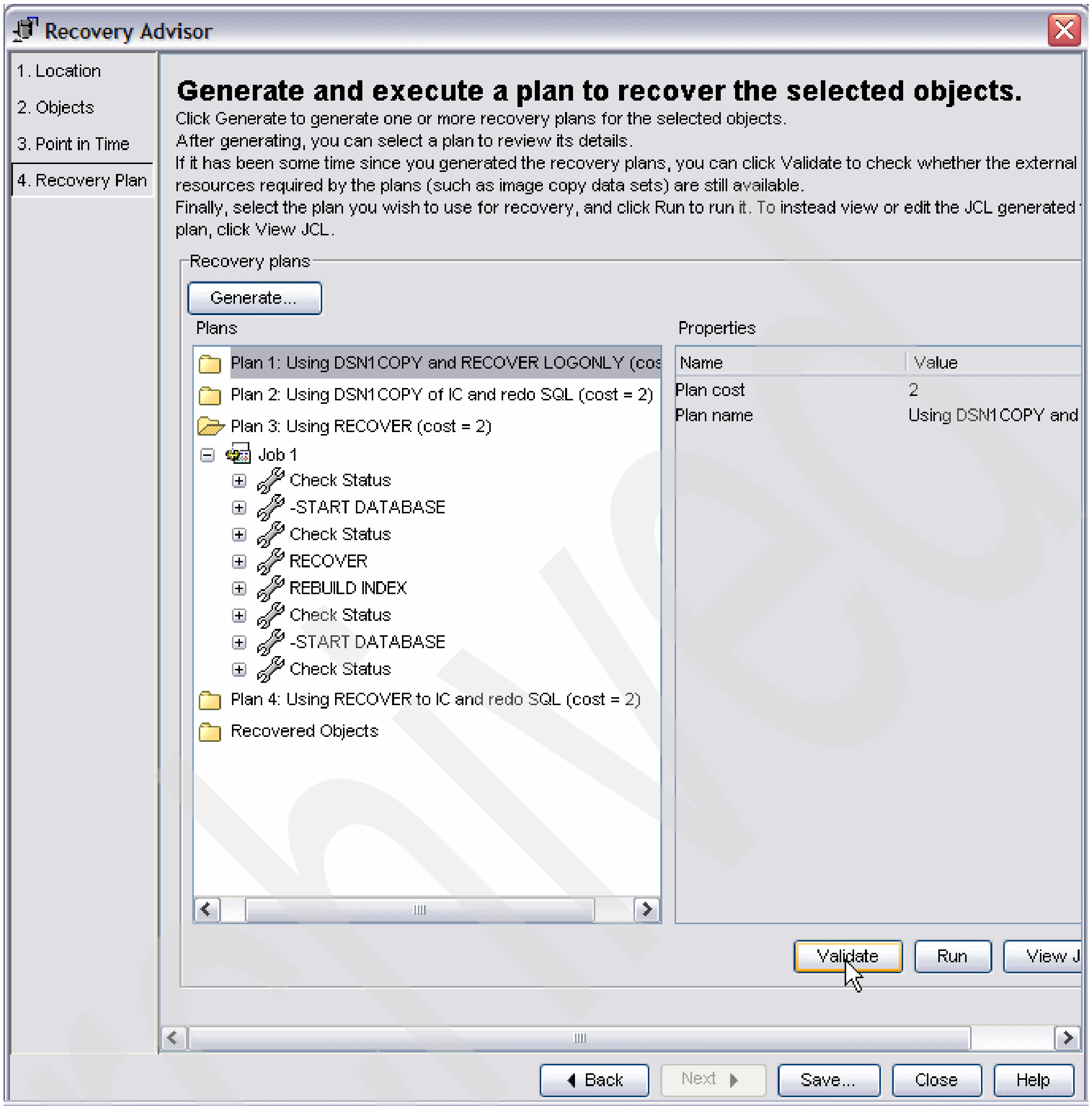

Figure B-31 Validating selected recovery plan

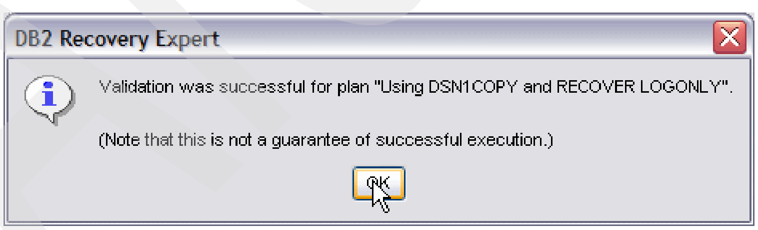

Figure B-32 Validation successful message

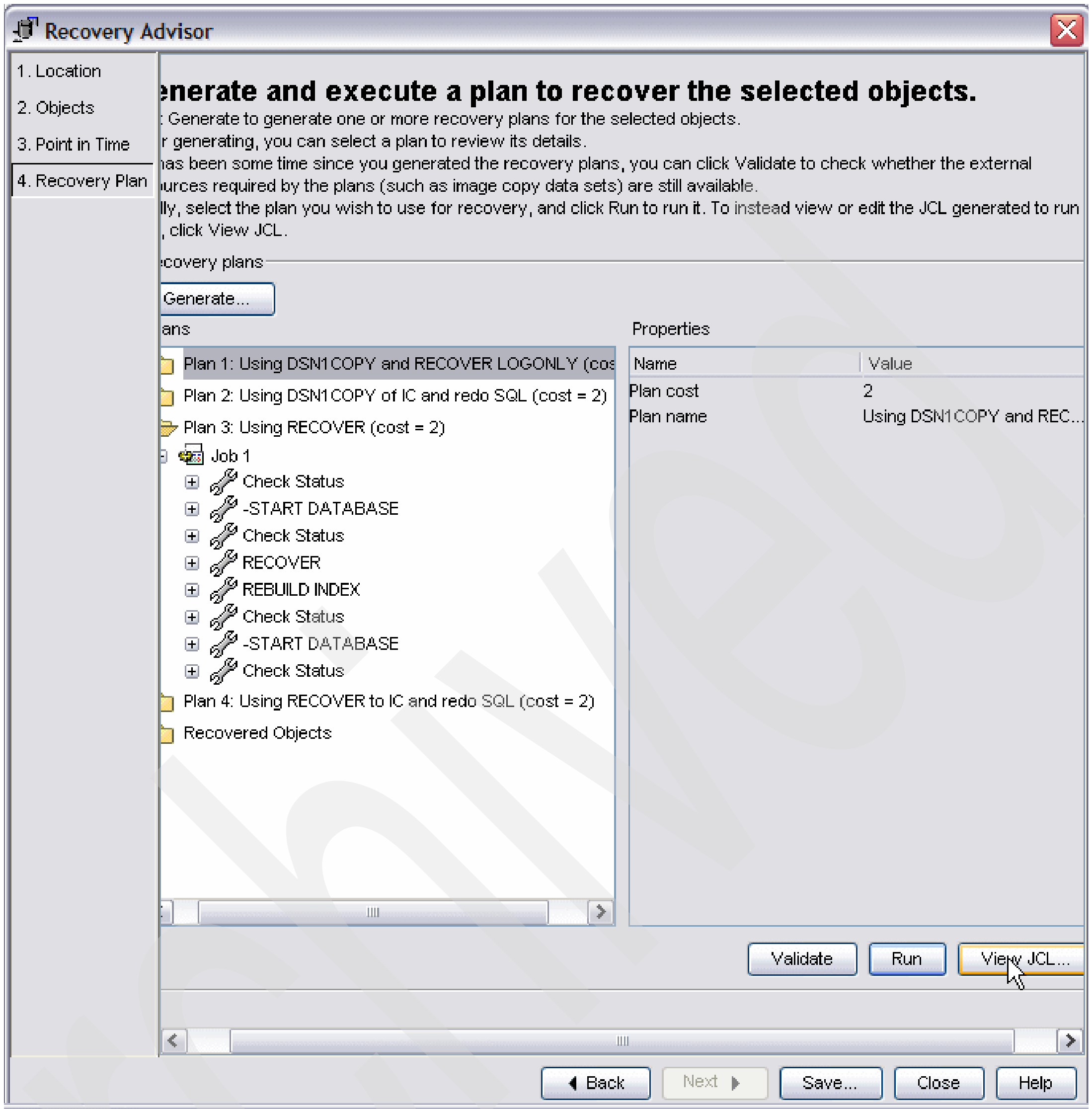

Figure B-33 Viewing (generated) JCL of the selected recovery plan

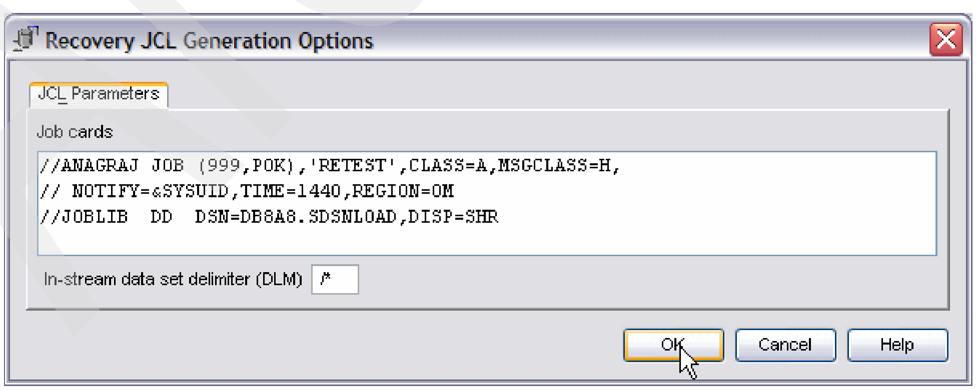

Figure B-34 Generated JCL of the selected recovery plan (1/3)

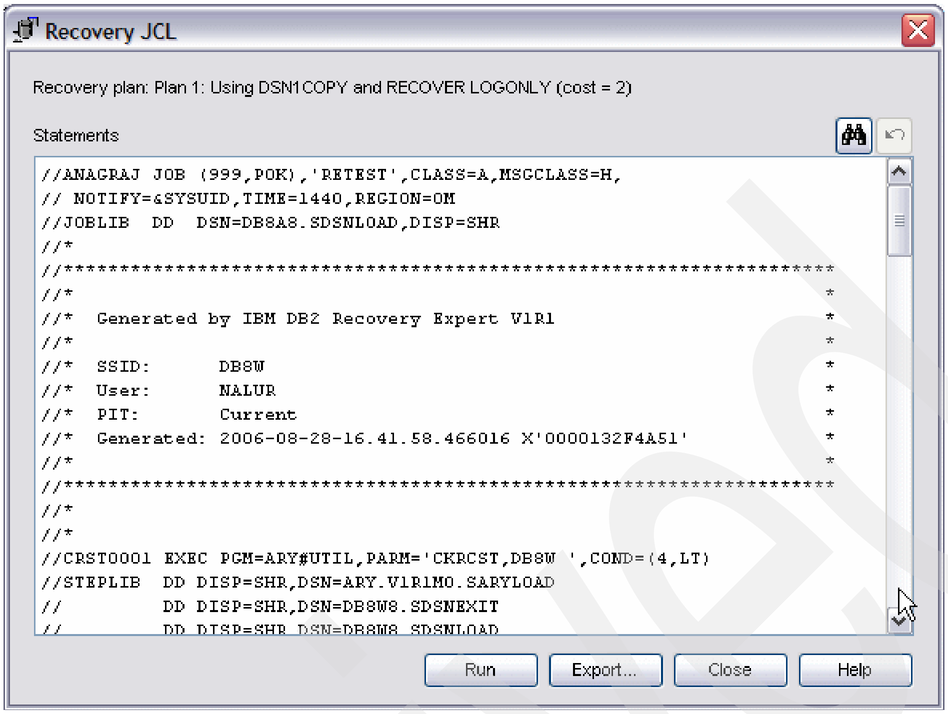

Figure B-35 Generated JCL of the selected recovery plan (2/3)

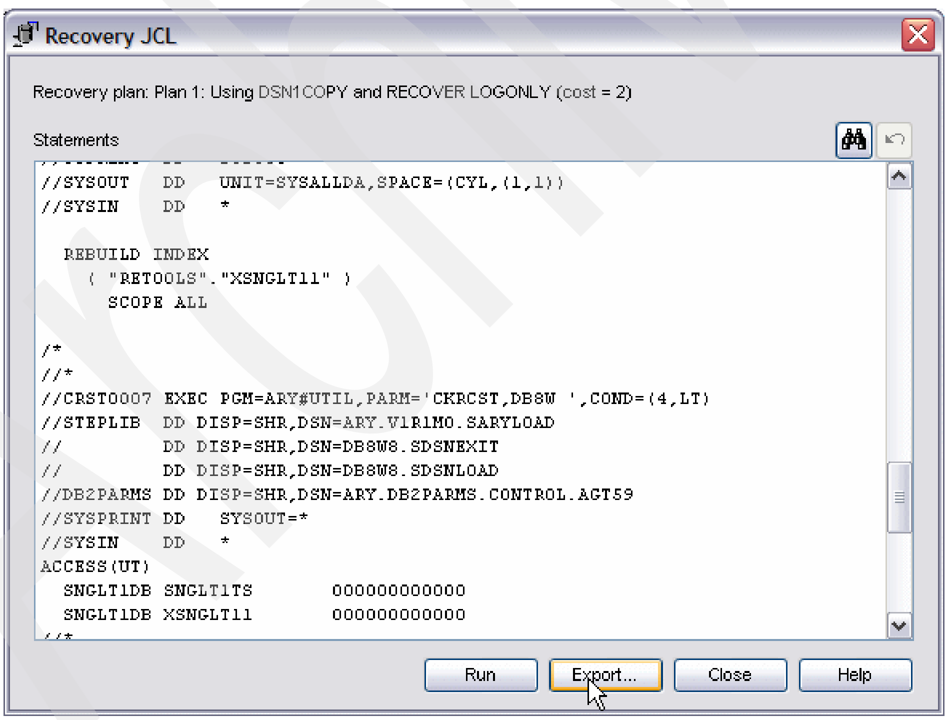

Figure B-36 Generated JCL of the selected recovery plan (3/3)

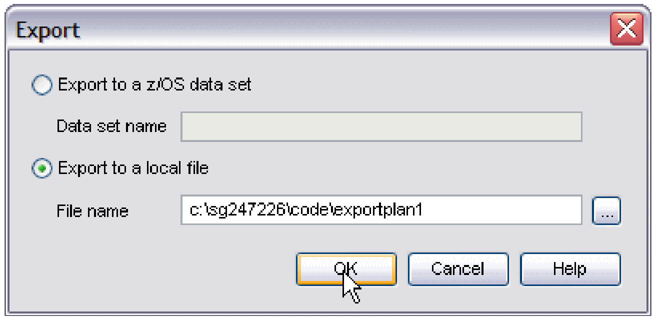

Figure B-37 Exporting generated JCL to a file (1/2)

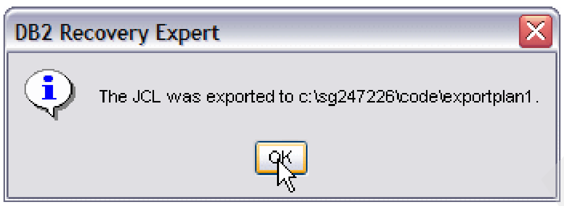

Figure B-38 Exporting generated JCL to a file (2/2)

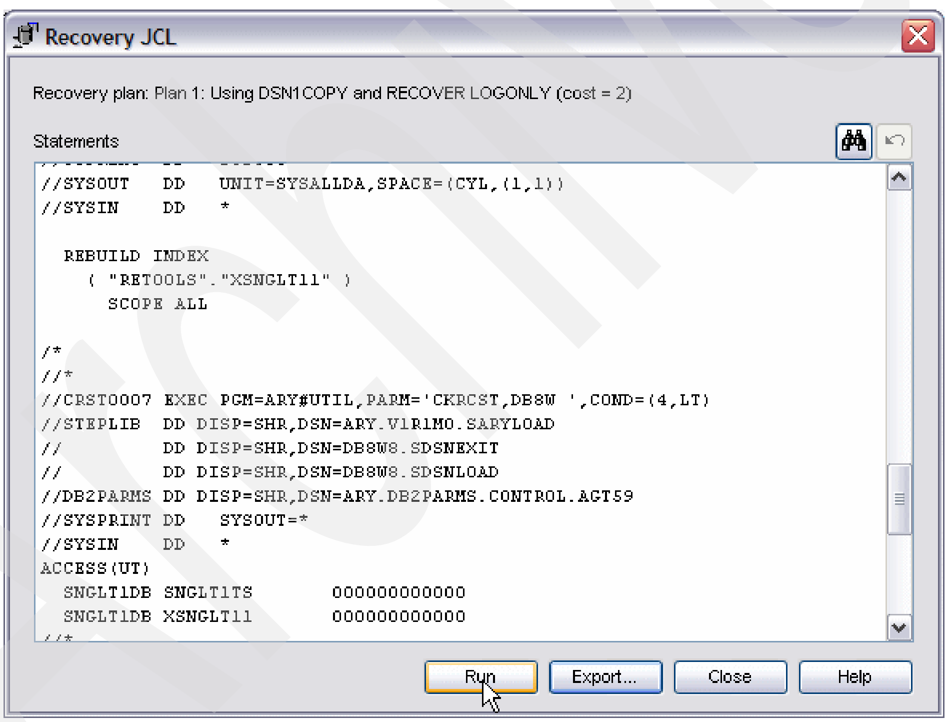

Figure B-39 Running the generated JCL for the selected recovery plan

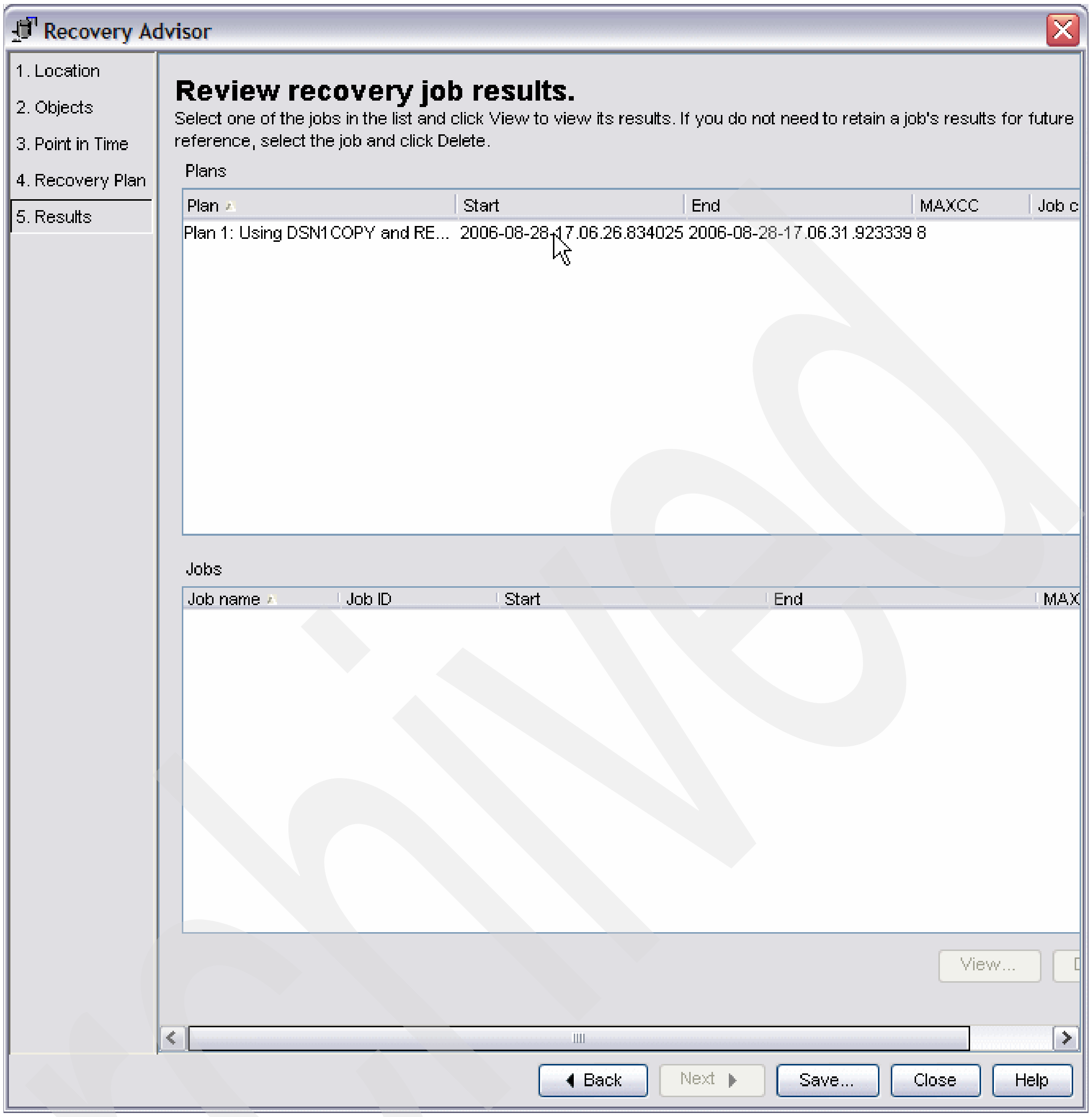

Figure B-40 Reviewing recovery job results (1/3)

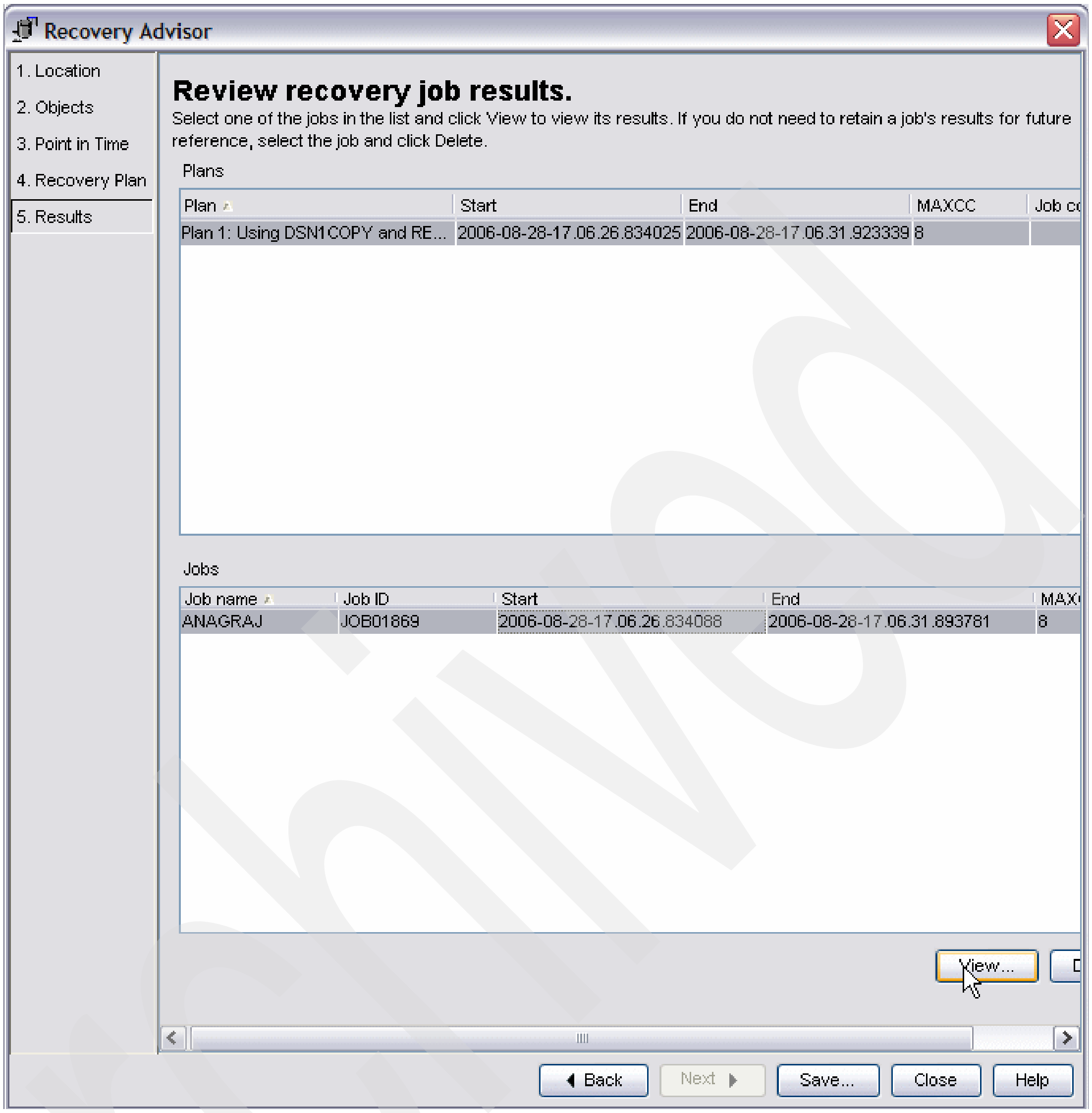

Figure B-41 Reviewing recovery job results (2/3)

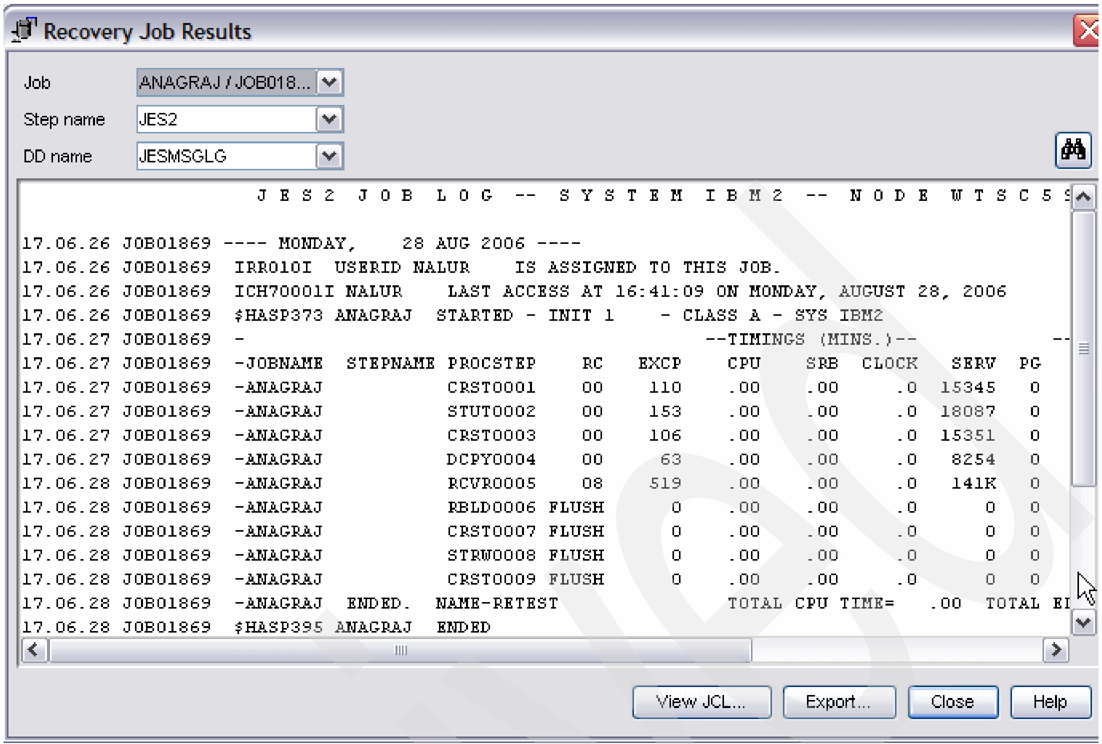

Figure B-42 Reviewing recovery job results (3/3)

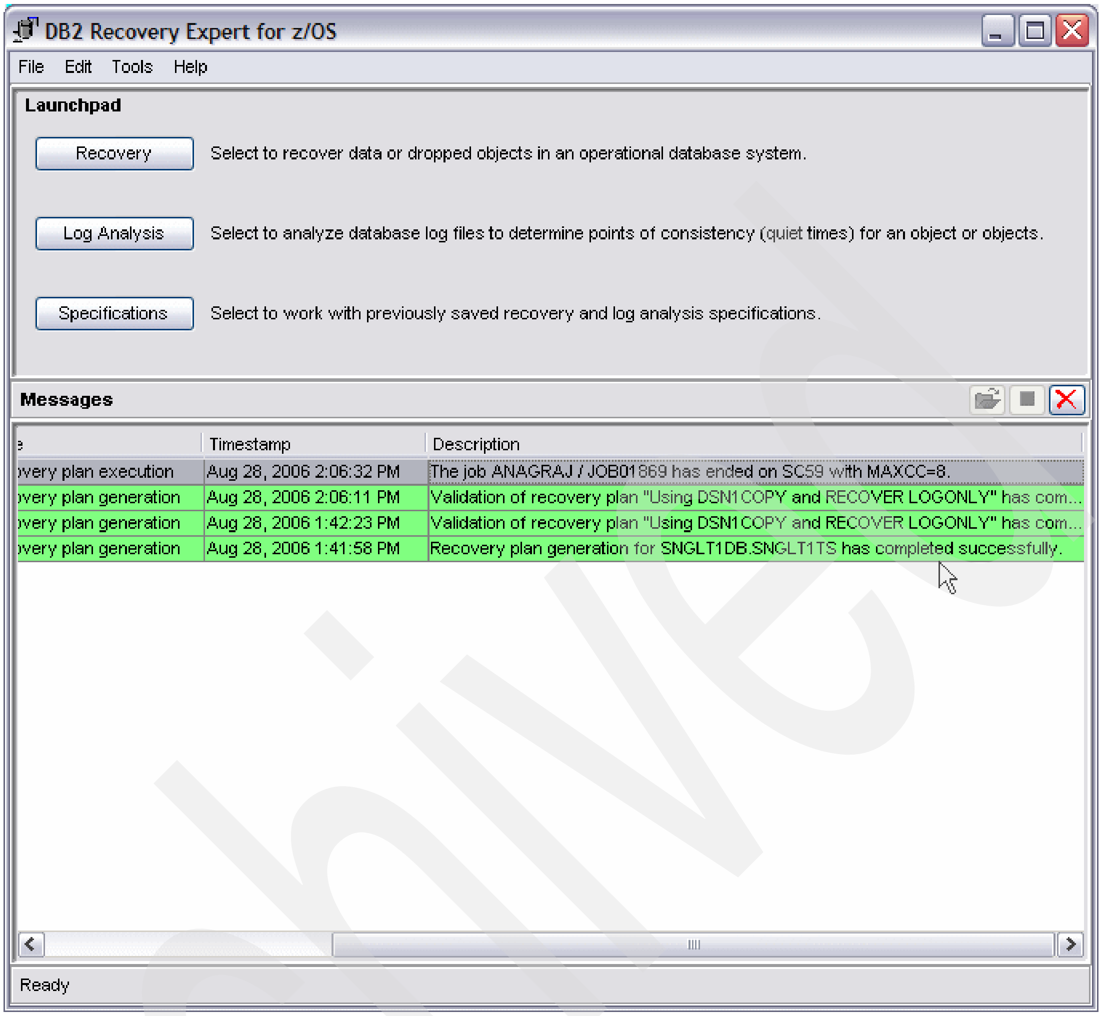

Figure B-43 Status of various tasks performed in the messages section of launchpad window

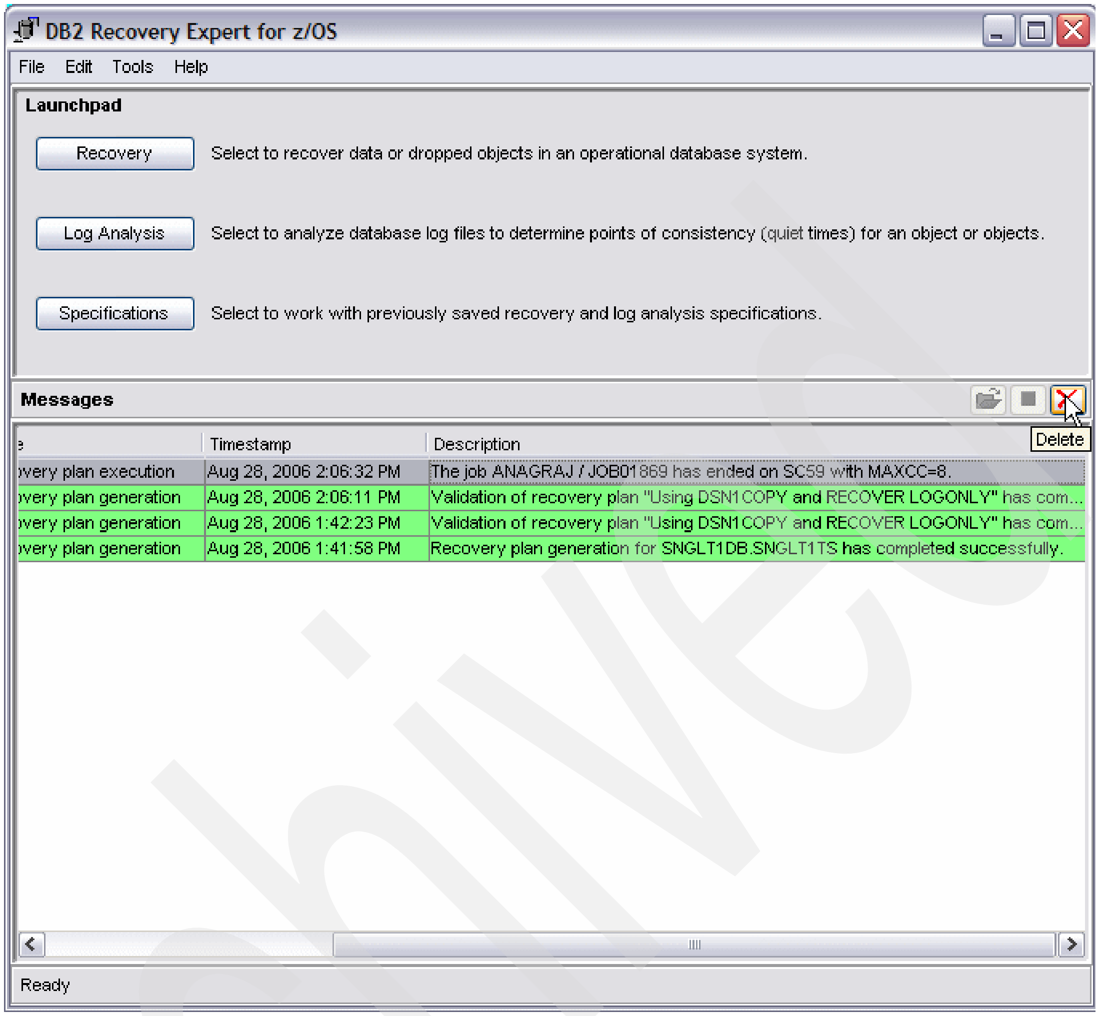

Figure B-44 Deleting a highlighted message in the launchpad window (1/2)

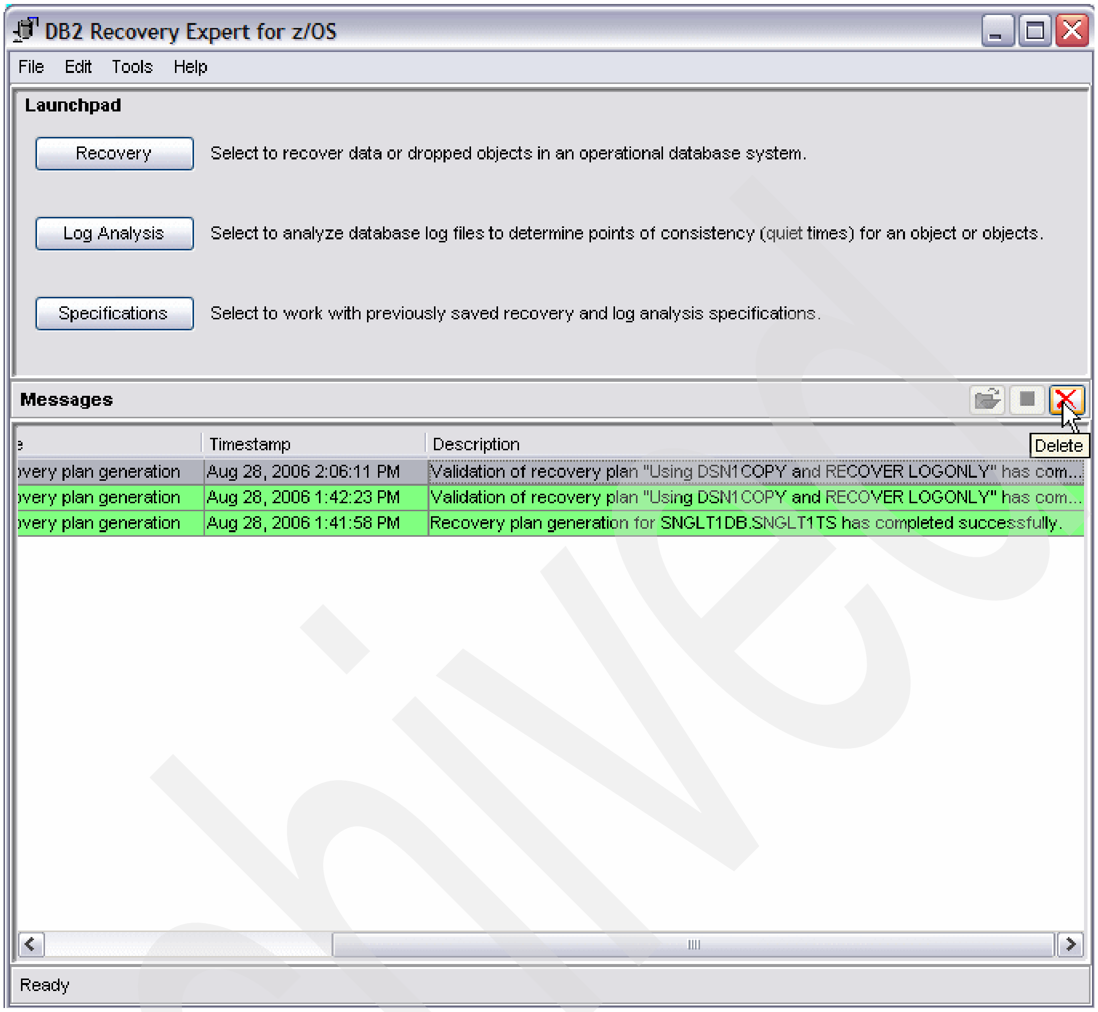

Figure B-45 Deleting a highlighted message in the launchpad window (2/2)

B.5 Recovery to RBA/LRSN

The triggering event for recovering an object or set of objects to a point-in-time (PIT) is usually application errors or operator errors, and sometimes a site failure which requires disaster recovery[3]. When you recover a single object such as a table space to a PIT, its dependent objects such as indexes or other table spaces in the same referential set or Grouper group *must* also be recovered because they are affected by the recovery action. Recovery Expert for z/OS automatically identifies all table spaces in the referential set when you identify just a single member of the referential set as the target of recovery, and all dependent objects such as indexes. It then generates appropriate control statements to recover all these objects to the PIT. It also automatically generates CHECK DATA control statements for table spaces that are part of a referential set as appropriate.

[3] Recovery Expert for z/OS does not support disaster recovery.

The following windows guide you through a typical recovery of an object to an RBA/LRSN which represents a point-in-time. To minimize clutter, we do not show all window interactions here.

1. After you log on to the Recovery Expert for z/OS, click **Recovery** in the Launchpad window shown in Figure B-46 on page 263. This takes you to the location selection window.
2. In the location selection window in Figure B-47 on page 264, click **DB8W** to view its details. Click **Next** to proceed to specify the objects that you want to recover.
3. Figure B-48 on page 265 through Figure B-50 on page 267 show the selection of the SNGLT1TS table space as the object to recover. Click **Next** to specify the point-in-time to recover this table space to.
4. Select the **Log RBA** radio button in Figure B-51 on page 268. You can directly type the RBA/LRSN if you have determined this by some external means, or click the ellipsis to select from an available list, as shown in Figure B-52 on page 269.
5. When you select the **Recover history events** button, the list of entries (image copy with share level reference and quiesce points) in SYSIBM.SYSCOPY that represent points-of-consistency are displayed, as shown in Figure B-52 on page 269.
6. To see image copies that were created with the share level change option, select the **Show events that are not points of consistency** check box, as shown in Figure B-53 on page 269.
7. Highlight the entry to recover to, which is a full image copy with share level reference in Figure B-54 on page 270; it has a Start RBA value (0000130D8CBB) associated with it. Click **OK** to proceed to Figure B-55 on page 270, which plugs the Start RBA value of the full image copy entry that is selected. When you click **Next** in Figure B-55 on page 270, it issues a warning message about the recovery point potentially not being a point-of-consistency, as shown in Figure B-56 on page 271. Click **Yes** to proceed to generate recovery plans.
8. Click **Generate** in Figure B-57 on page 271 and bypass the recovery plan generation options (click **OK** in Figure B-58 on page 272) to view the generated recovery plans, as shown in Figure B-59 on page 272. The execution of a selected recovery plan is not shown here. Figure B-60 on page 273 shows the successful execution messages in the Messages section of the Launchpad window.

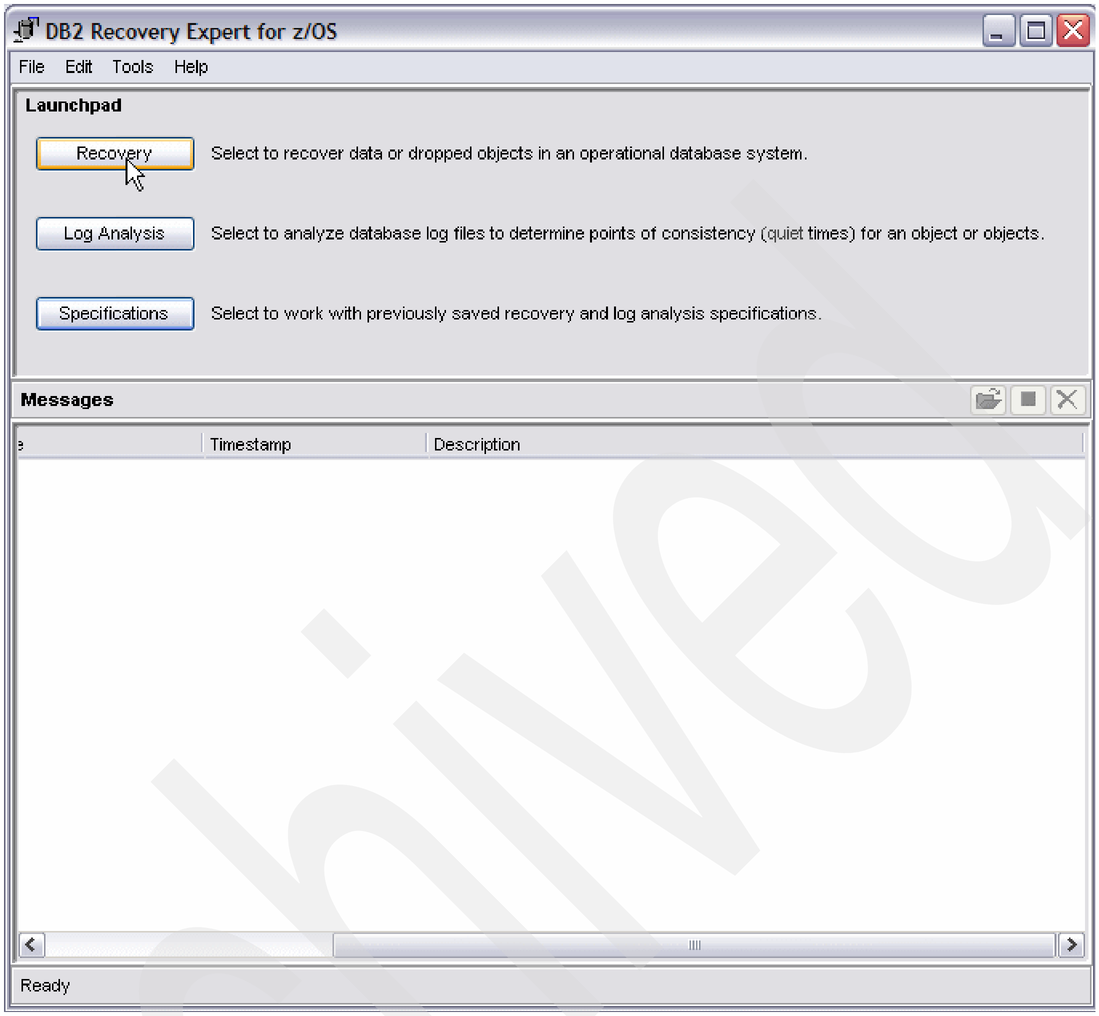

Figure B-46 Launchpad

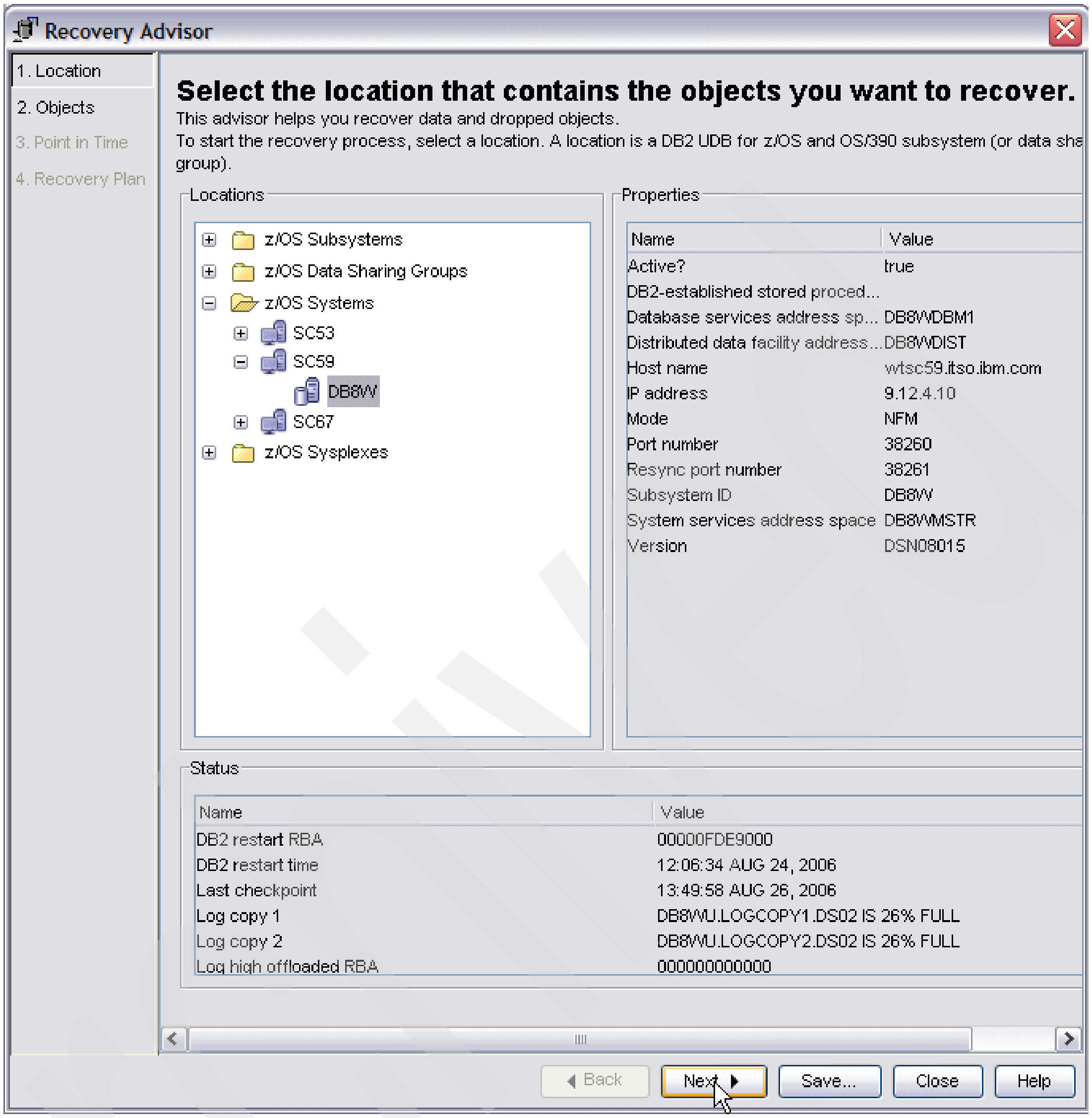

Figure B-47 Selecting the DB8W location

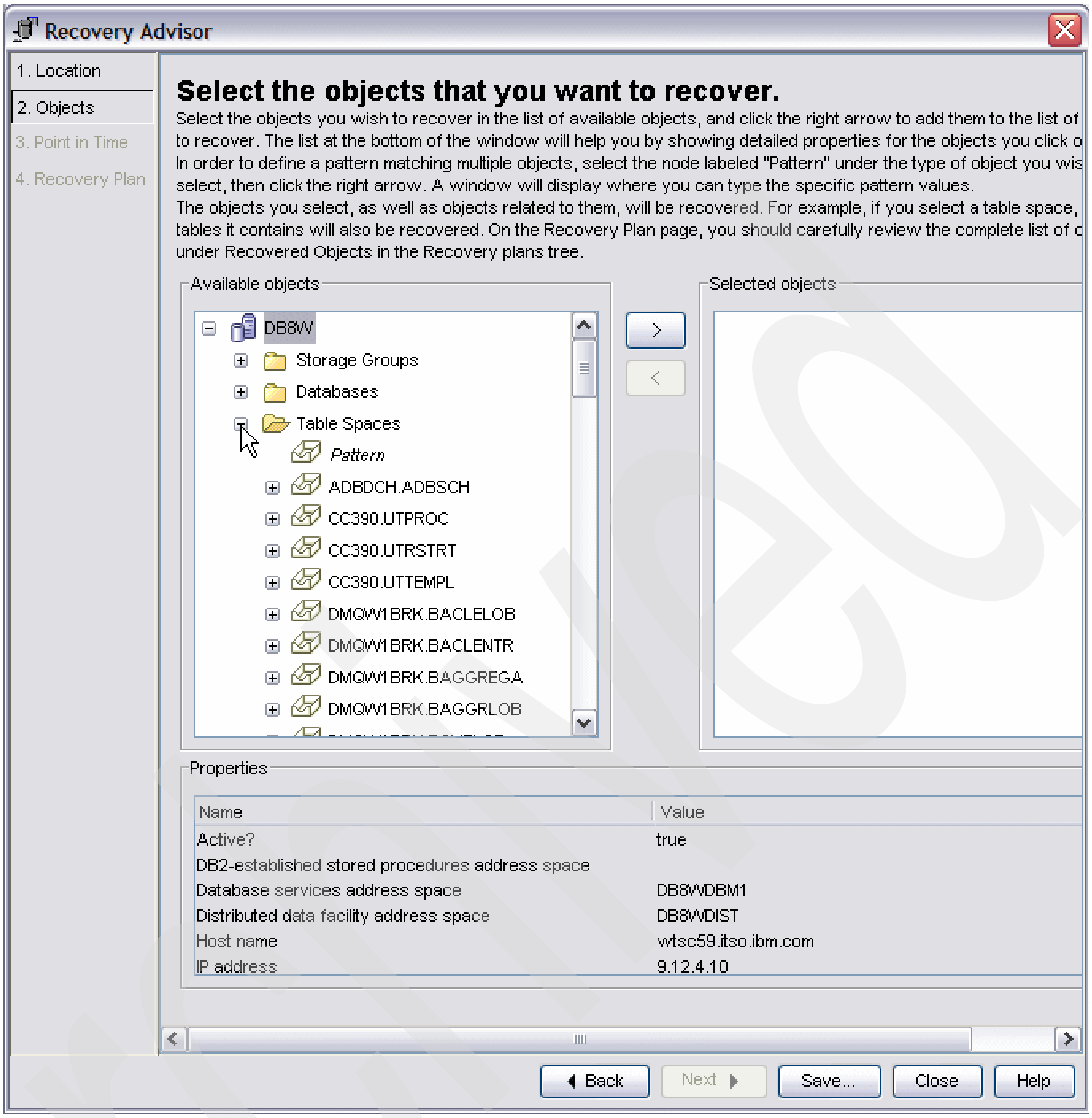

Figure B-48 Selecting the object SINGLT1TS table space to recover (1/3)

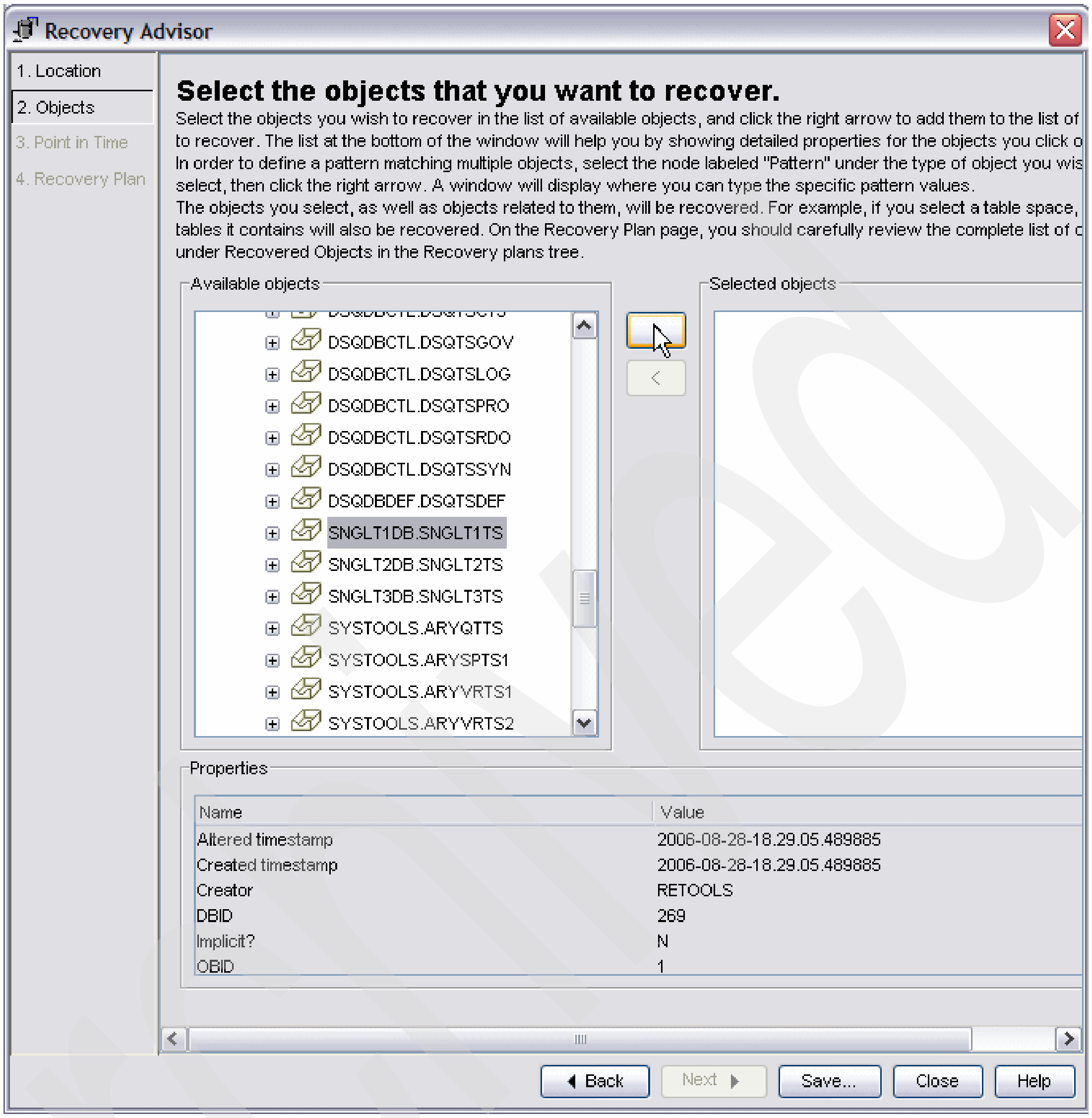

Figure B-49 Selecting the object SINGLT1TS table space to recover (2/3)

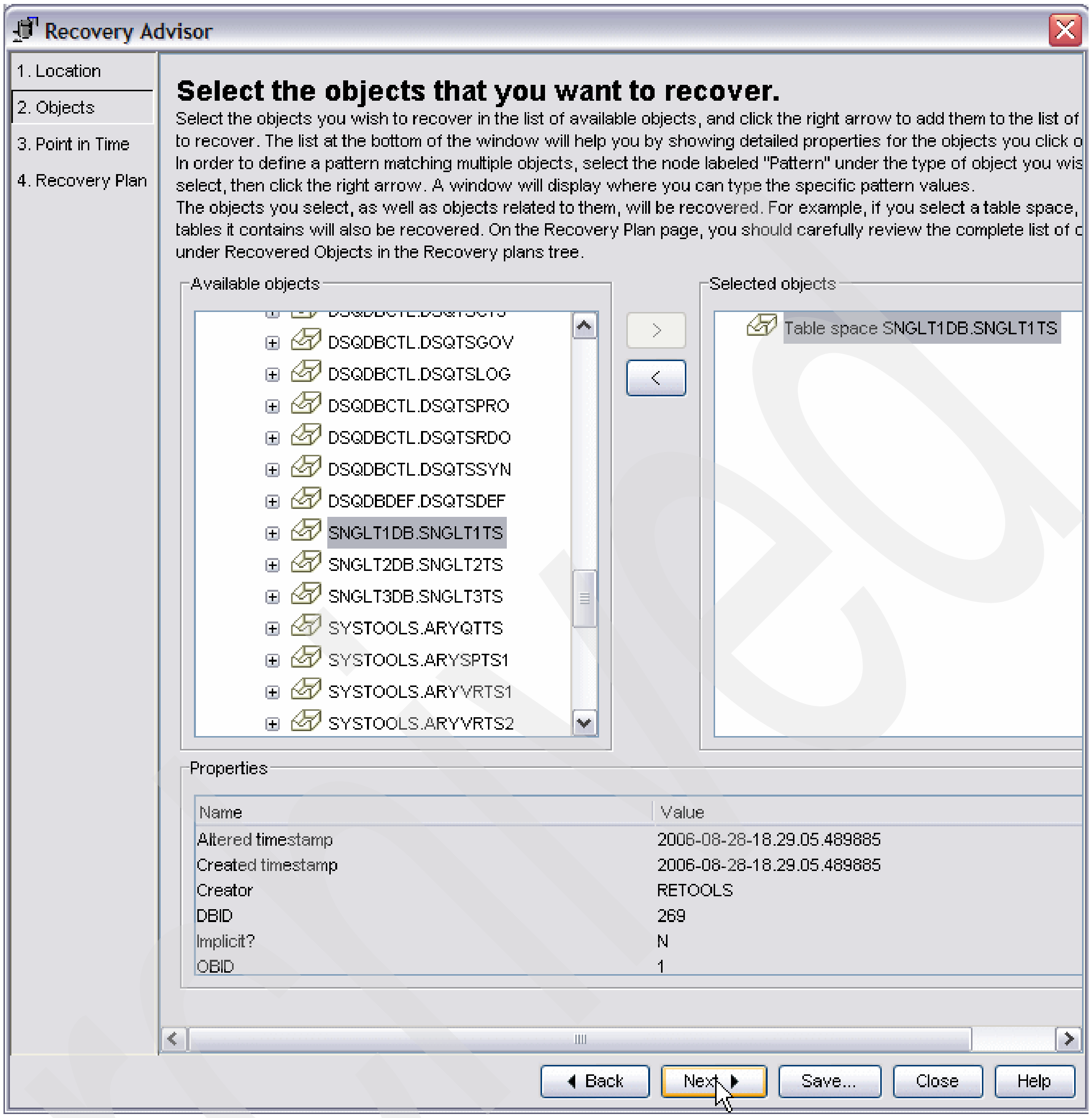

Figure B-50 Selecting the object SINGLT1TS table space to recover (3/3)

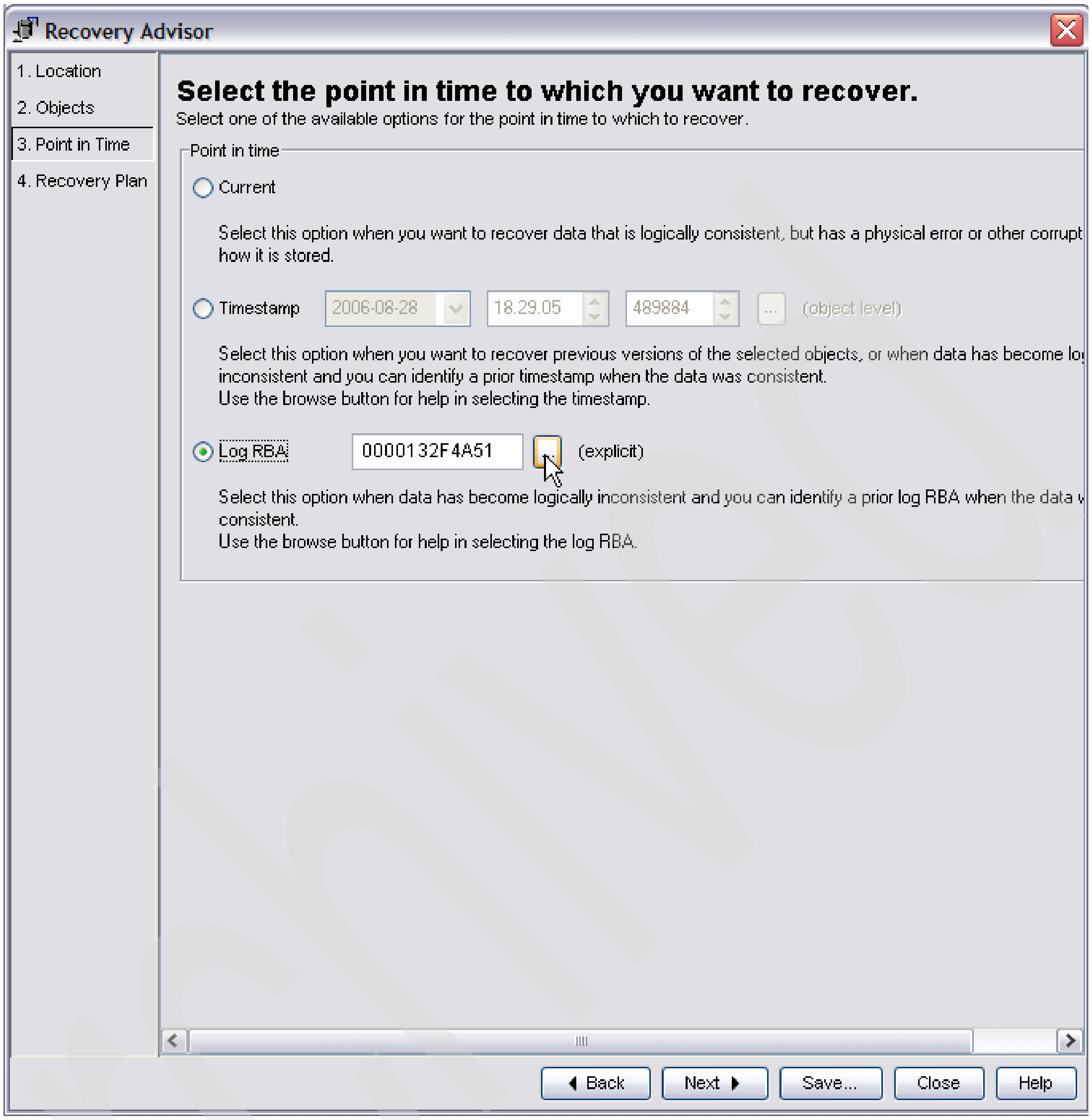

Figure B-51 Selecting the point (RBA/LRSN) to recover to (1/5)

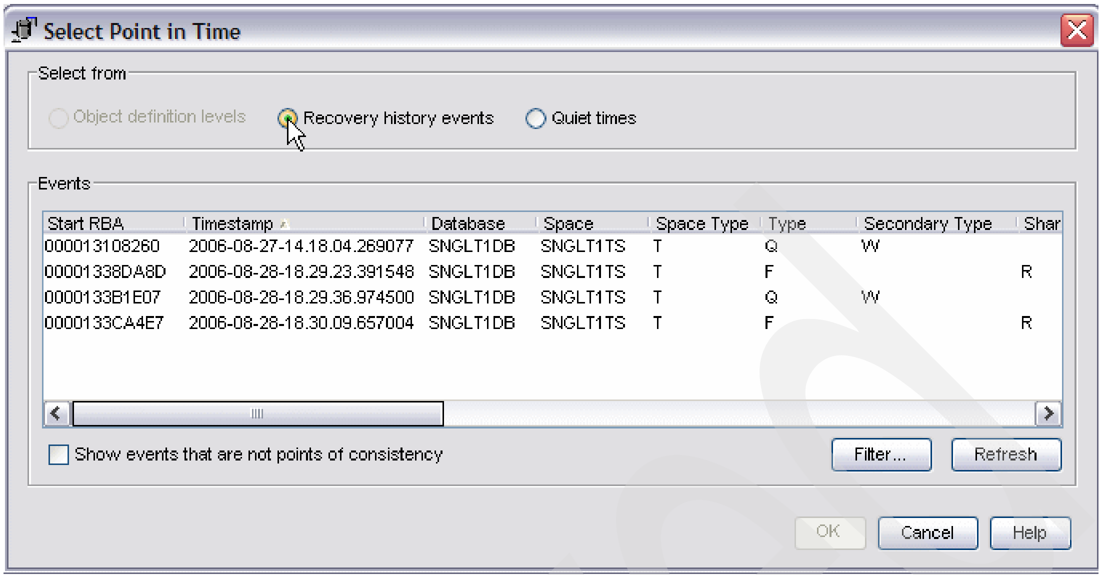

Figure B-52 Selecting the point (RBA/LRSN) to recover to (2/5)

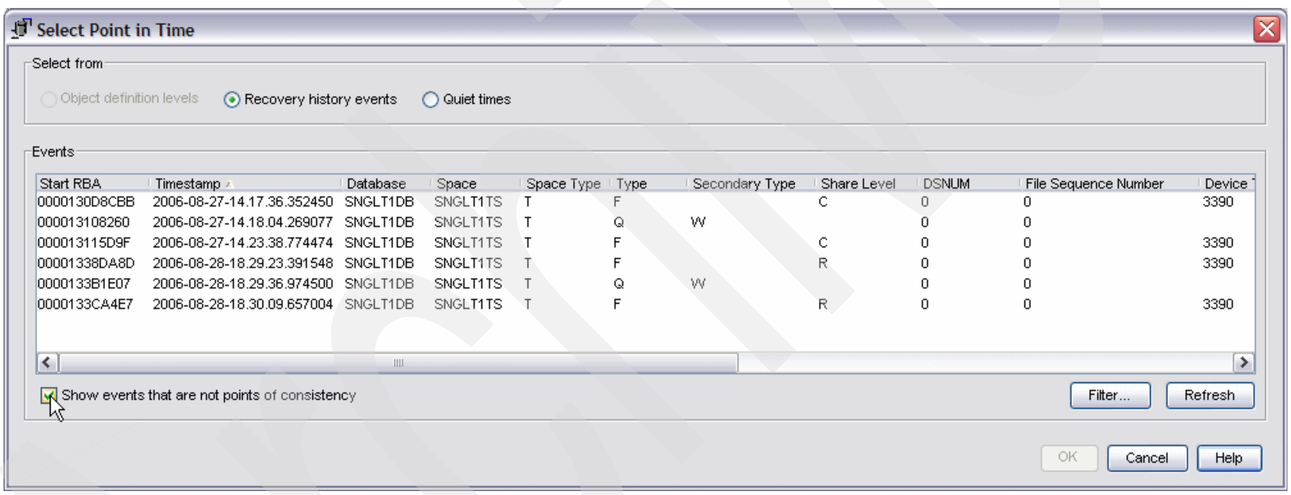

Figure B-53 Selecting the point (RBA/LRSN) to recover to (3/5)

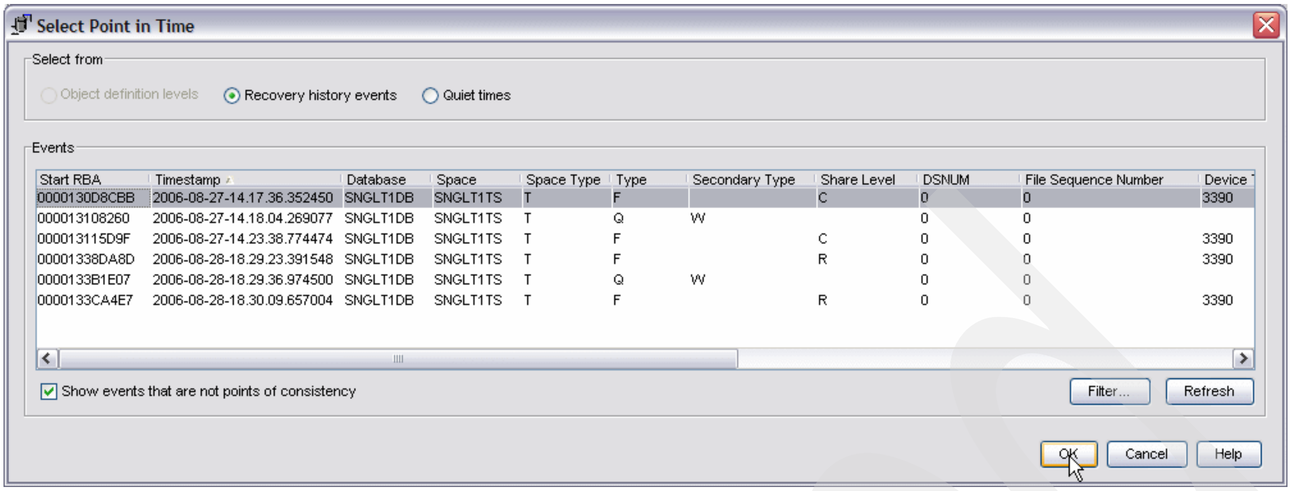

Figure B-54 Selecting the point (RBA/LRSN) to recover to (4/5)

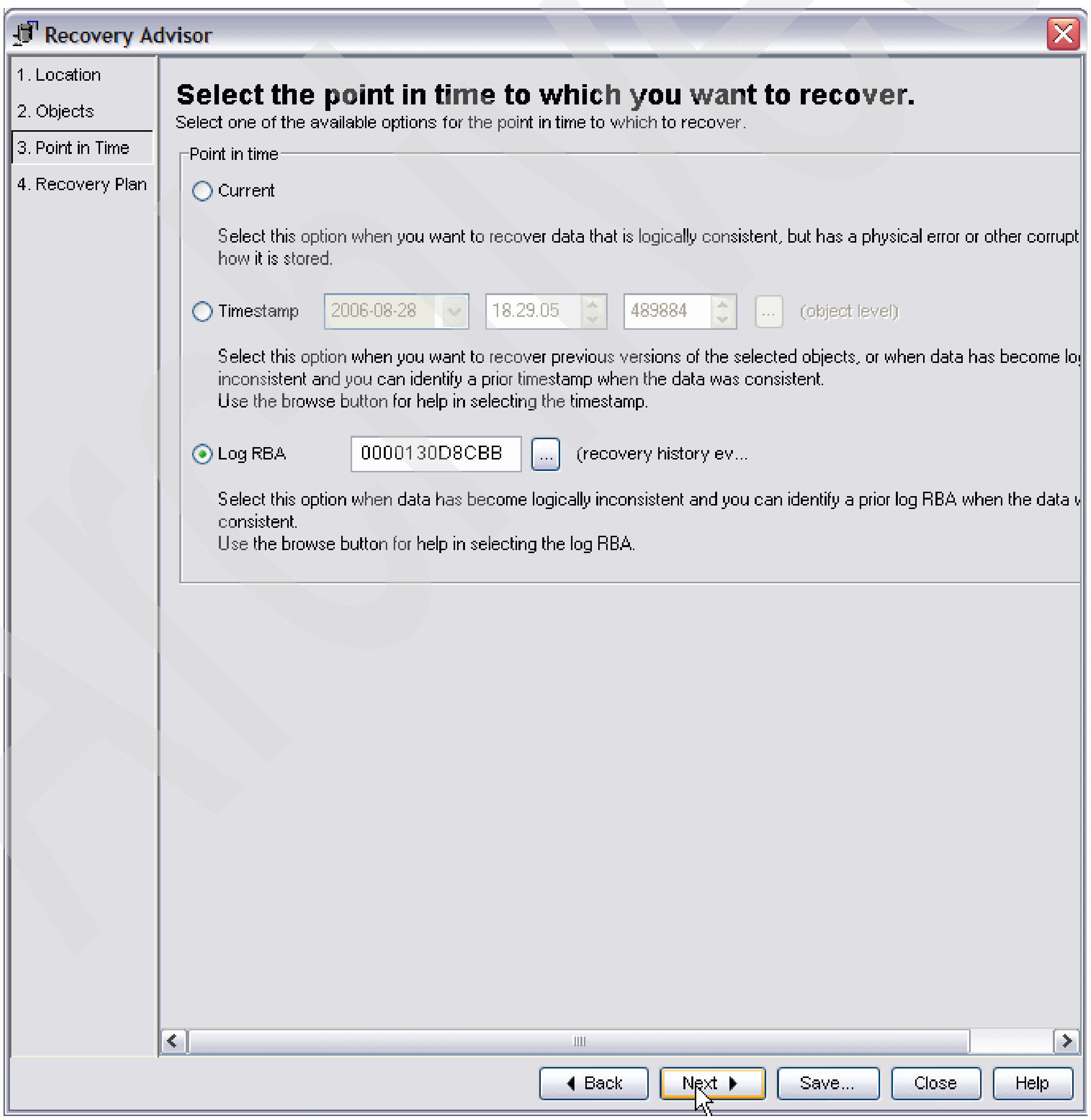

Figure B-55 Selecting the point (RBA/LRSN) to recover to (5/5)

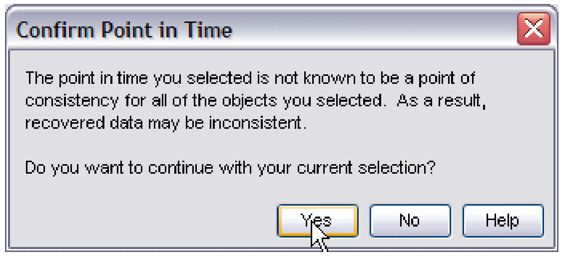

Figure B-56 Warning message that selected RBA might not be a point-of-consistency

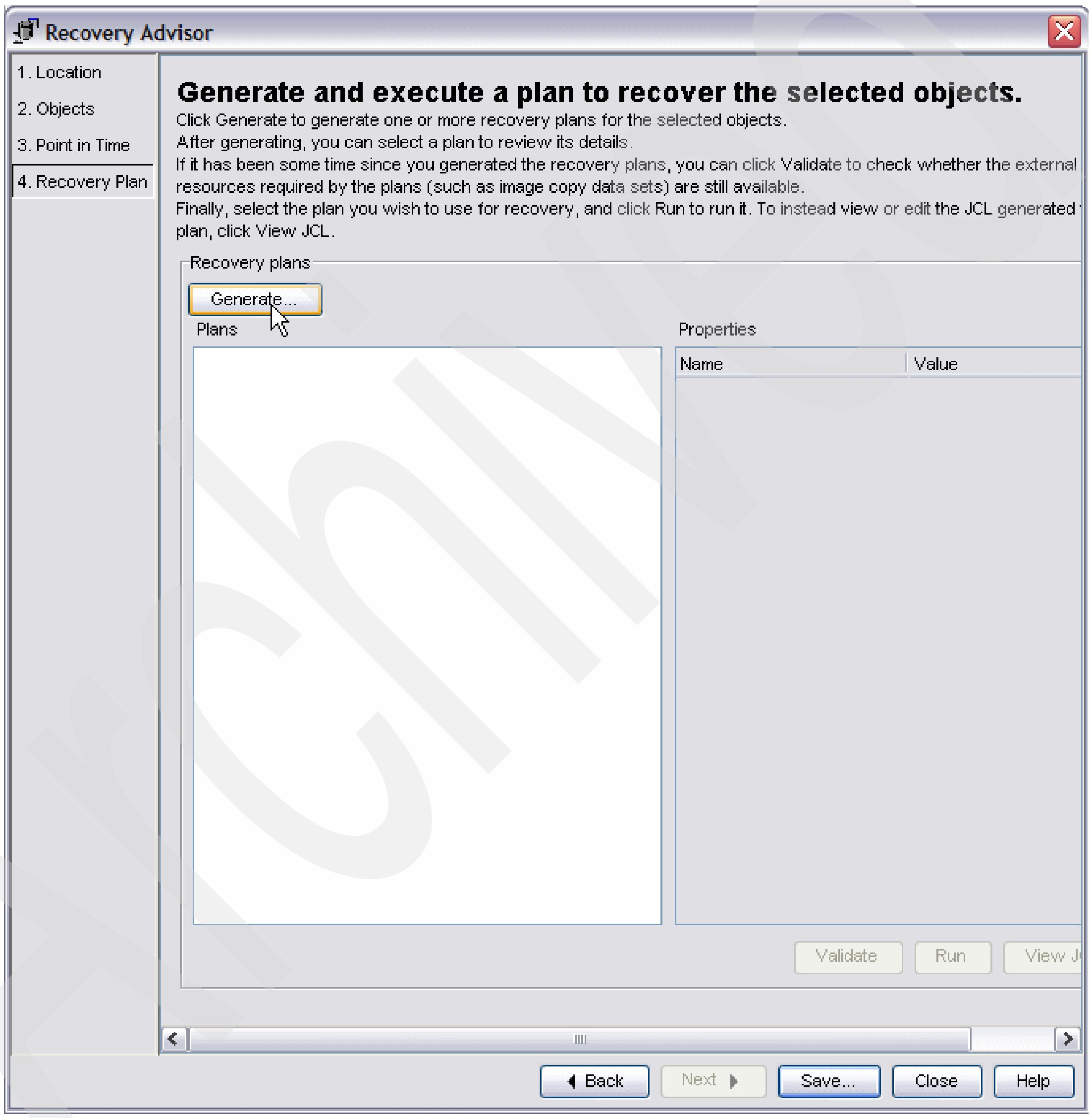

Figure B-57 Generating recovery plans

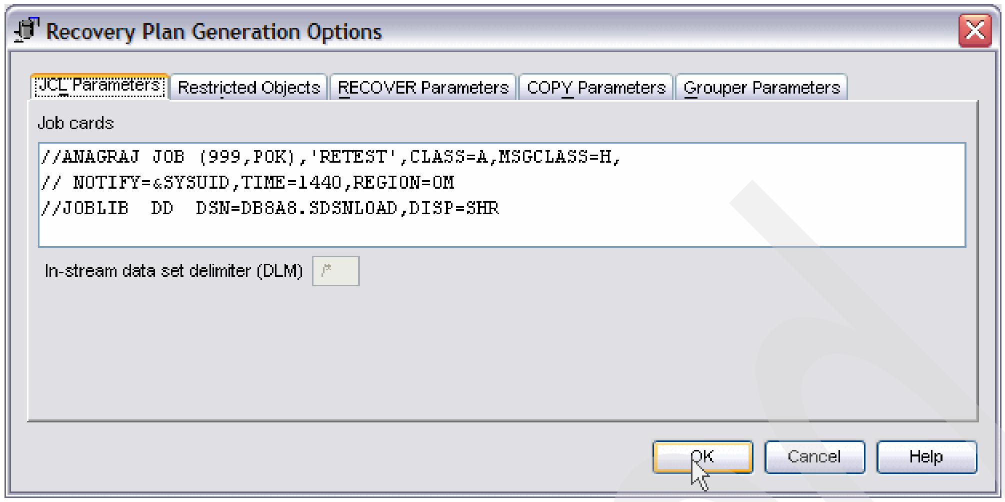

Figure B-58 Recovery plan generation options: JCL parameters

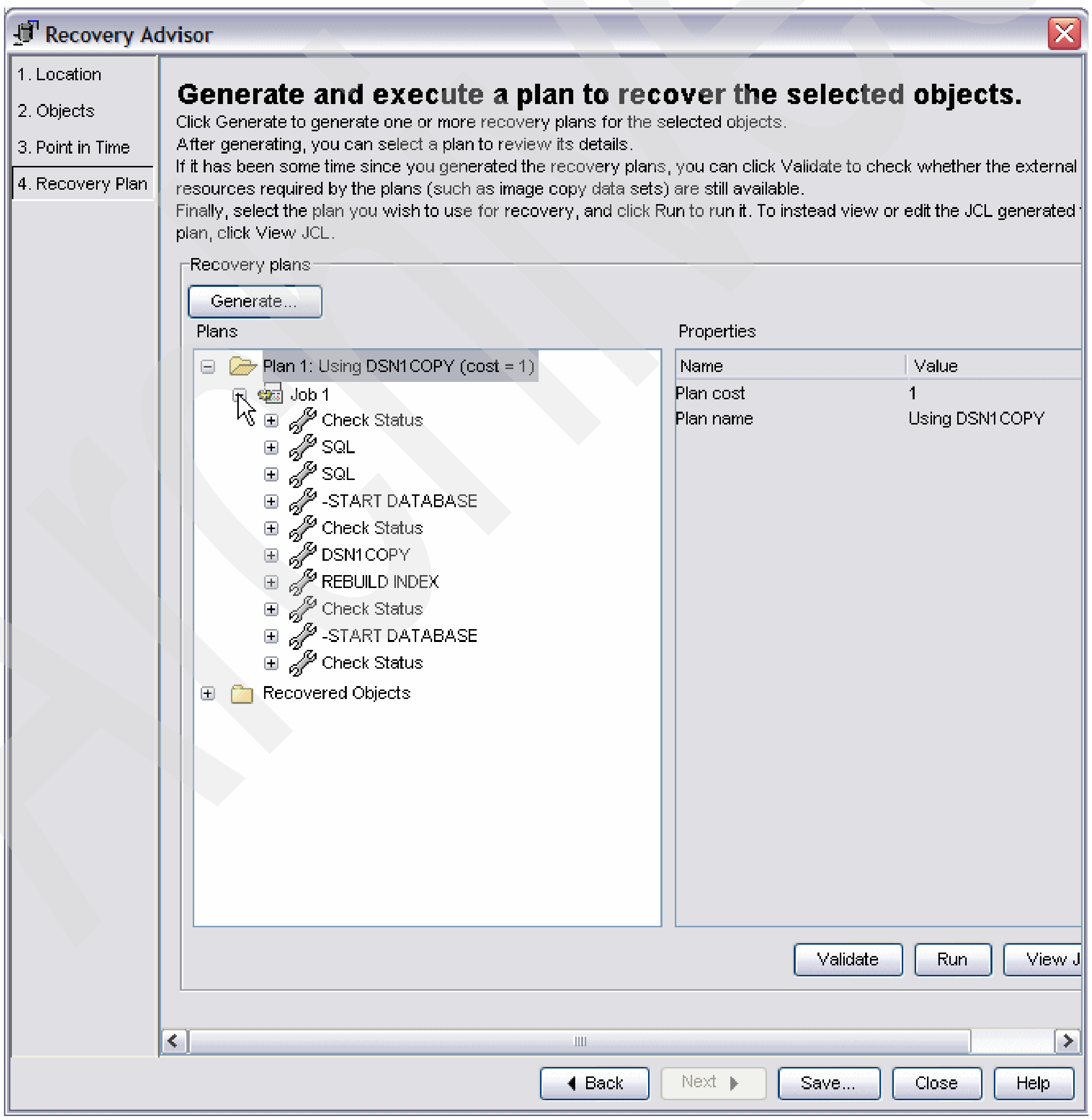

Figure B-59 Generated recovery plans with expanded steps in Plan 1

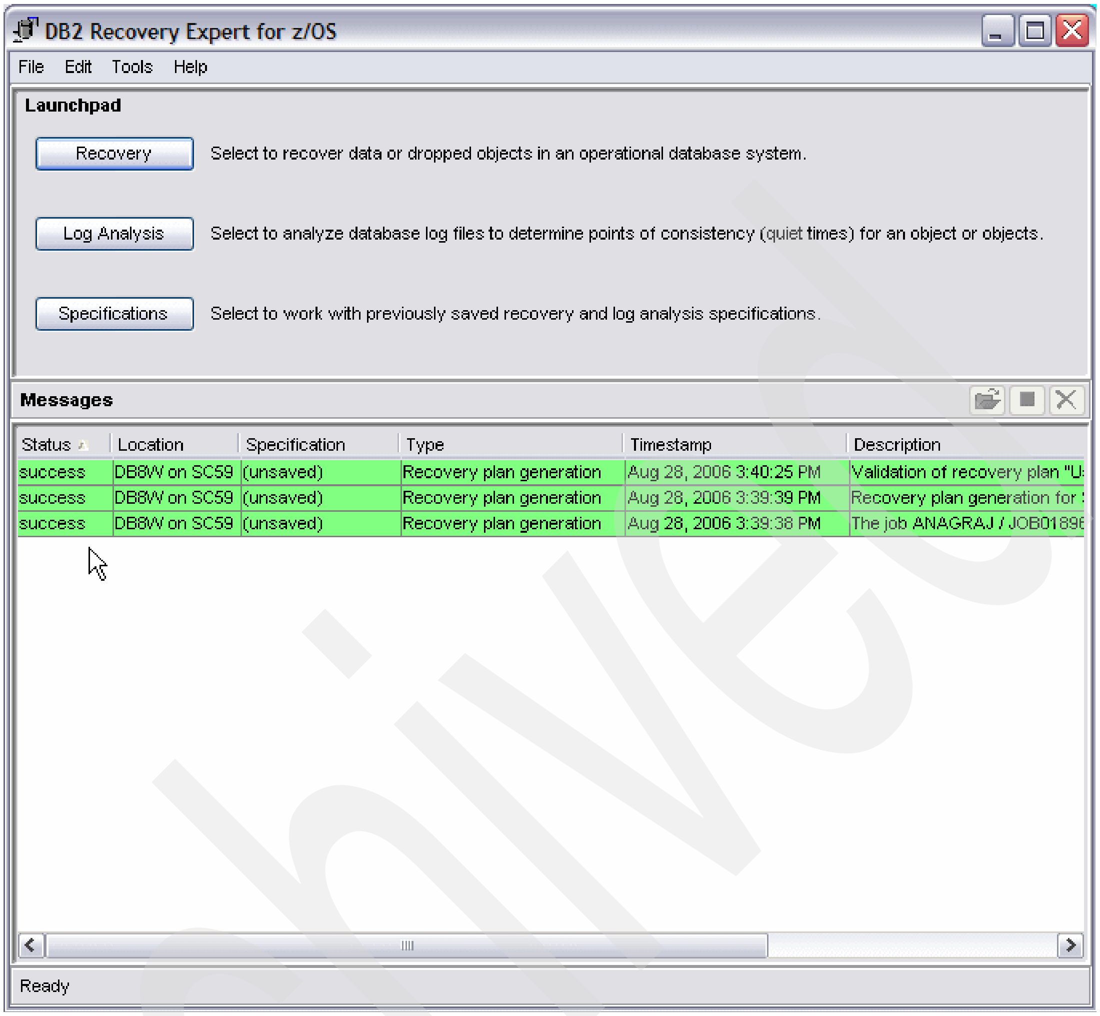

Figure B-60 Status messages in the messages section of the launchpad window

B.6 Log analysis

Log analysis provides the ability to analyze database log files to determine points-of-consistency, or quiet times, for objects. The following windows guide you through a typical log analysis process to identify quiet times for a given object within a specified time interval. To minimize clutter, we do not show all window interactions here.

1. After you log on to the Recovery Expert for z/OS, click **Log Analysis** in the Launchpad window shown in Figure B-61 on page 274. This takes you to the location selection window.
2. In the location selection window in Figure B-62 on page 275, click **D8F1** to view its details. Click **Next** to proceed to specify the range of log data that you want to analyze.
3. In Figure B-63 on page 276, you can specify the log range as an interval preceding the current moment or a starting and ending date and time. Specify the required interval, select the **Use SYSLGRNX**[4] check box, and modify the minimum quiet time to look for (default is 2 minutes), as shown in Figure B-64 on page 277. Click **Next** to select the objects to look for.

[4] You can use the SYSLGRNX option to optimize the log files that have to be read. Do not use this option if errors occur, or if the overhead of using it outweighs the savings it provides.

4. Figure B-65 on page 278 through Figure B-67 on page 280 show the PARTTS table space selected as the object to look for. Click **Next** to initiate the log analysis process.
5. Figure B-68 on page 281 reviews the log analysis job output. Figure B-69 on page 282 shows the successful execution message in the Launchpad window.
6. The log analysis records any quiet times detected in the SLR tables, which you can view from the Recovery Expert for z/OS GUI client, as shown in windows Figure B-70 on page 283 through Figure B-80 on page 291.
7. Figure B-70 on page 283 through Figure B-73 on page 286 describe the selection of the location and the object PARTTS to recover. In Figure B-74 on page 287, select the **Timestamp** radio button and click the ellipsis to proceed to Figure B-75 on page 288.
8. Select the **Quiet times** radio button to view the quiet times recorded in the SLR, as shown in Figure B-76 on page 289. Highlight the only quiet time shown and click **OK** to view the quiet time selected as the timestamp to recover to, as shown in Figure B-80 on page 291. The rest of the process to generate recovery plans and run a selected recovery plan is not shown here.

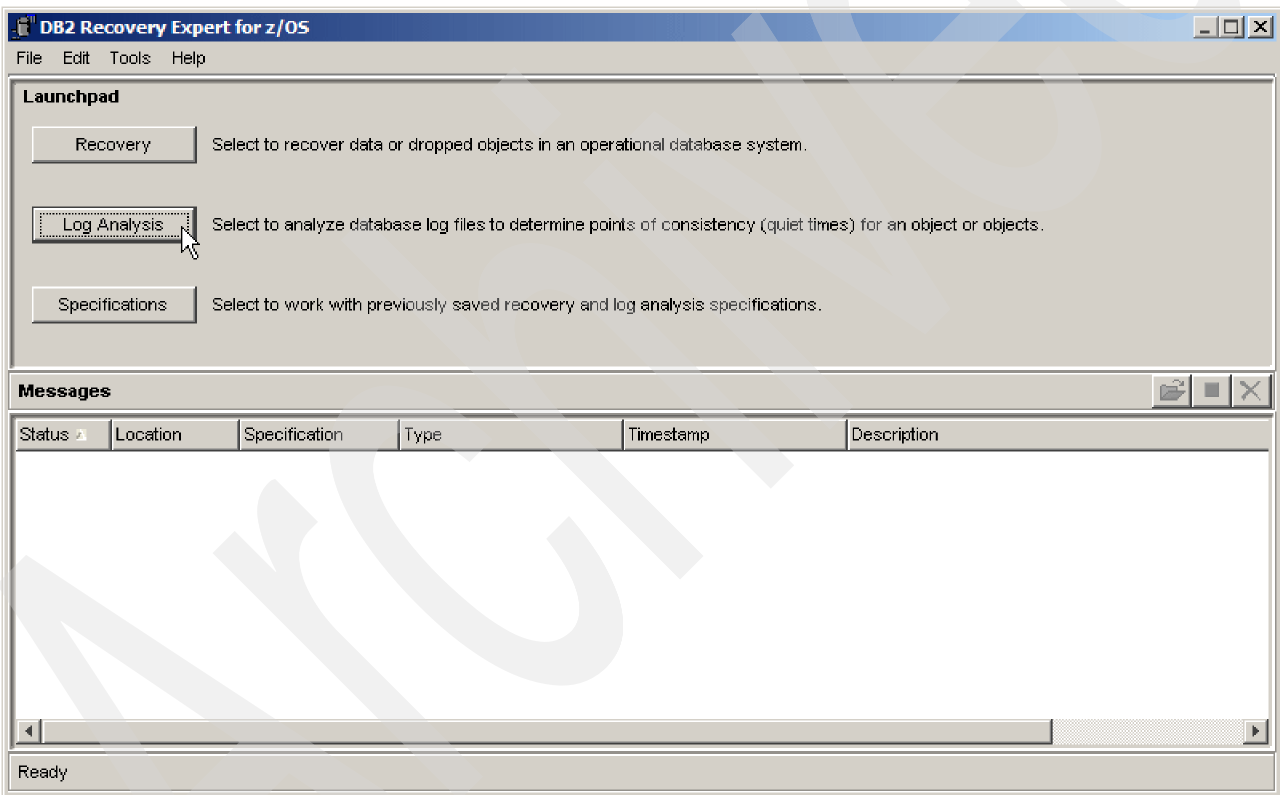

Figure B-61 Launchpad

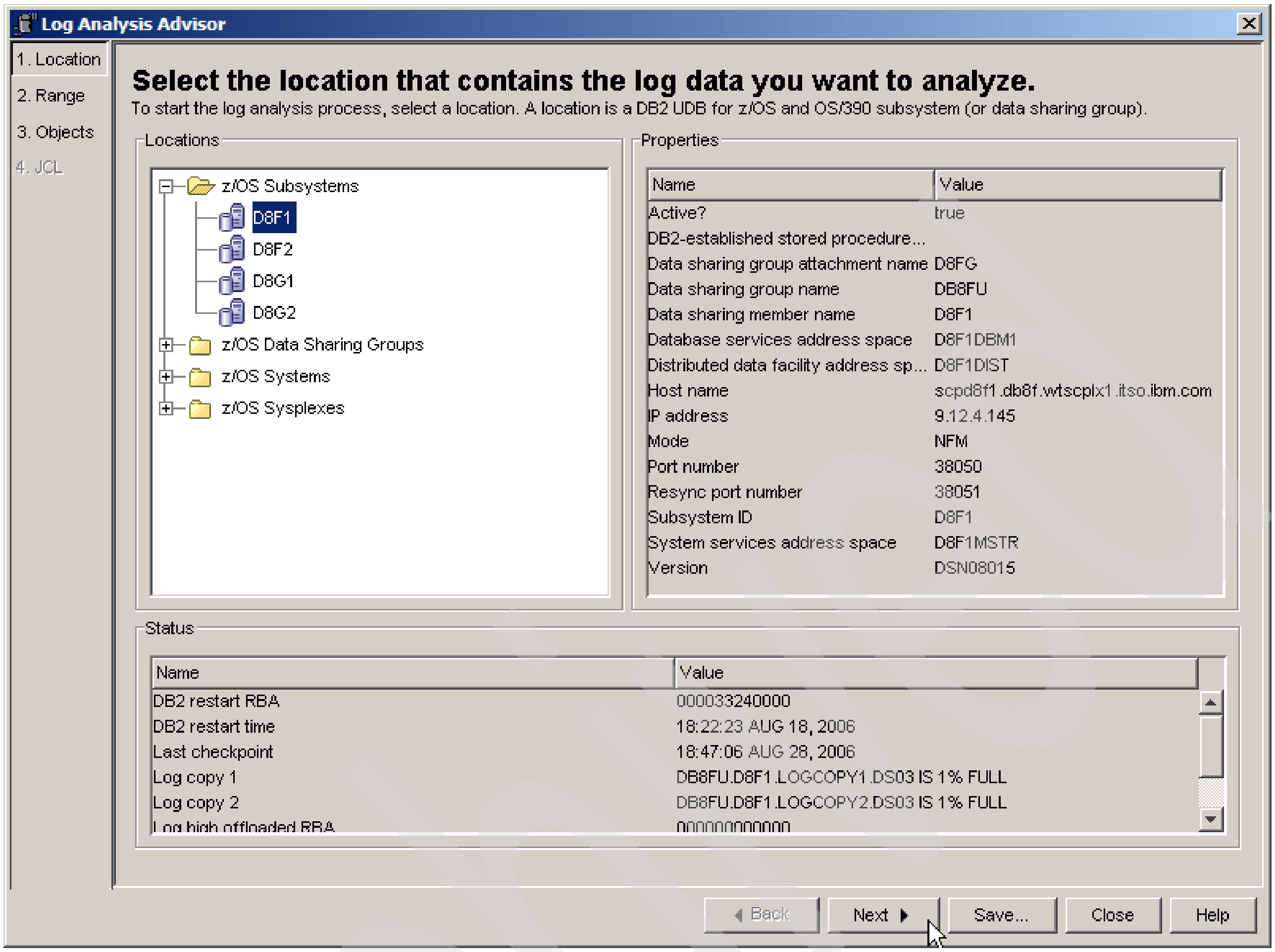

Figure B-62 Selecting the location that contains the log data you want to analyze

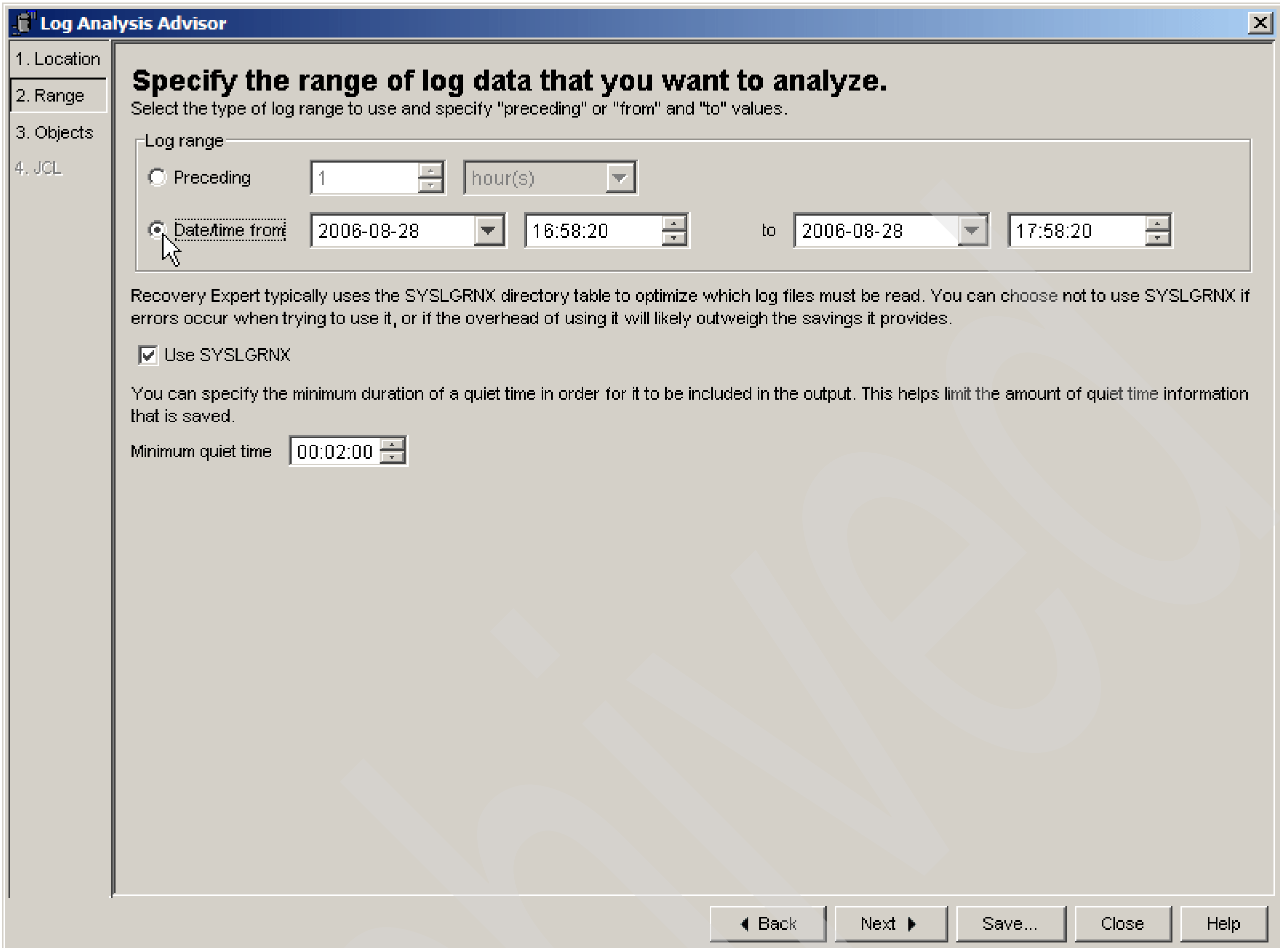

Figure B-63 Specifying the range of log data you want to analyze (1/2)

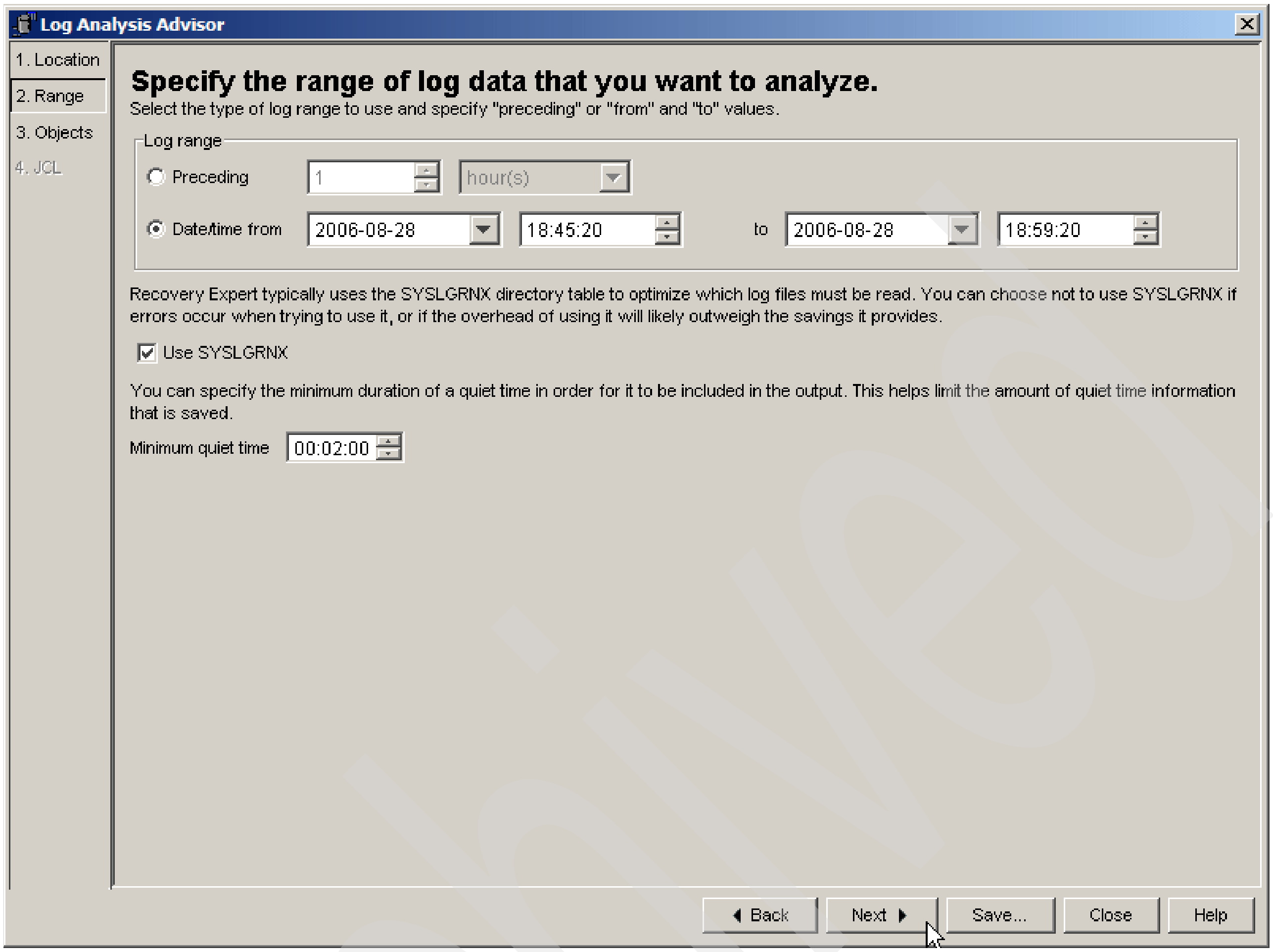

Figure B-64 Specifying the range of log data you want to analyze (2/2)

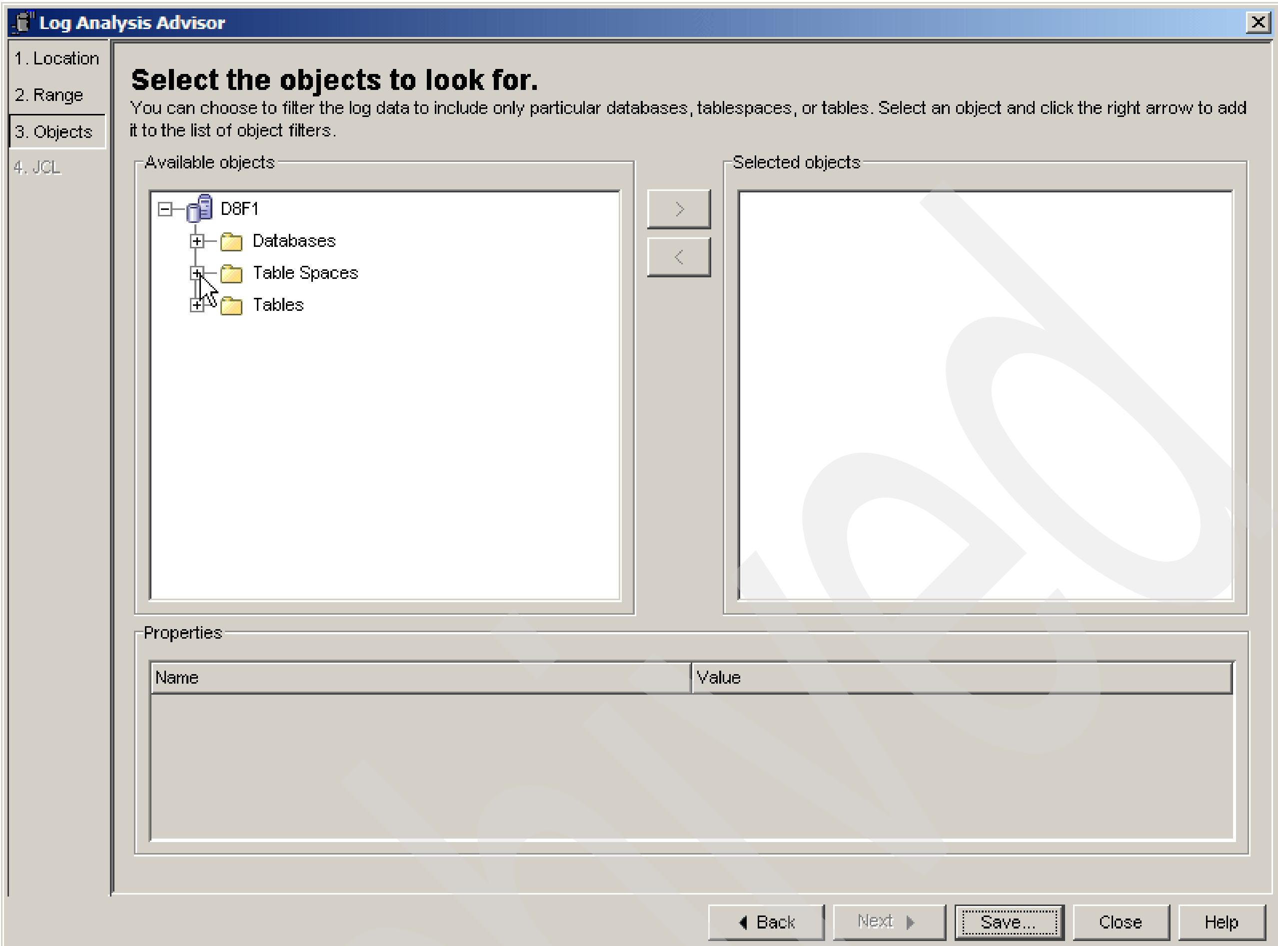

Figure B-65 Selecting the objects to look for (1/3)

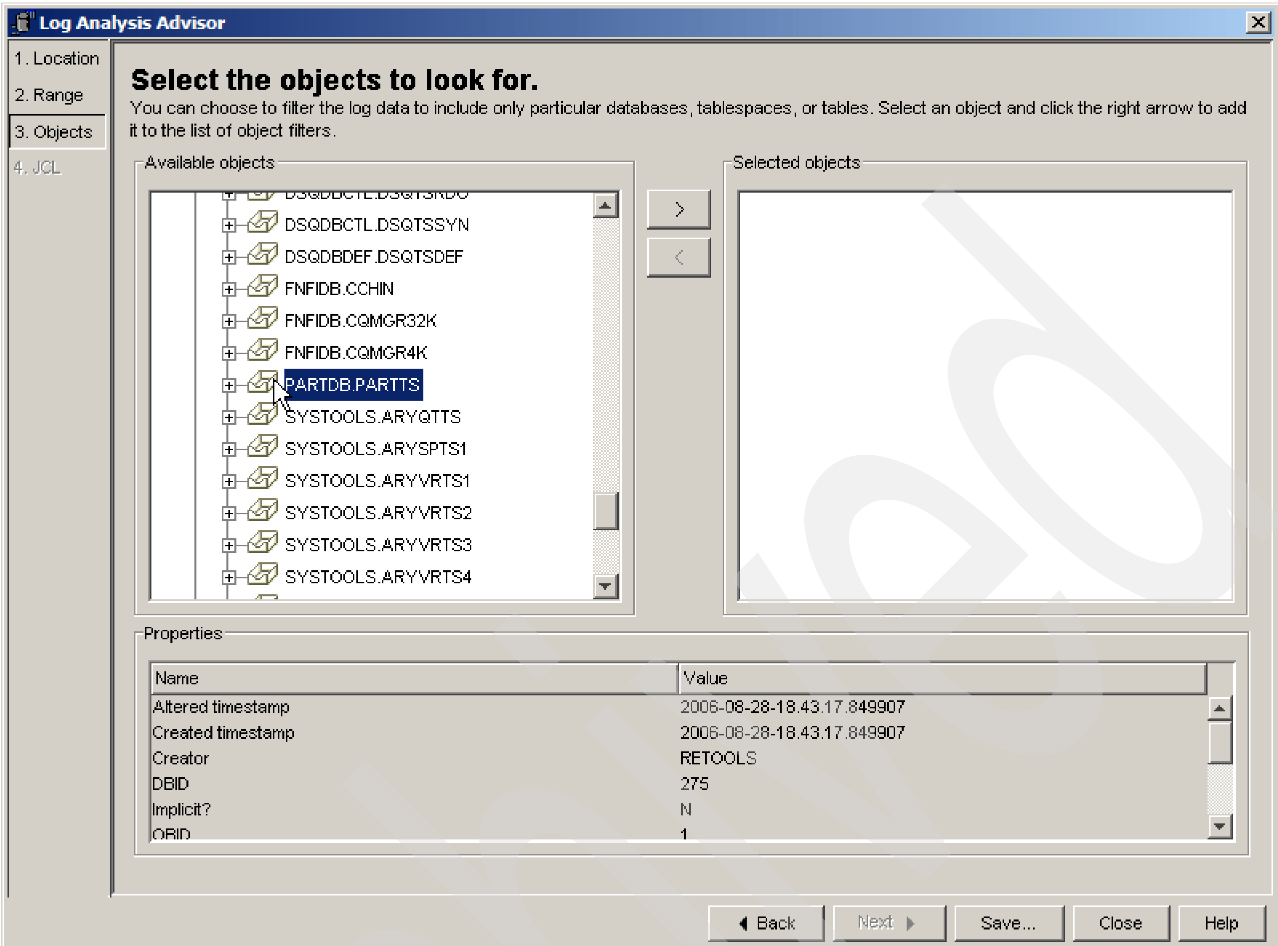

Figure B-66 Selecting the objects to look for (2/3)

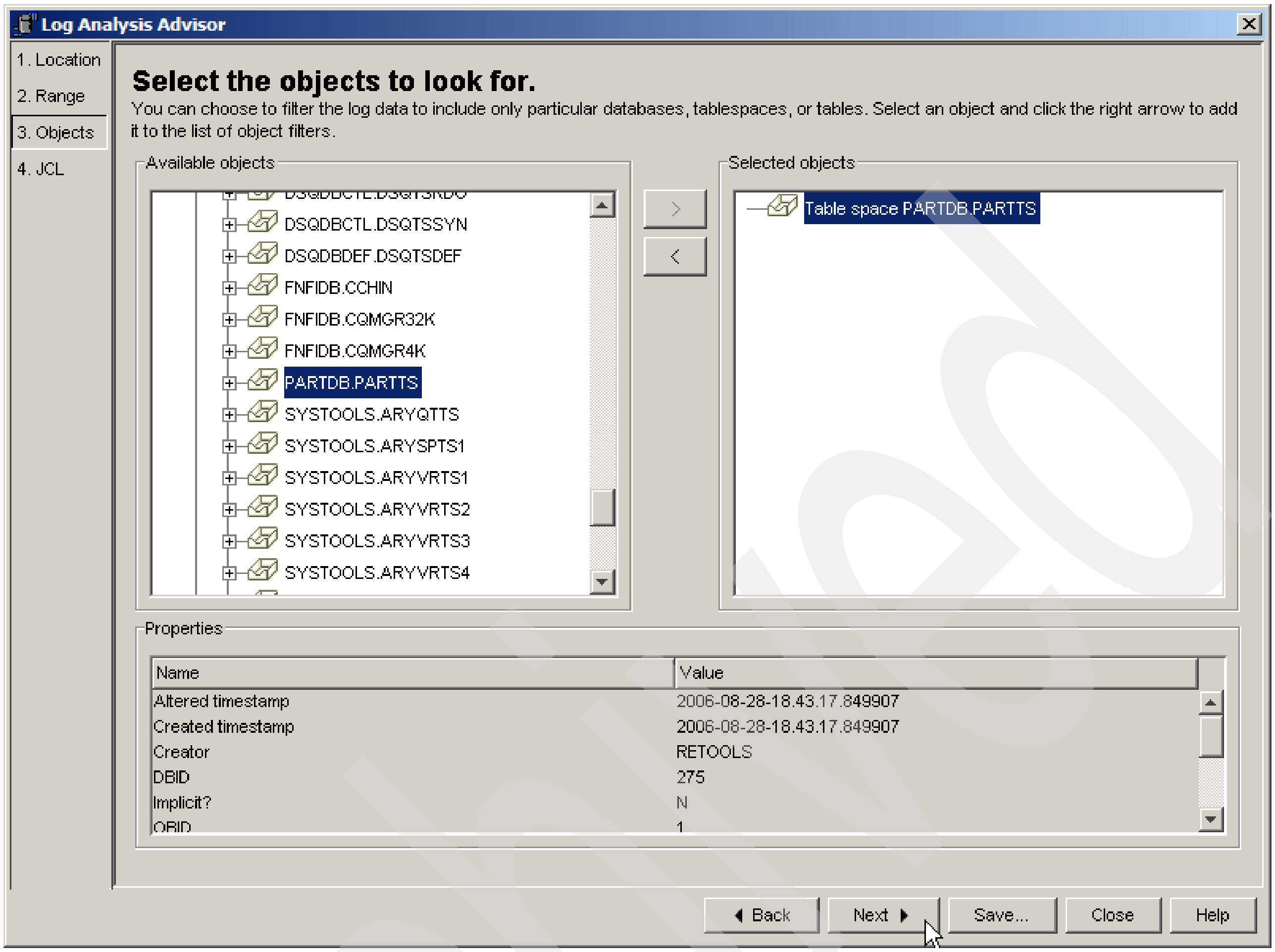

Figure B-67 Selecting the objects to look for (3/3)

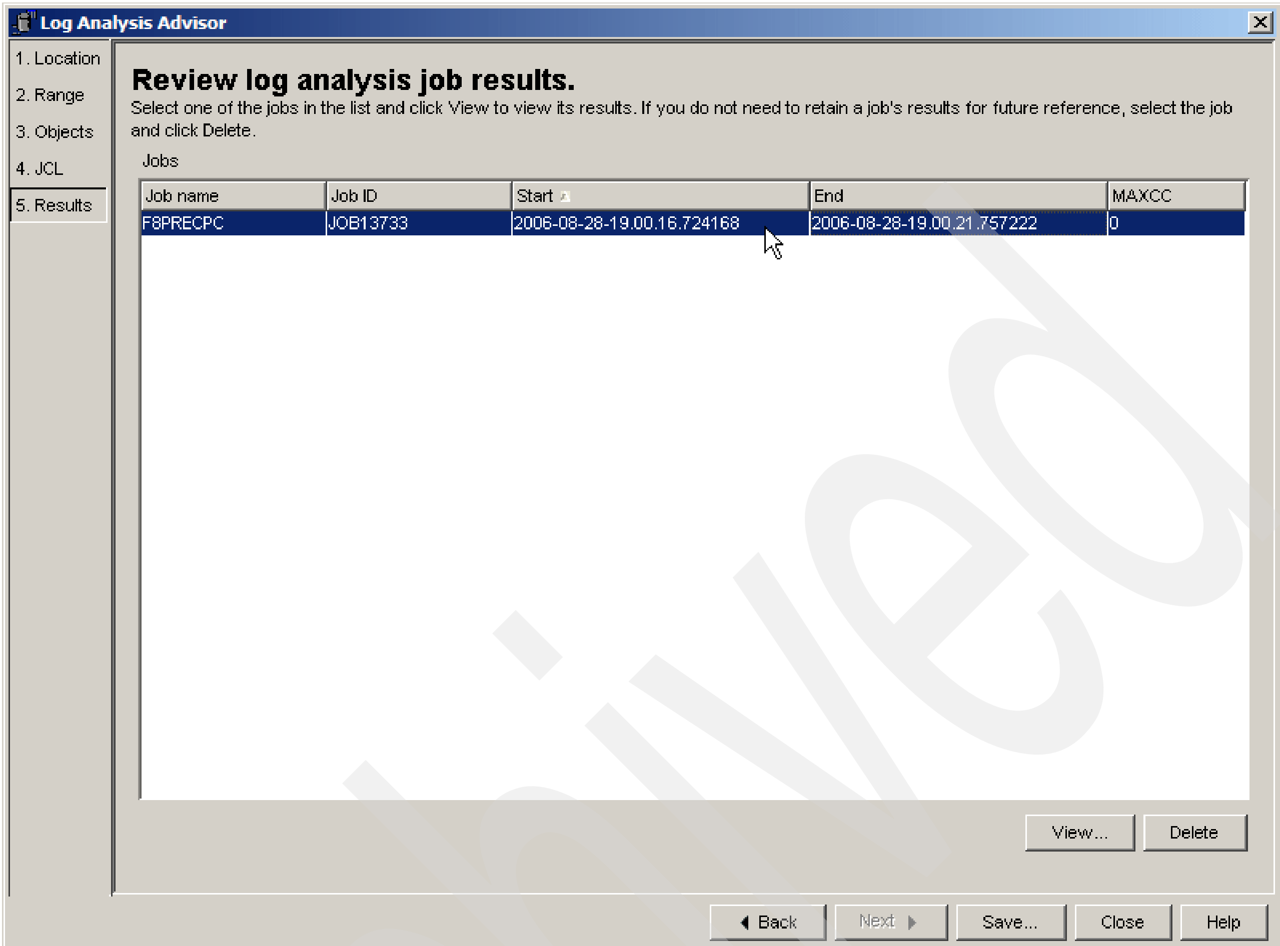

Figure B-68 Reviewing the log analysis job results

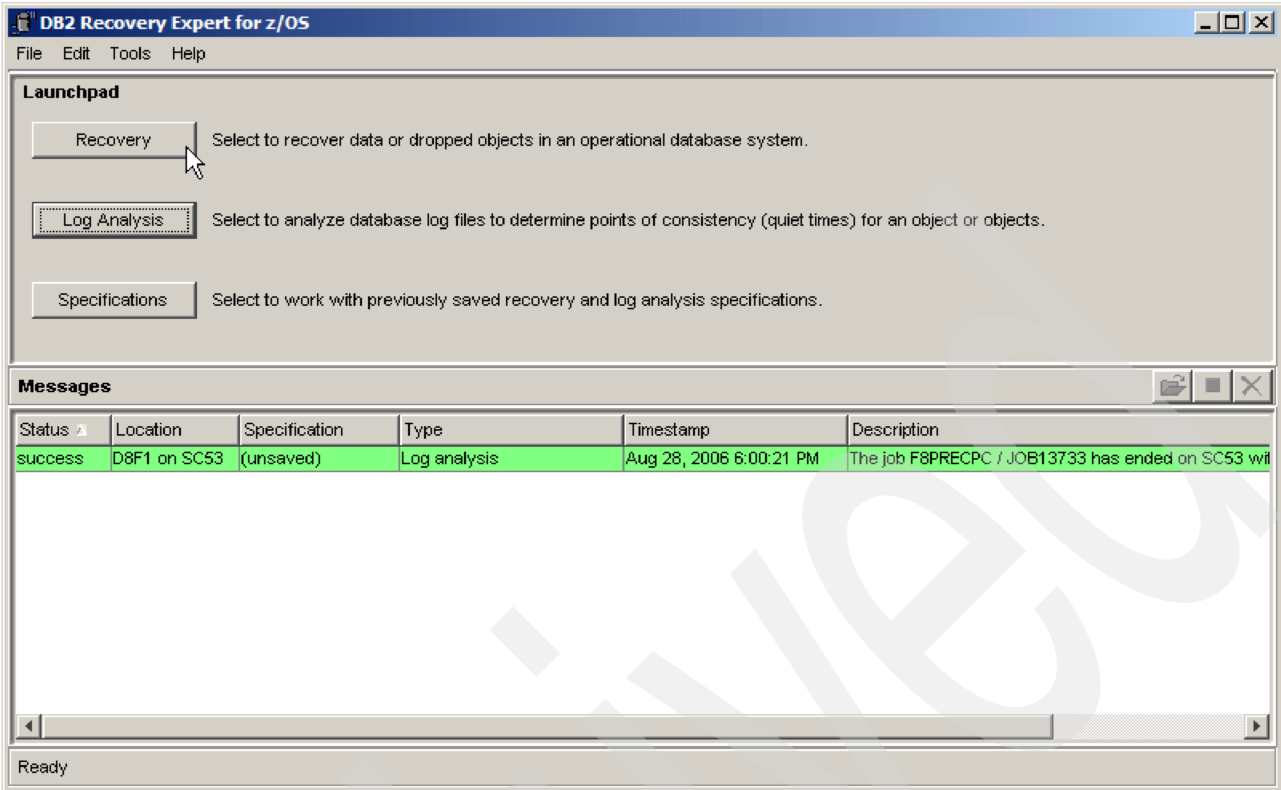

Figure B-69 Successful execution message in the messages section of the launchpad window

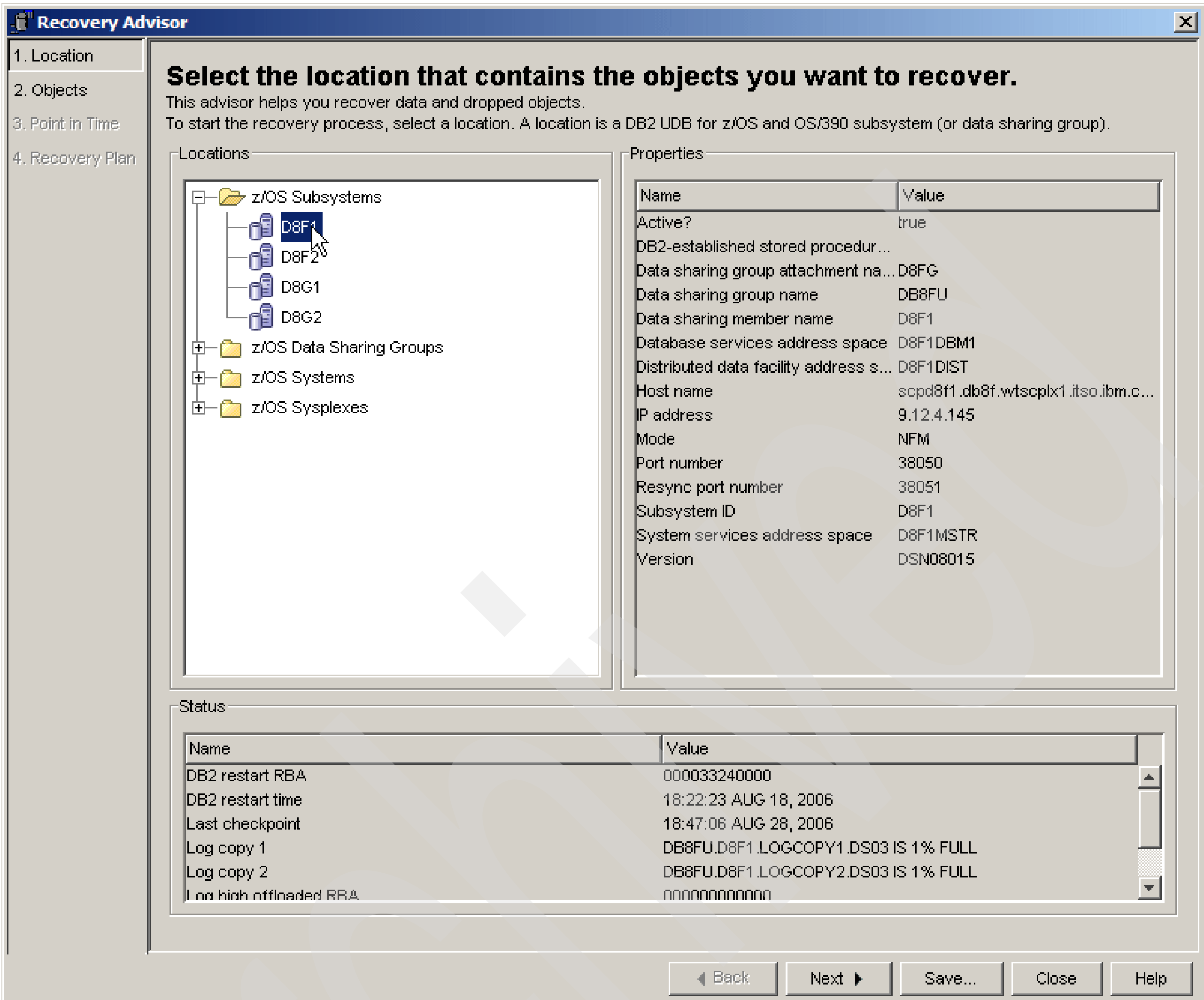

Figure B-70 Selecting the location D8F1 of the objects you want to recover (1/4)

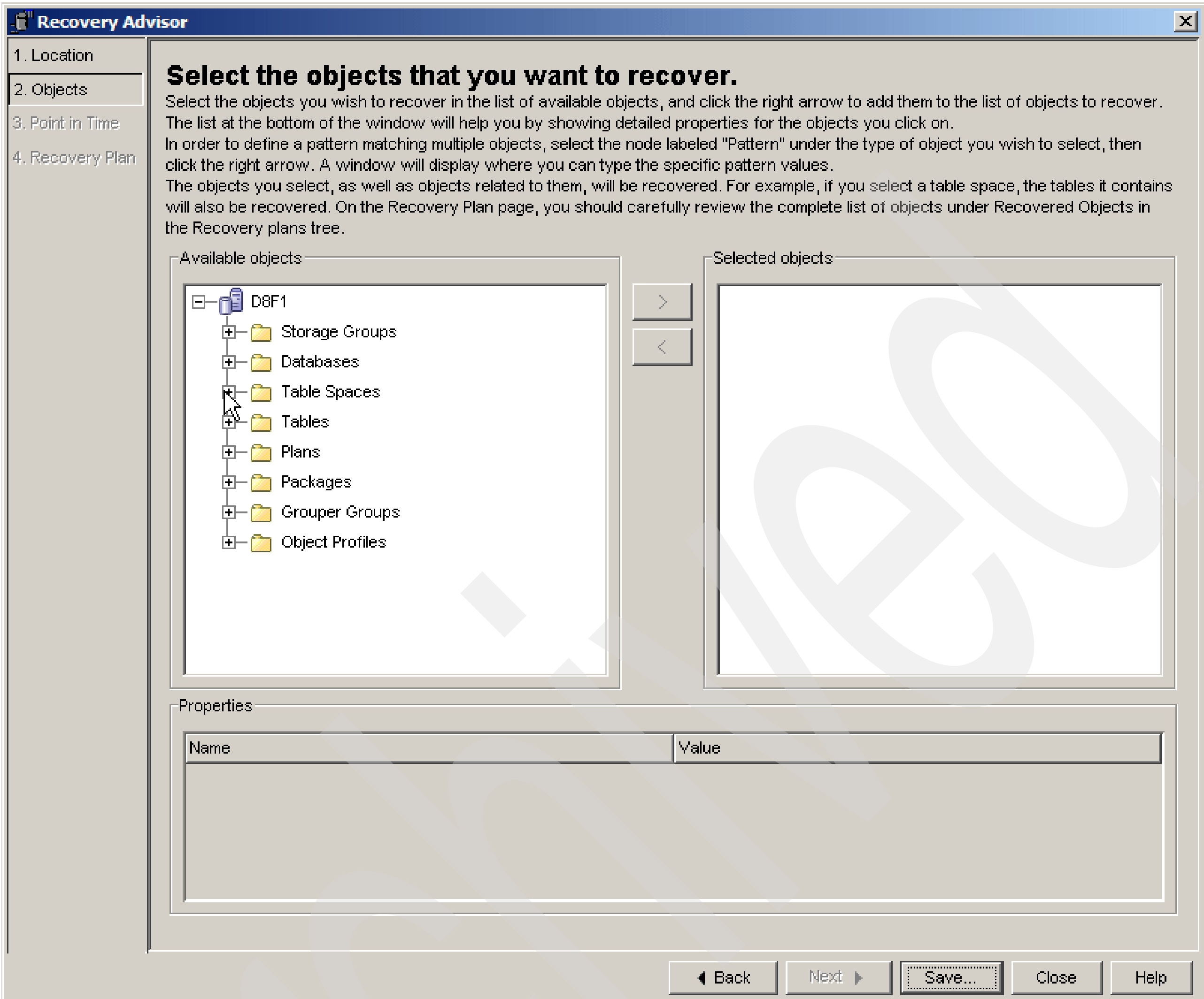

Figure B-71 Selecting the objects PARTTS you want to recover (2/4)

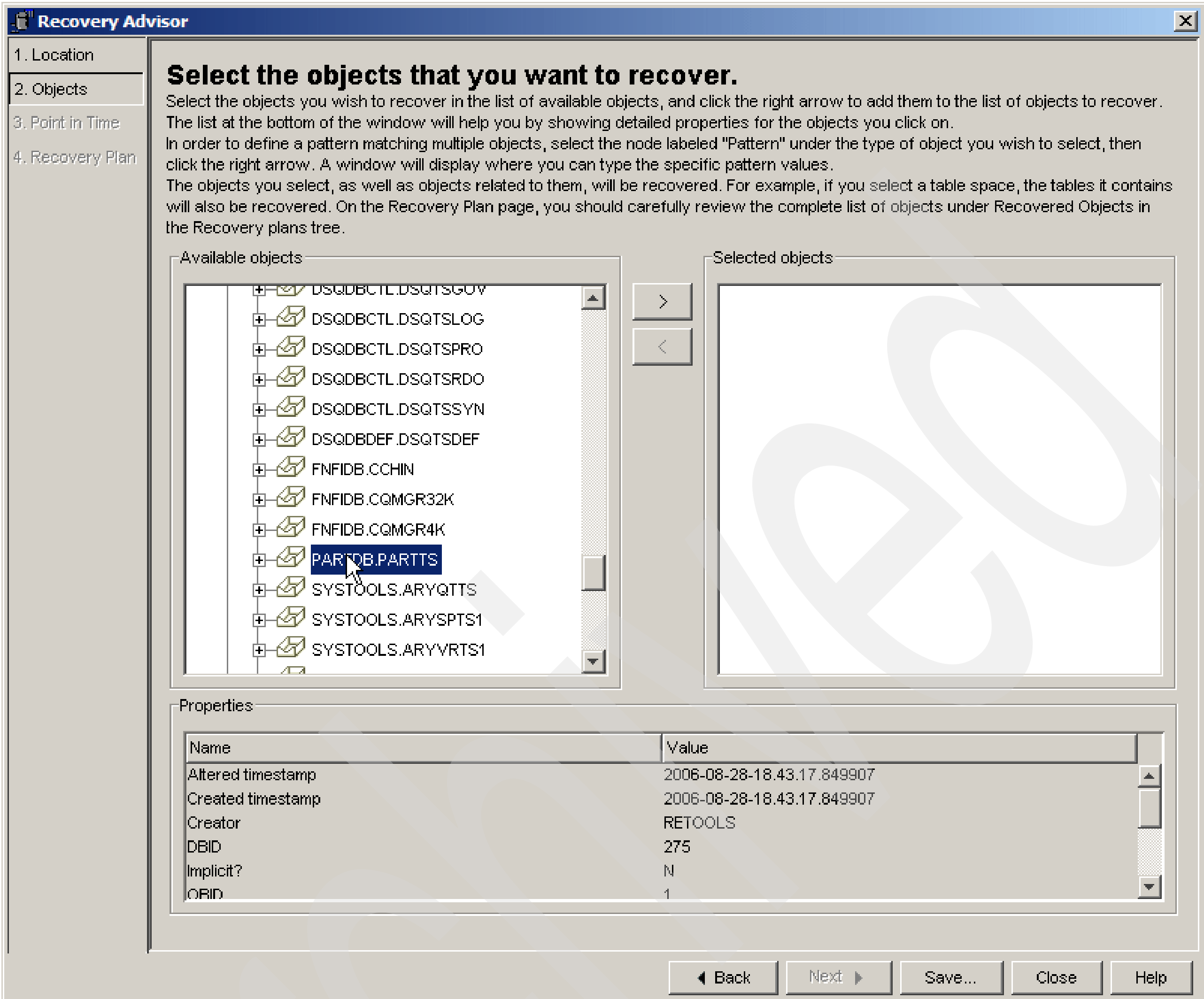

Figure B-72 Selecting the objects PARTTS you want to recover (3/4)

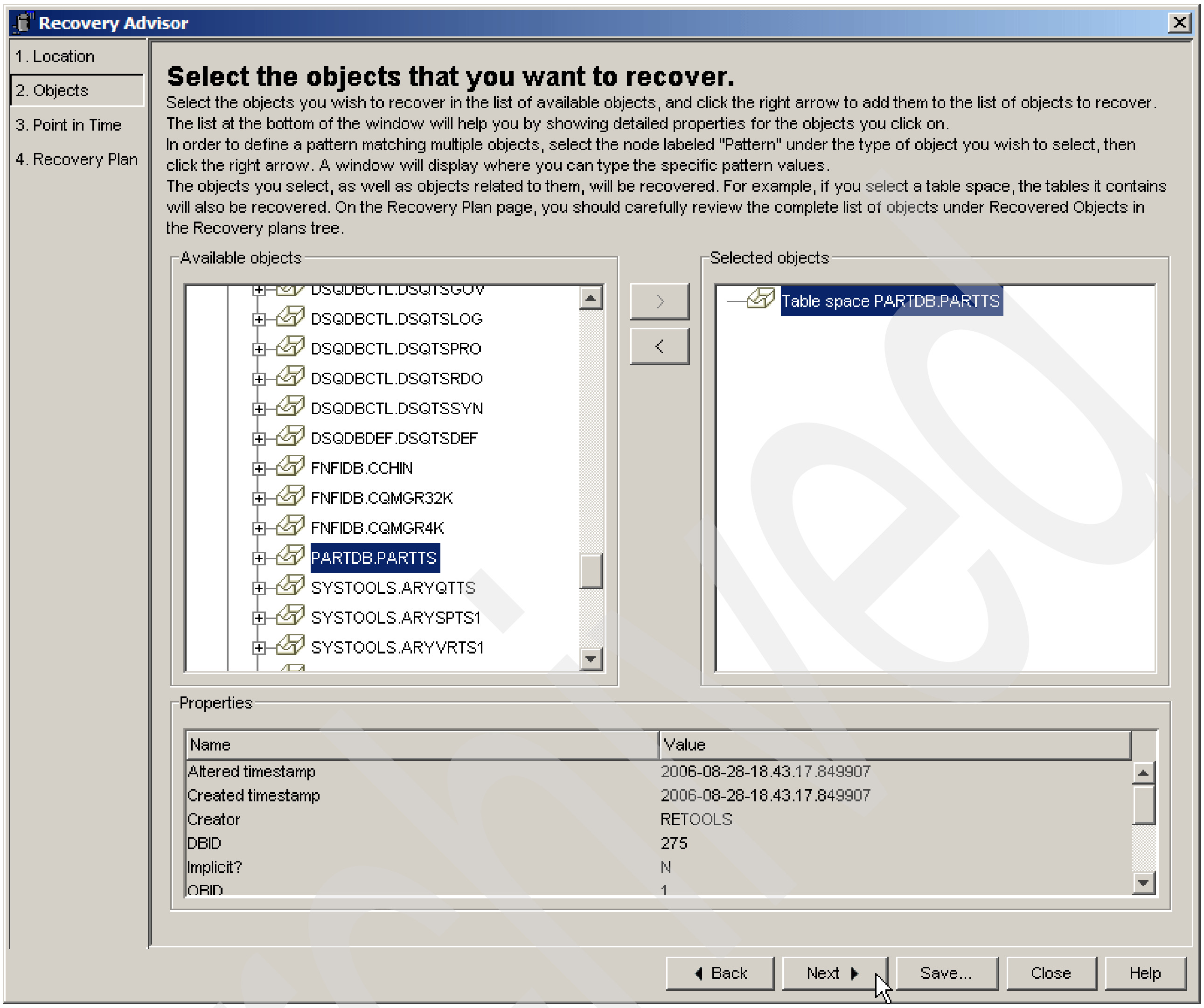

Figure B-73 Selecting the objects PARTTS you want to recover (4/4)

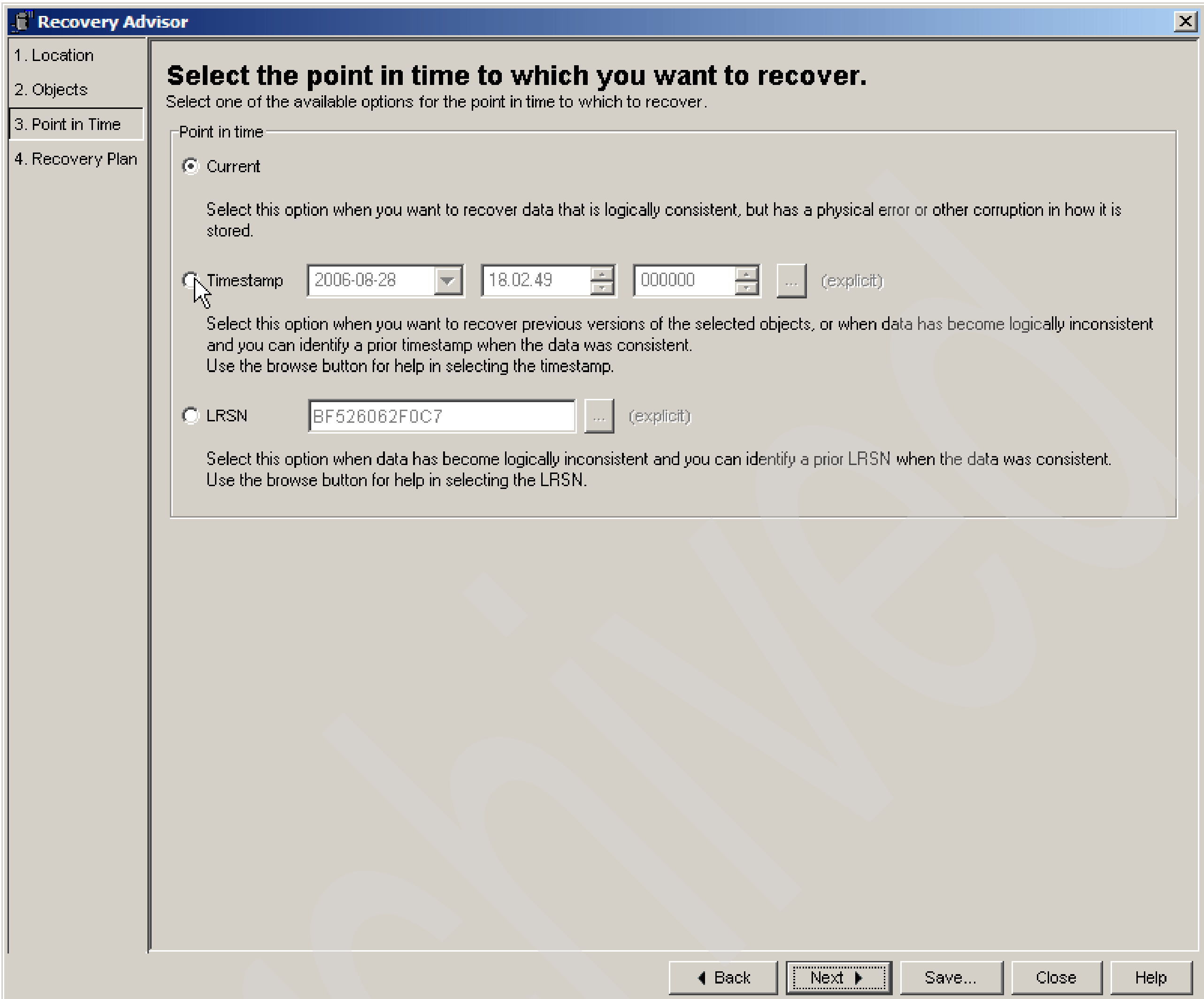

Figure B-74 Selecting the quiet time point to recover to (1/7)

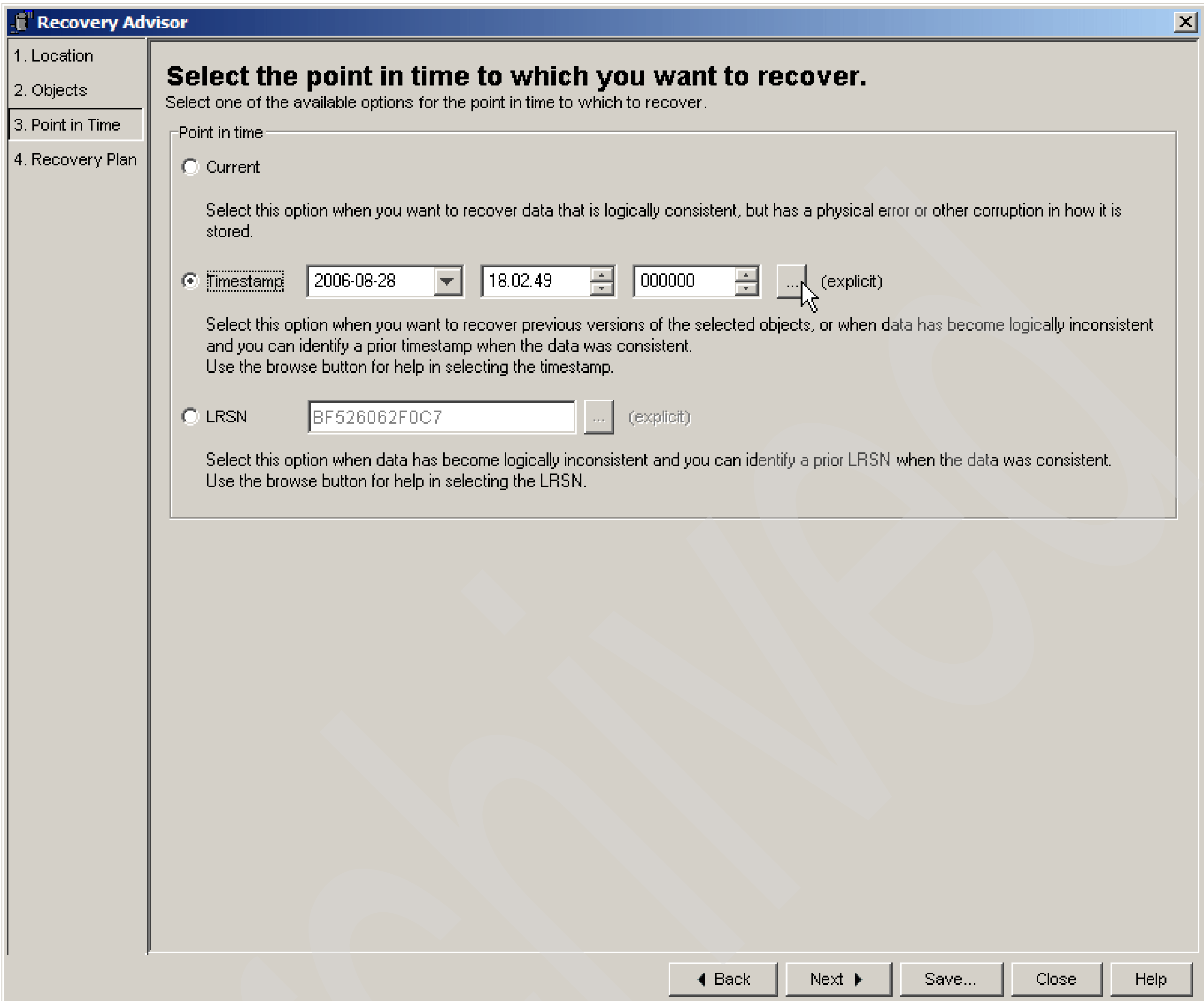

Figure B-75 Selecting the quiet time point to recover to (2/7)

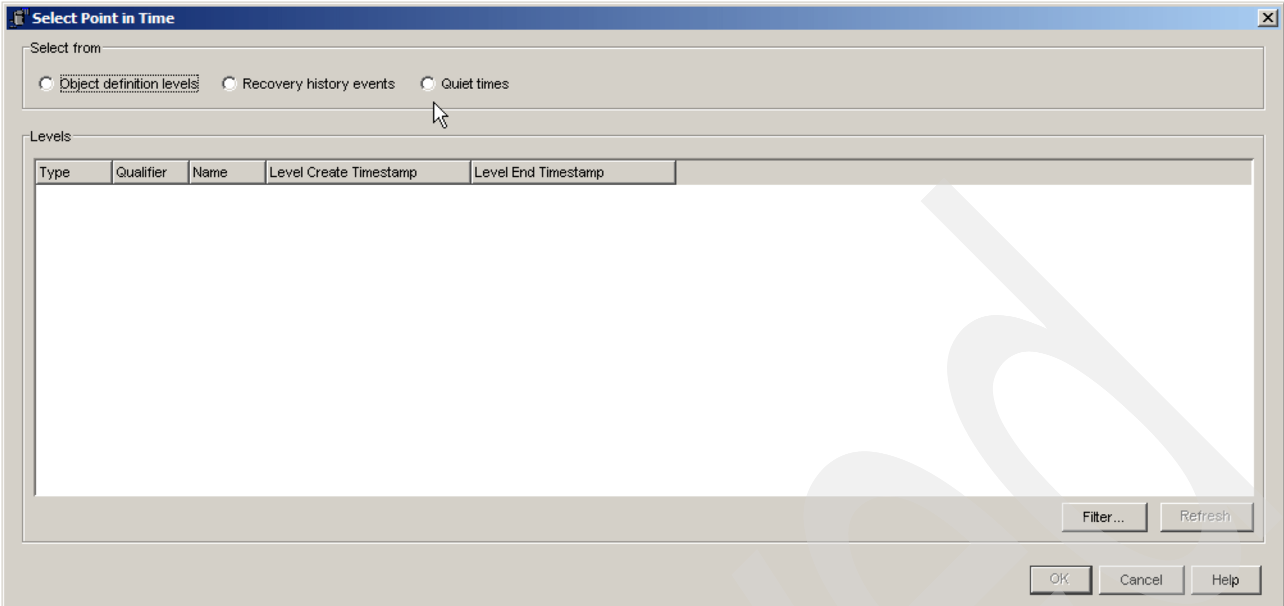

Figure B-76 Selecting the quiet time point to recover to (3/7)

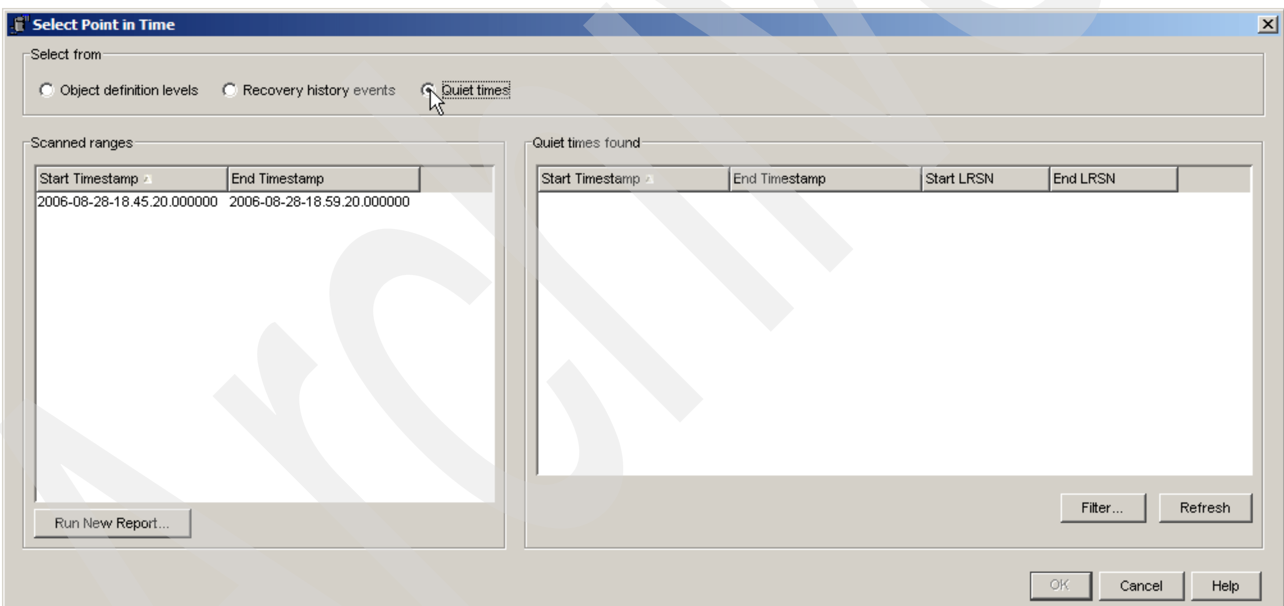

Figure B-77 Selecting the quiet time point to recover to (4/7)

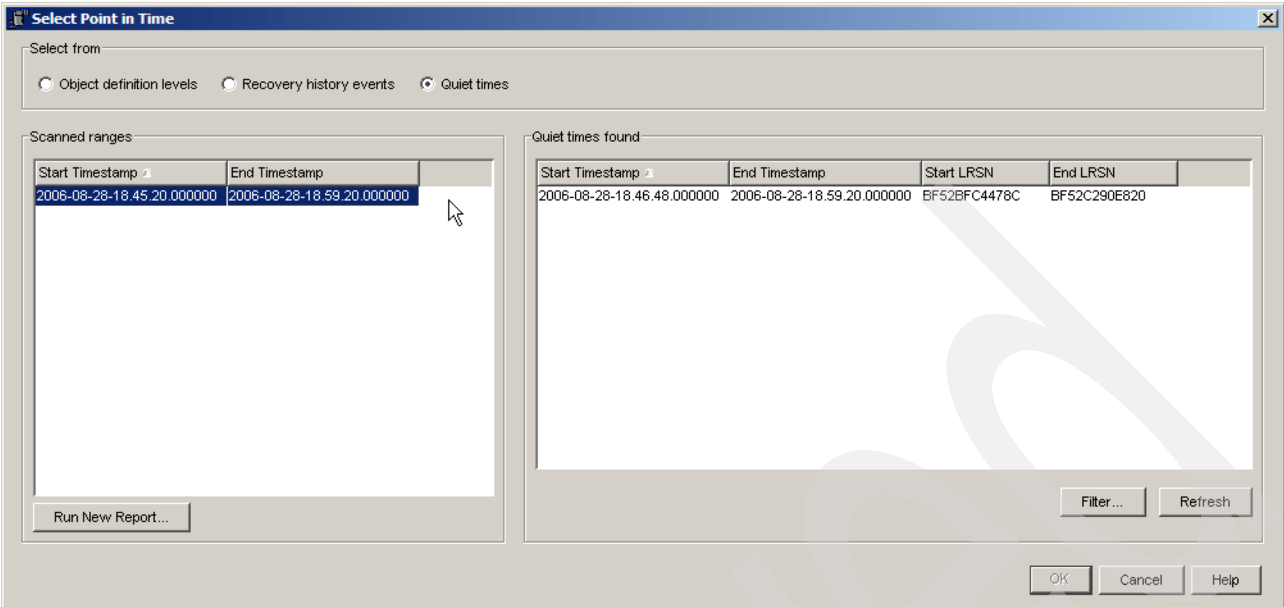

Figure B-78   Selecting the quiet time point to recover to (5/7)

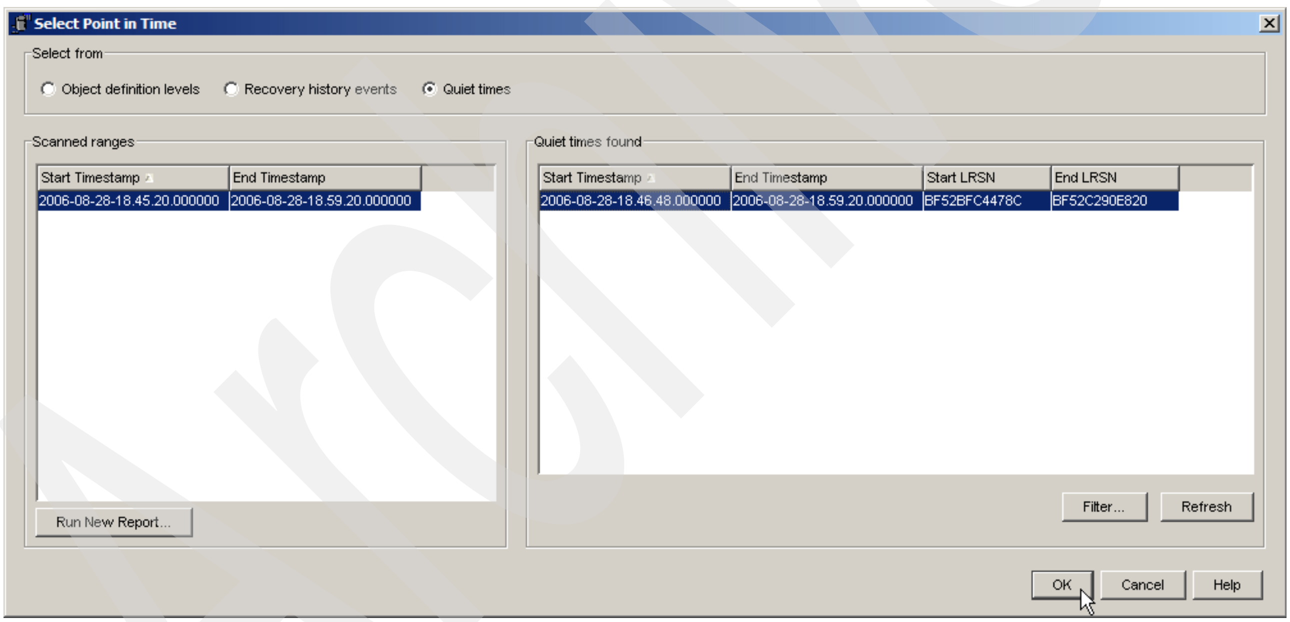

Figure B-79   Selecting the quiet time point to recover to (6/7)

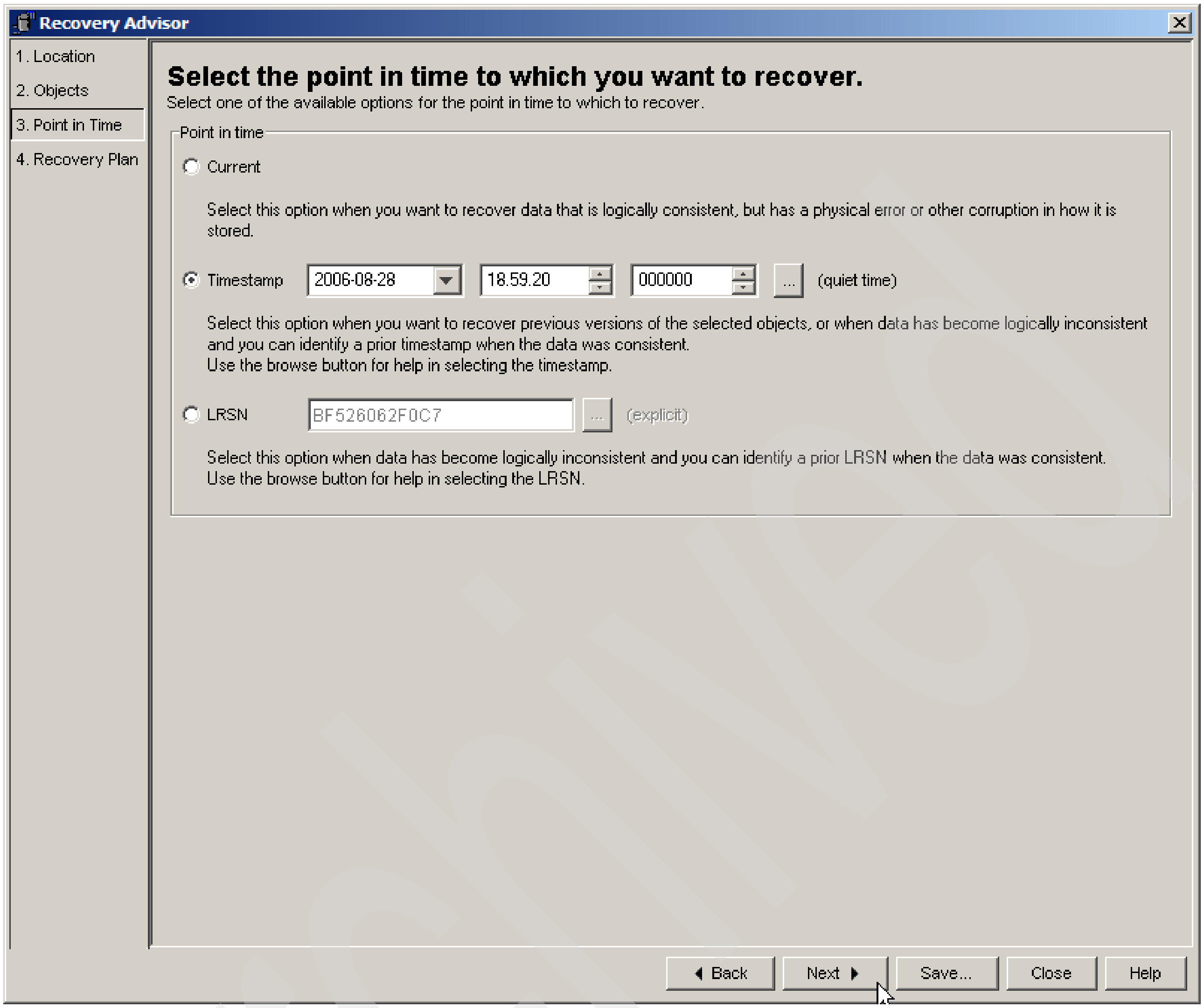

Figure B-80 Selecting the quiet time point to recover to (7/7)

B.7 Specifications

As mentioned previously, when you have to interrupt your Recovery Expert for z/OS GUI client activity to attend to more pressing matters, Recovery Expert for z/OS GUI client provides the option to save specifications to continue at a later time. You can also share specifications with your colleagues, if required.

The following windows guide you through a typical example of how to save a specification and then retrieve it at a later time. To minimize clutter, we do not show all the window interactions here.

1. Assume that you are working on generating a recovery specification for table space SNGLT1TS, as shown in Figure B-81 on page 293, and an emergency occurs requiring your services elsewhere urgently. To prevent the loss of the work done so far, click **Save** in Figure B-81 on page 293 to save the specification. This takes you to Figure B-82 on page 294.

2. In Figure B-82 on page 294, provide the necessary details and specify whether this specification is shareable by others. Click **OK** to view the saved message, as shown in Figure B-83 on page 294.
3. After you have dealt with the emergency, you can retrieve the saved specification by connecting to the Recovery Expert for z/OS server and clicking **Specifications**, as shown in Figure B-84 on page 295.
4. The window shown in Figure B-85 on page 296 opens. Specify the filtering criteria and click **Refresh** to view the saved specifications, as shown in Figure B-86 on page 296. Highlight the recently saved specification and click the open folder icon in Figure B-87 on page 297 to open this specification. This takes you to Figure B-88 on page 298. You can now carry on working where you left off as though there had been no interruption.
5. Figure B-89 on page 299 through Figure B-93 on page 301 show other operations that you can perform on the highlighted specification such as rename, export, copy, and delete. You can also import other specifications.
6. It is also possible to save a specification while a task is in progress, as shown in Figure B-94 on page 301 through Figure B-98 on page 304. After you generate a recovery plan and click the **Run** button (Figure B-94 on page 301), you decide to close the Recovery Advisor. This prompts you to save the specification, as shown in Figure B-95 on page 302. If you save the specification, you return to the Launchpad window with a message indicating that the recovery task is running, as shown in Figure B-96 on page 302. You can close this and restart later, and the status of the recovery task is updated to indicate the current status. You can then open the specification by opening the message in the Launchpad window shown in Figure B-96 on page 302, and the Recovery Advisor automatically returns to the step it is currently in. Alternatively, if it has completed, it takes you to the next logical step. If the task is still processing, then the elapsed time indicator indicates the total elapsed time for the task, as shown in Figure B-97 on page 303. After the task completes, the Recovery Advisor proceeds to the next logical tab to review the job results, as shown in Figure B-98 on page 304.

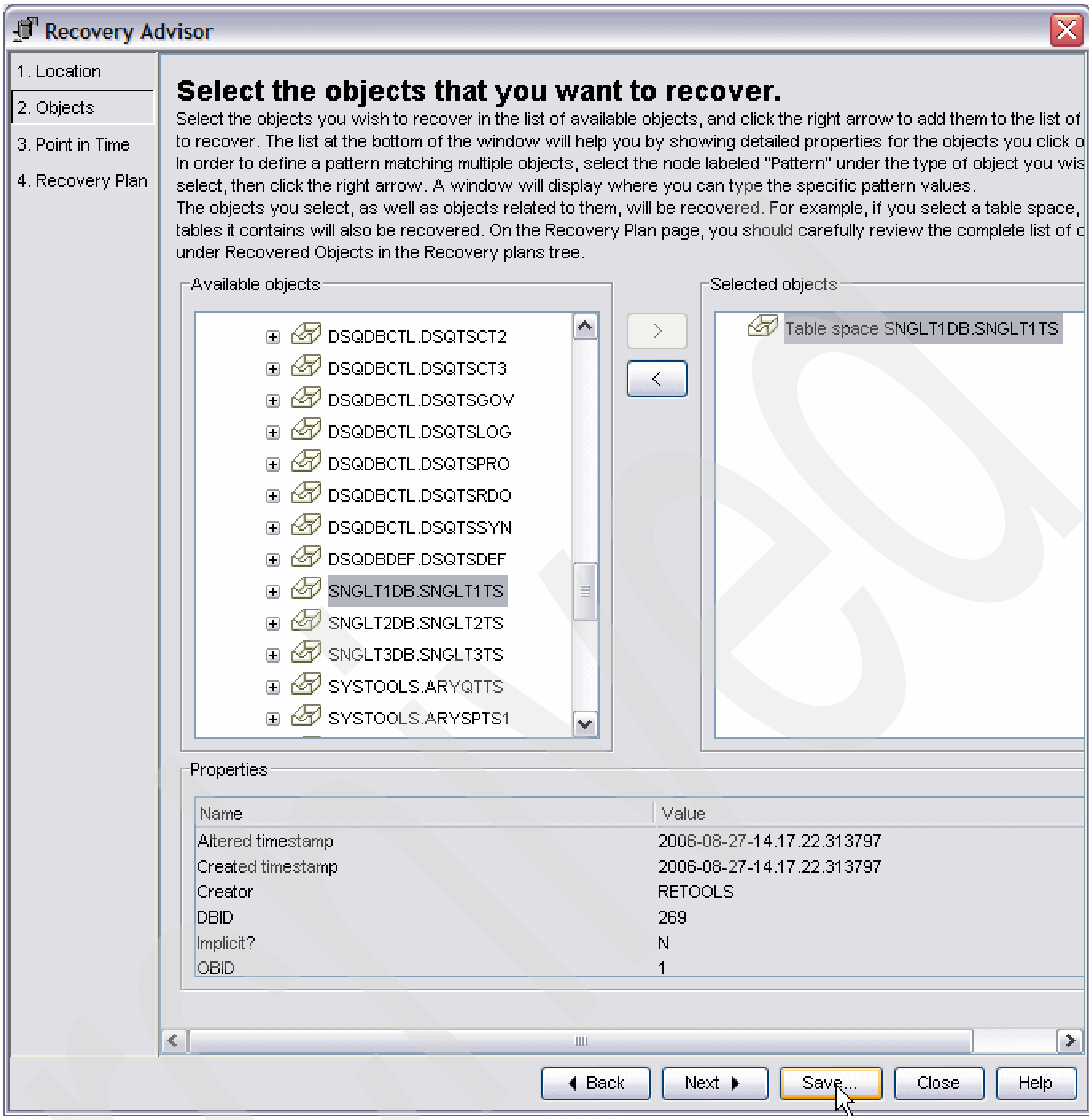

Figure B-81 Selecting objects SINGLT1TS table space to recover: Save specification

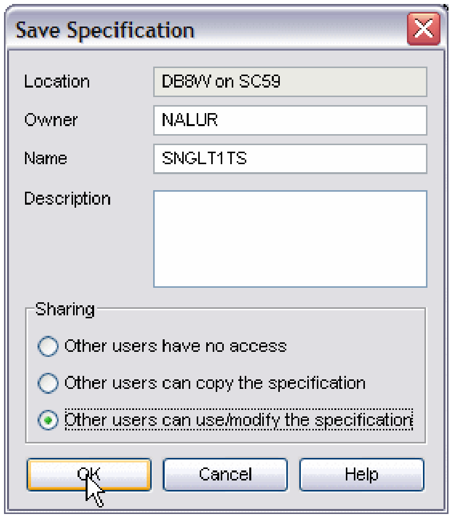

Figure B-82 Save specification details

Figure B-83 Specification saved message

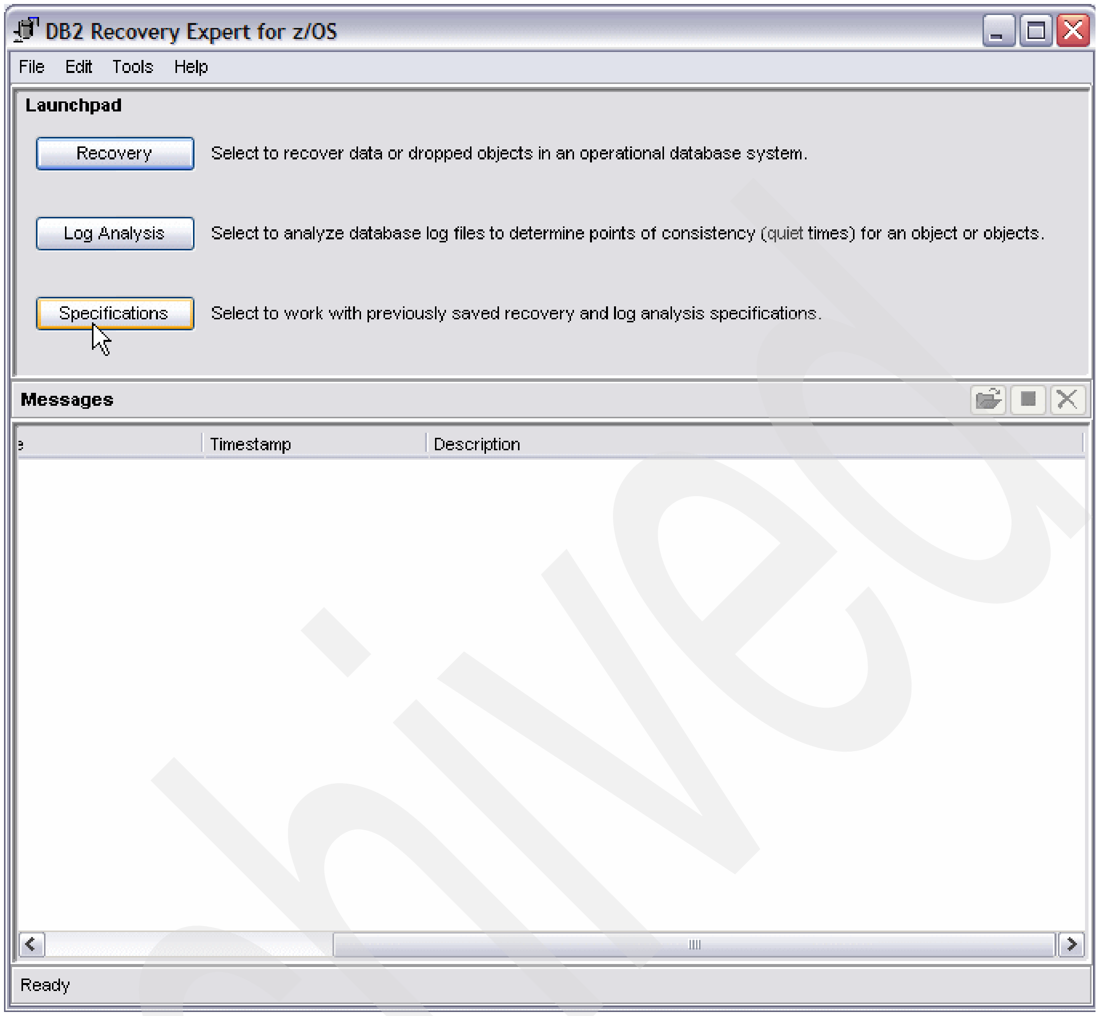

Figure B-84 Launchpad

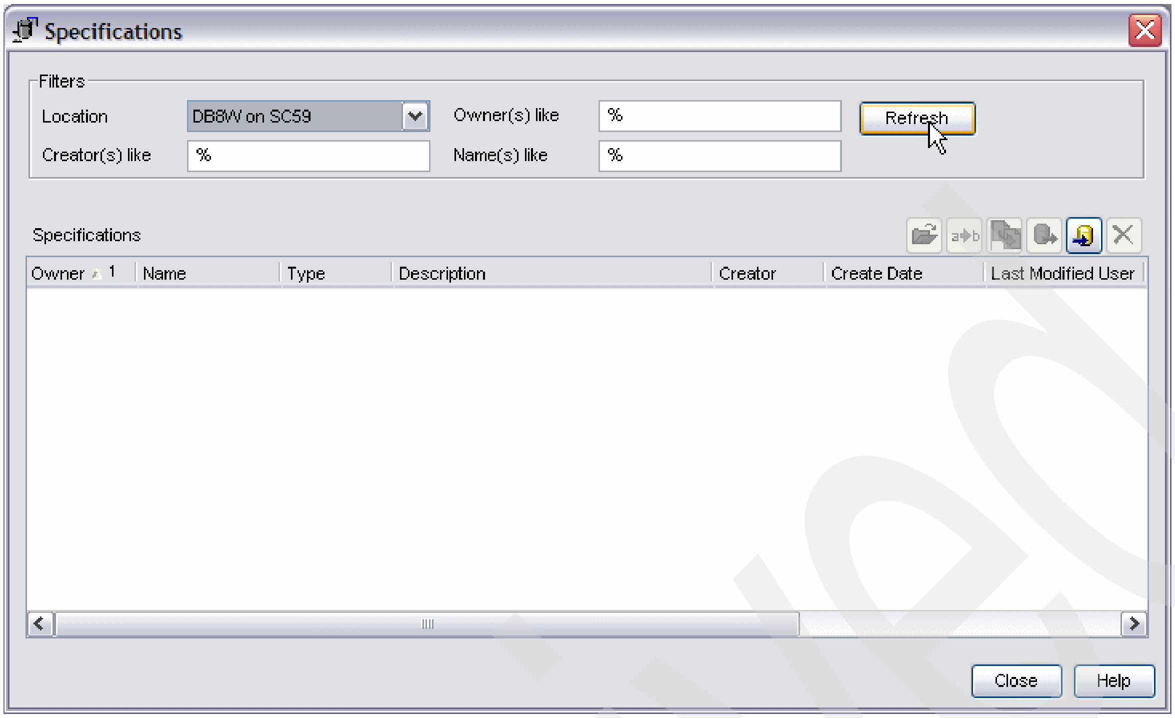

Figure B-85 Filtering specifications of interest (1/2)

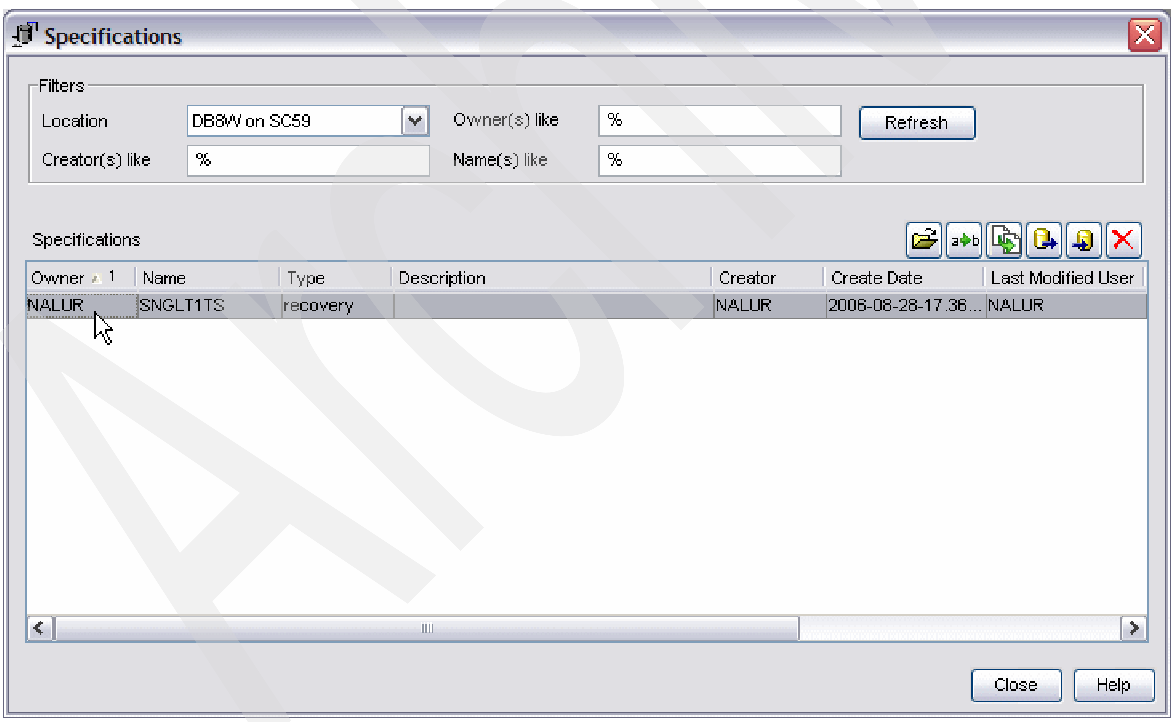

Figure B-86 Filtering specifications of interest (2/2)

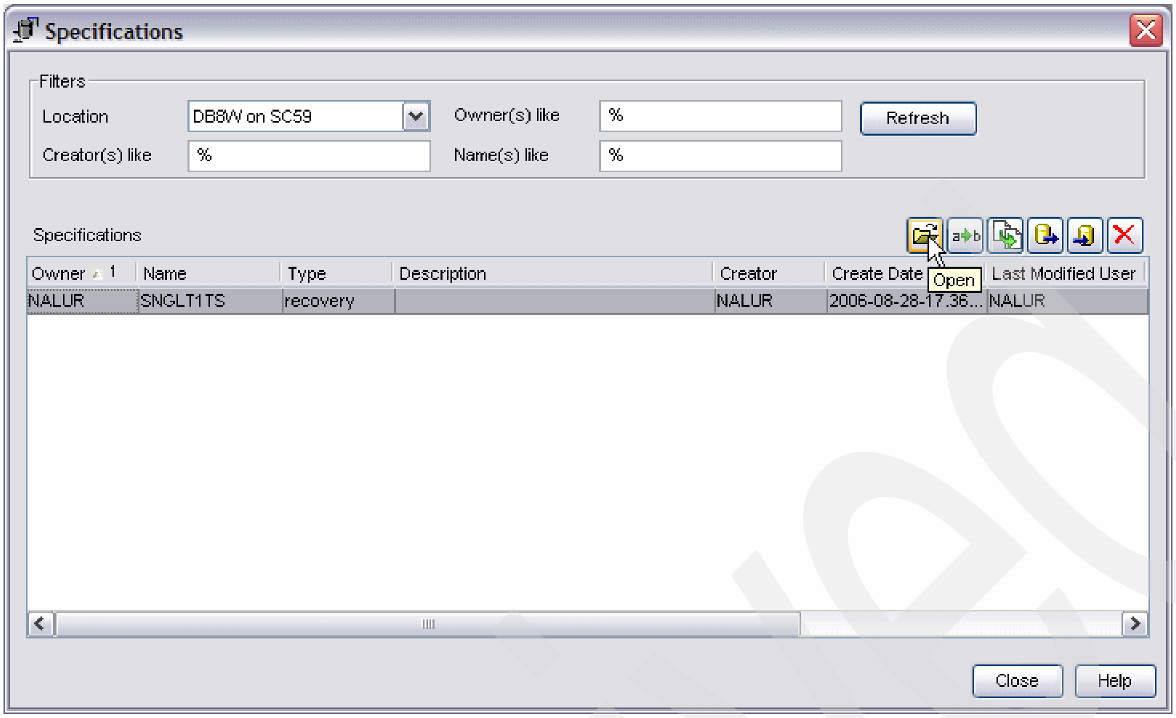

Figure B-87 Opening highlighted specification (1/2)

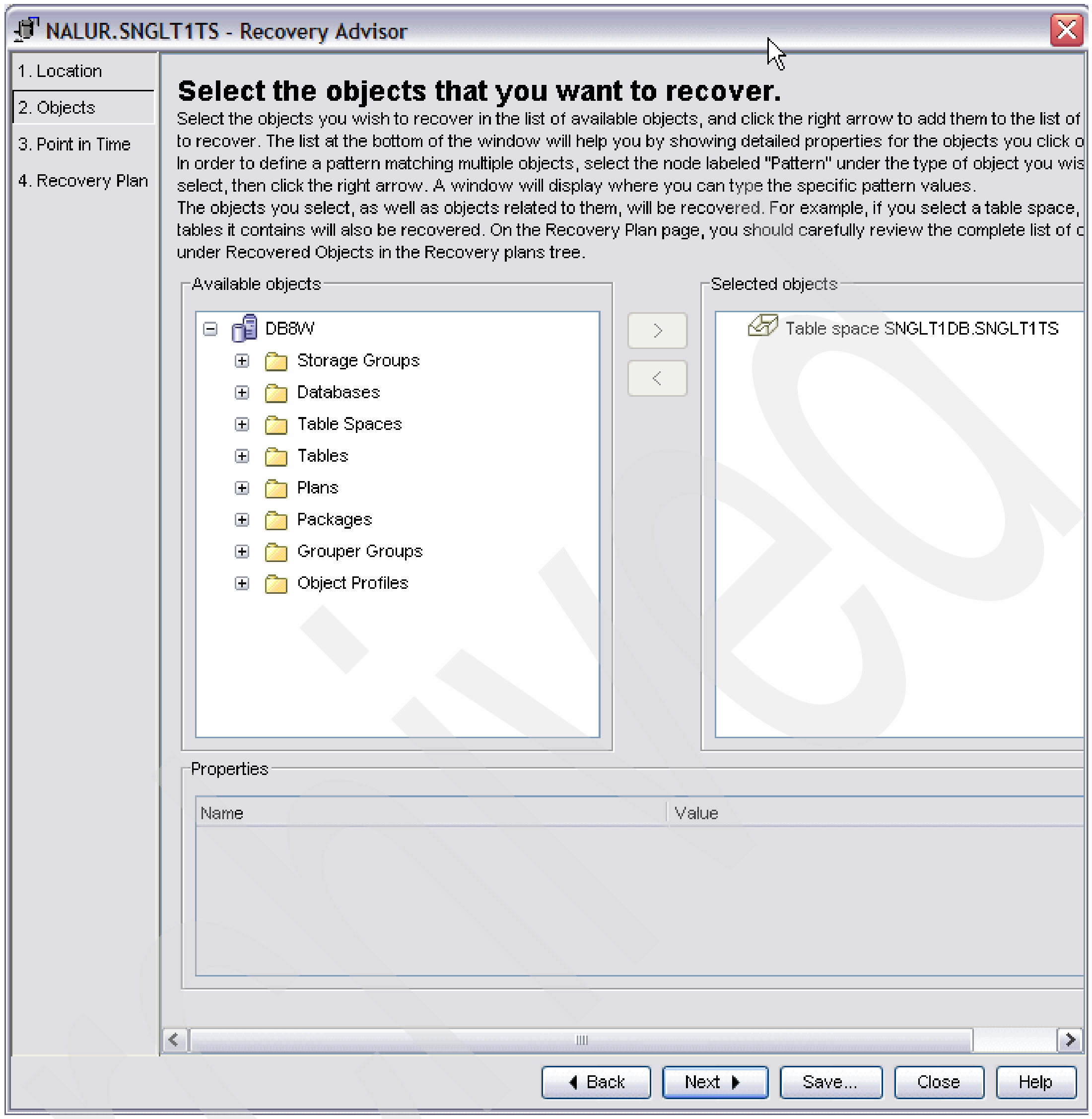

Figure B-88 Opening highlighted specification (2/2)

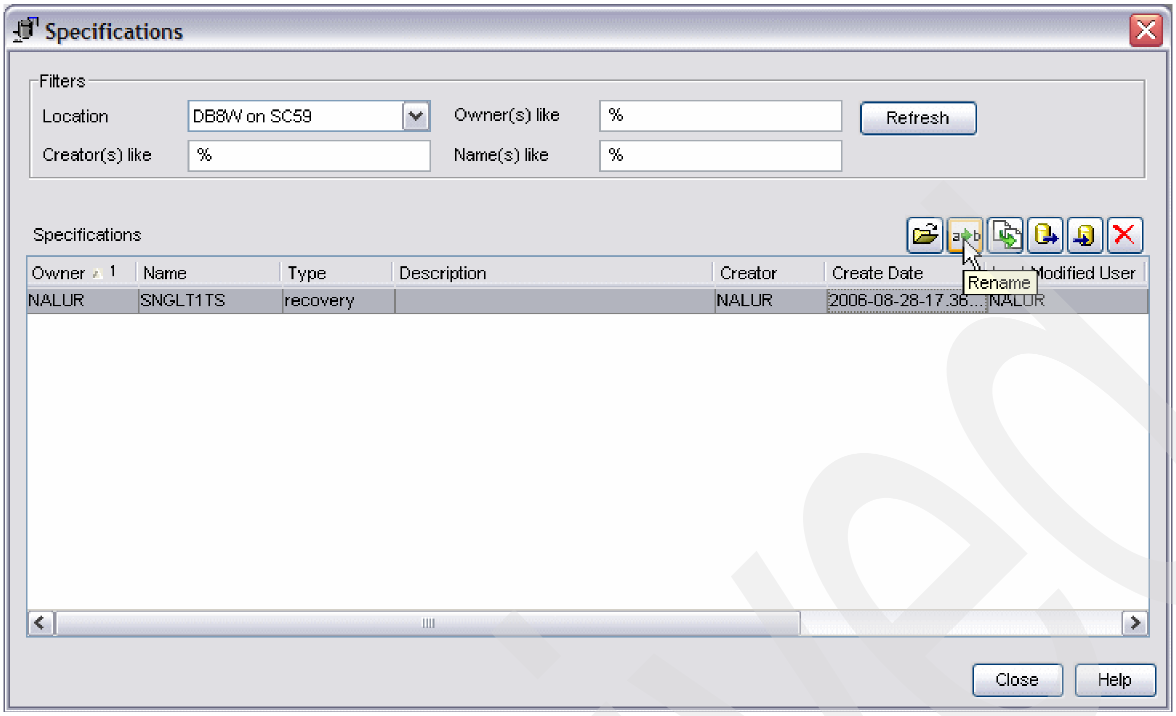

Figure B-89   Rename option for highlighted specification

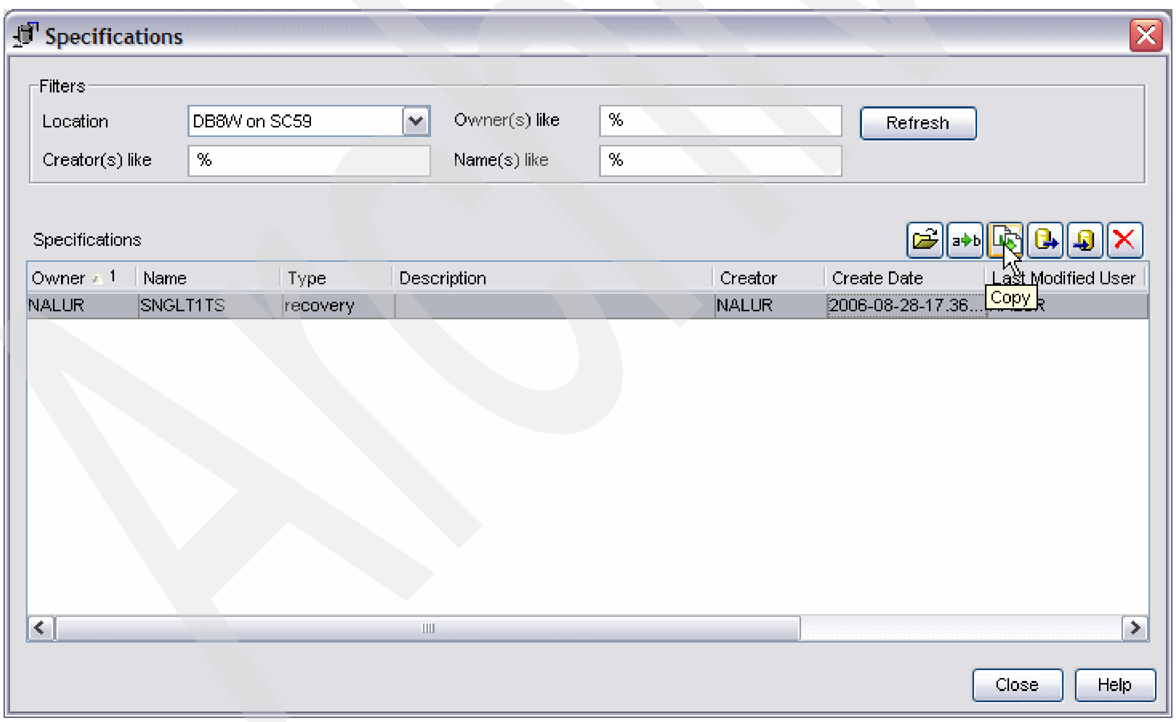

Figure B-90   Copy option for highlighted specification

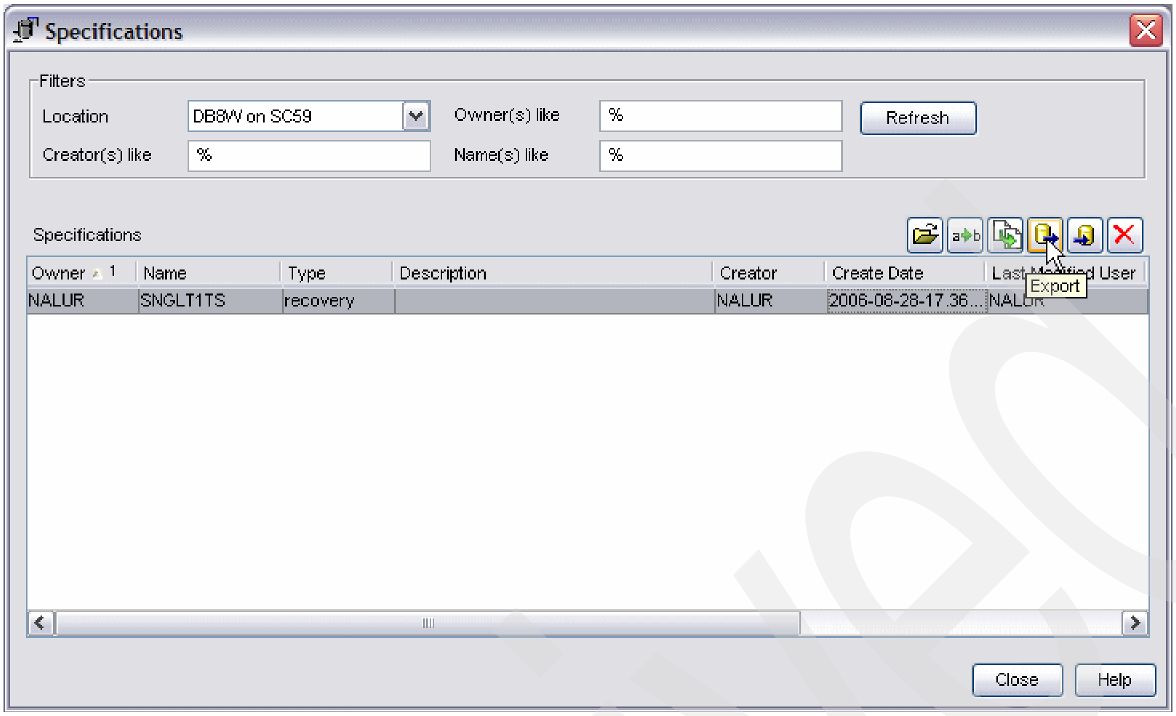

Figure B-91 Export option for highlighted specification

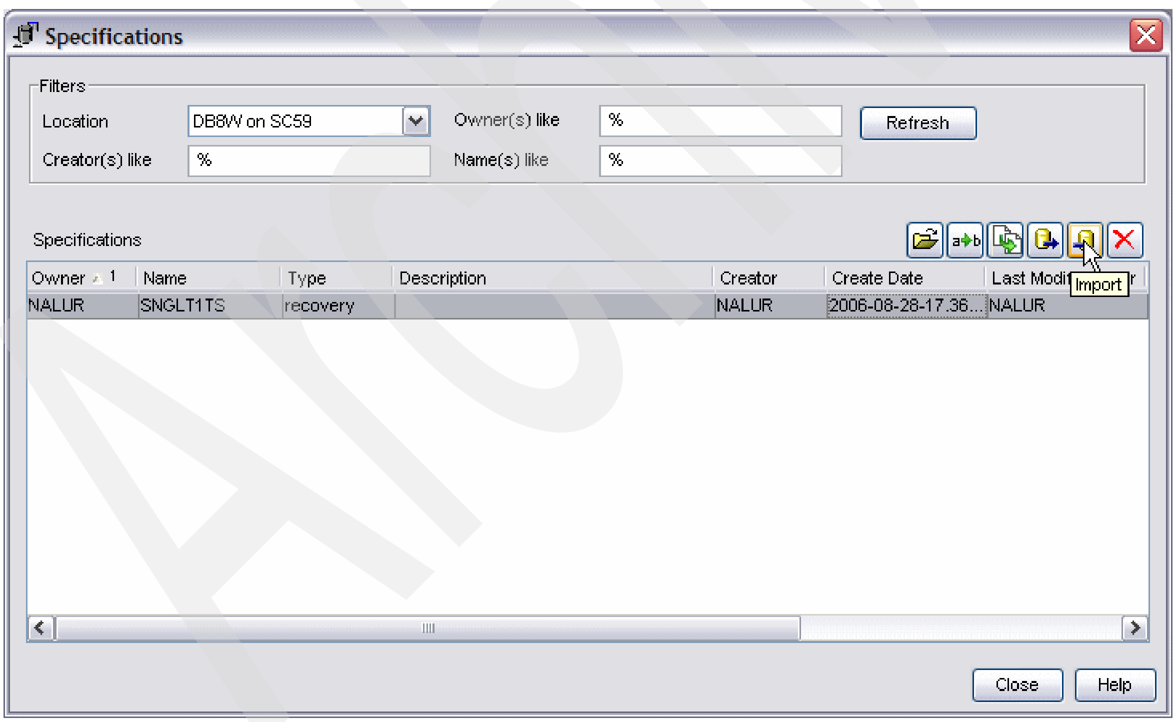

Figure B-92 Import a specification option

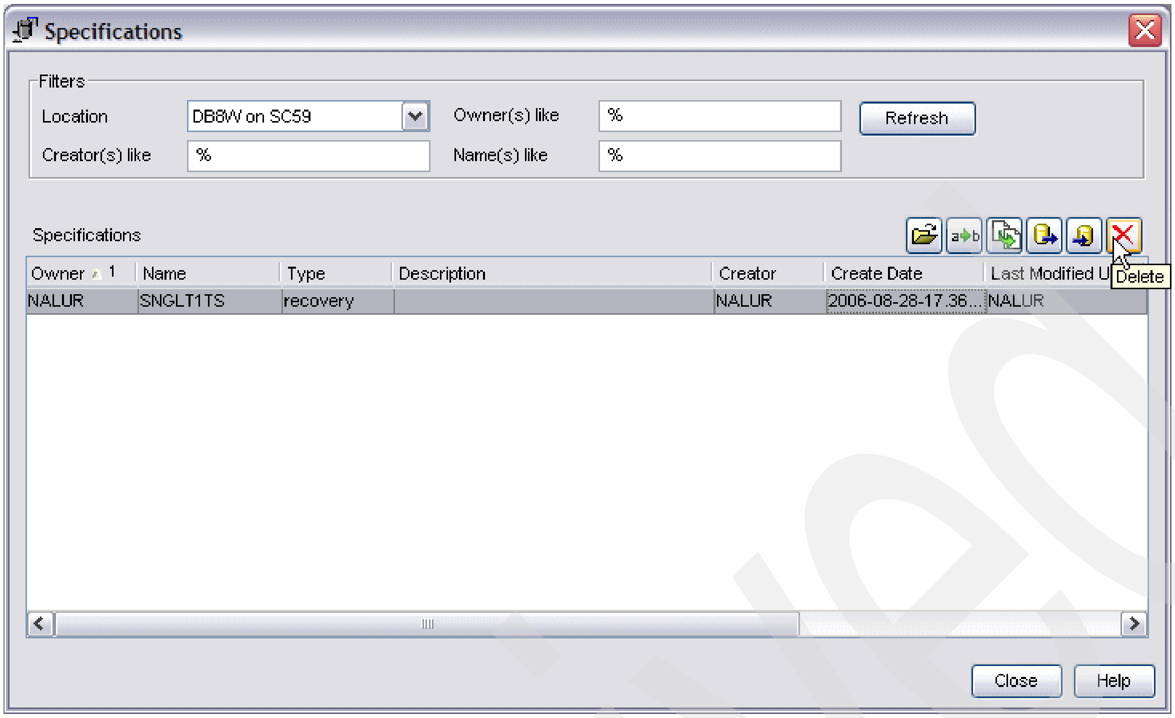

Figure B-93 Delete option for highlighted specification

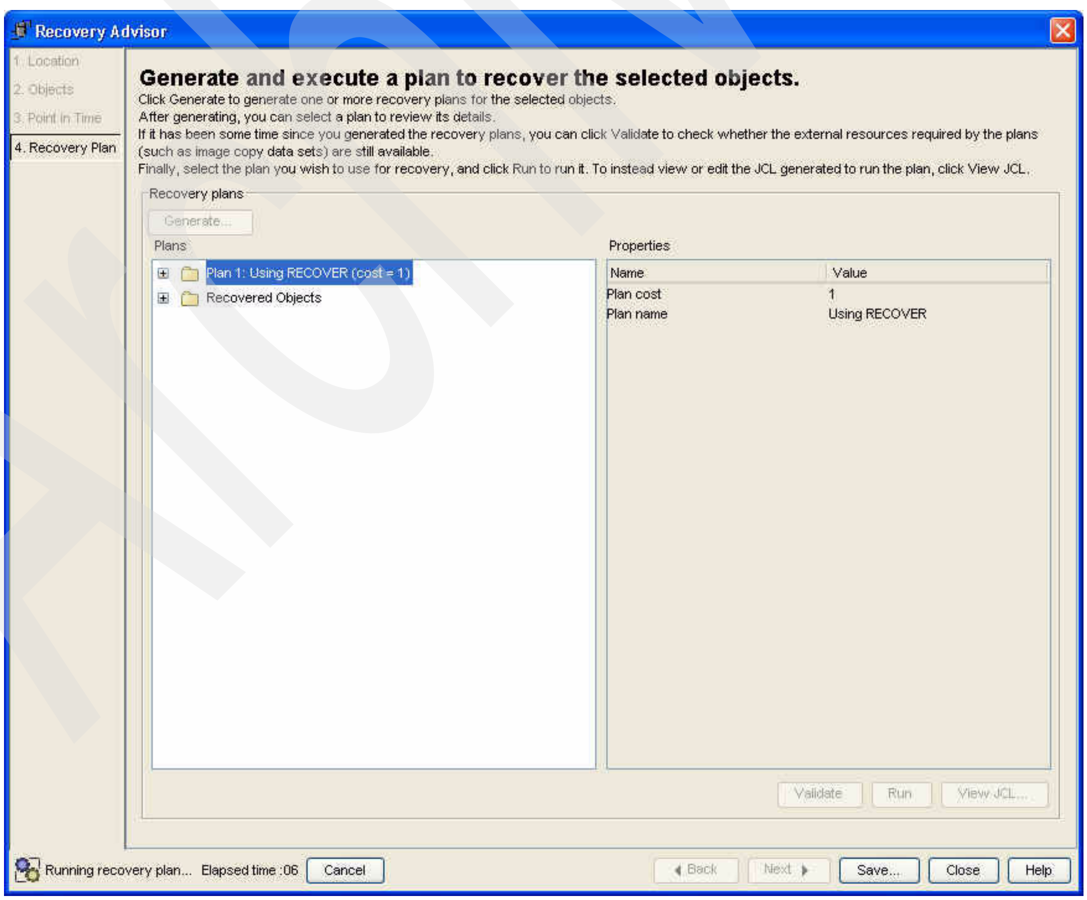

Figure B-94 Running recovery plan

Figure B-95 Save specification prompt

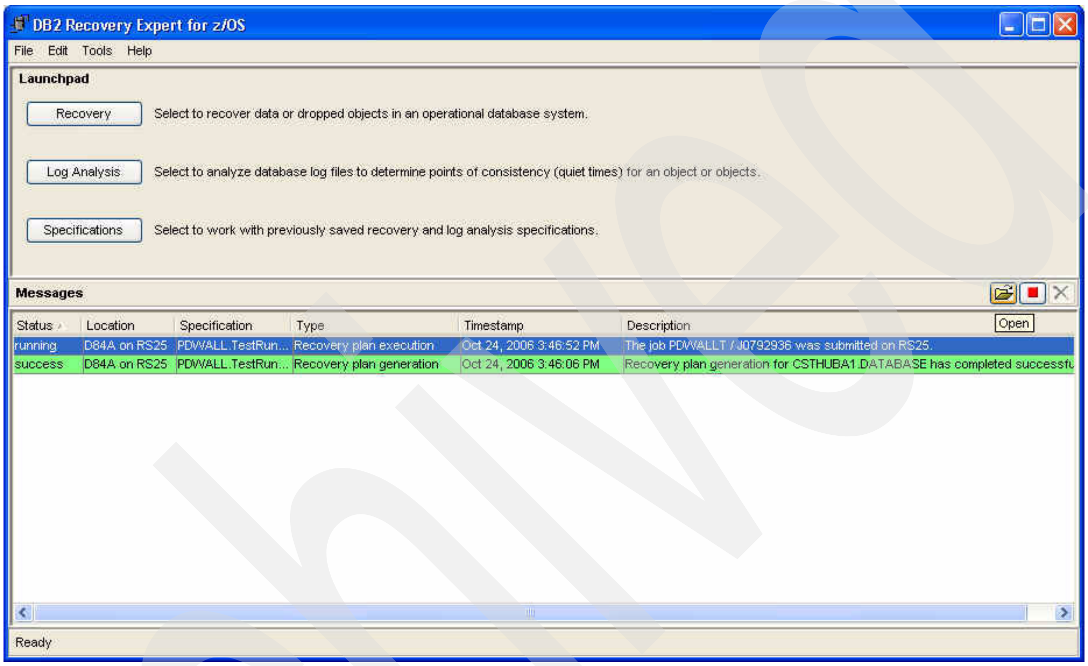

Figure B-96 Recovery running status in launchpad

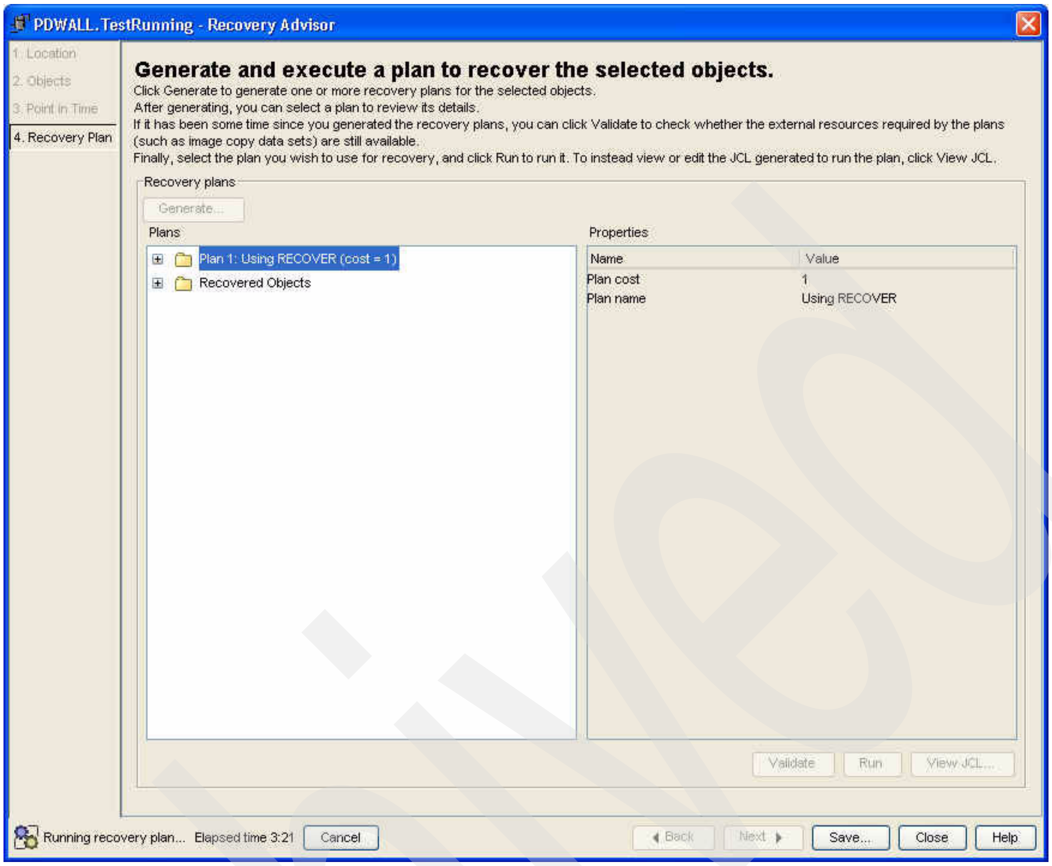

Figure B-97 Opened specification running recovery plan

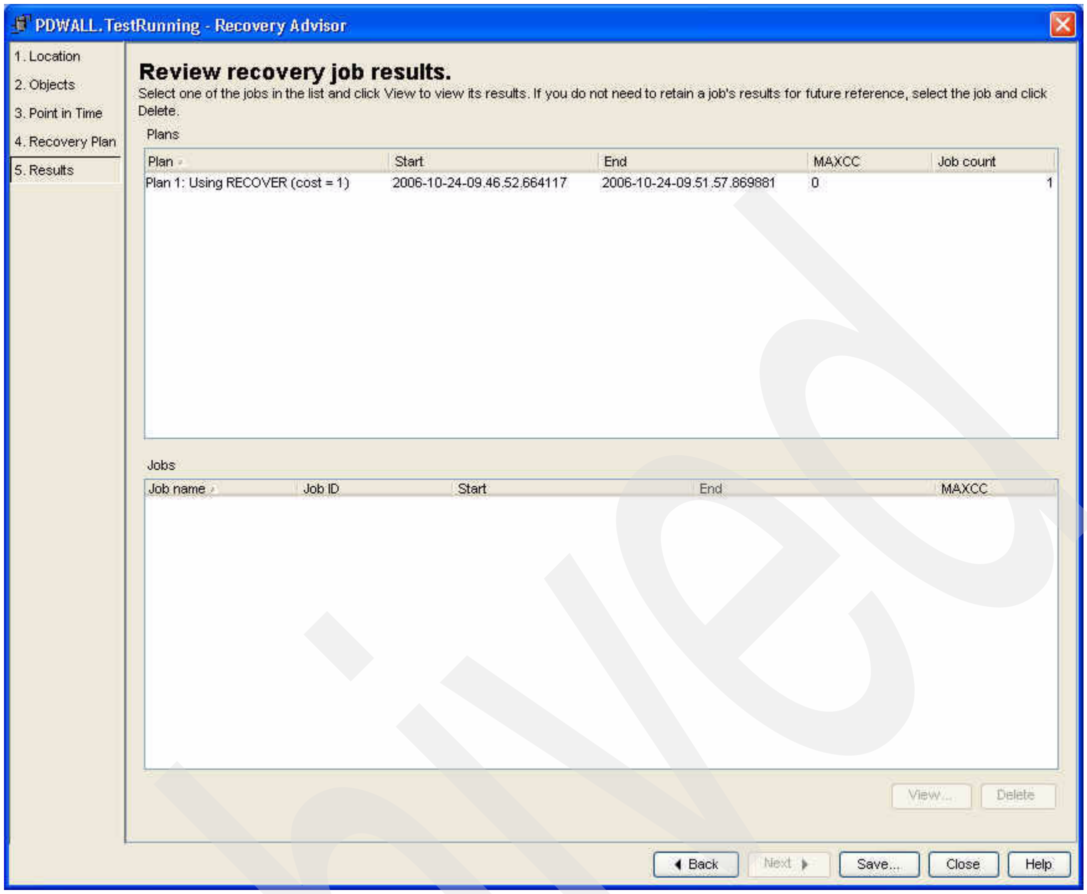

Figure B-98 Reviewing recovery job results

C

Overview of schema level repository and maintenance considerations

In this appendix, we provide an overview of the schema level repository (SLR) and recommendations for its maintenance.

C.1 Introduction

The SLR is a set of 51 DB2 tables (as shown in Table C-1 on page 309) that serves as an archive to hold object definitions and alterations to object definitions. The default database name for the SLR is SYSTOOLS and the default schema name is SYSTOOLS. You can customize these defaults. You have to run the ARYDDLA, ARYDDL8[1], ARYDDL81[2], and ARYDDL82[3] members in the ARY.V1R1M0.SARYSAMP library to define the SLR objects in the appropriate DB2 subsystem.

During installation, the SLR is loaded with data extracted from the DB2 catalog tables by the ARYSJ002 job in the ARY.V1R1M0.SARYSAMP library. The SLR is critical to the recovery of dropped objects and point-in-time recovery of existing[4] objects.

> **Important:** If the SLR does not hold object definitions, alterations to object definitions including the fact that the object was dropped, and image copy information in it, recovery of dropped objects is *not* possible. Therefore, it is critical that you schedule frequent updates of the SLR to ensure proper recovery of dropped objects. As an extreme example, update the SLR whenever object definitions are created, changed, image copies and quiesce points taken. In practice, you can schedule an SLR update every few hours or daily. If you perform important application object definition updates, then update the SLR immediately after the object definition updates.
>
> If the SLR does not contain object definitions, alterations to object definitions, and image copy information in it, then the recovery of the undropped object to a point-in-time might not produce a result of synchronizing the data and the DDL at the required moment in time.

When an object is created, it your responsibility to update the SLR to record this information. When you update the SLR, it creates a row in one of the SLR tables with a creation time that corresponds to the creation time of the object. This entry in the SLR has a creation time and a blank value for the end time of this object definition level. When you alter this object, such as adding a column and running the SLR update again, a new entry is created for this object in the SLR with a new creation time corresponding to when the SLR update is performed (not the exact time when the object is created) and a blank value for the end time of this object definition level. The previous entry (which now represents the earlier object definition level of this object) is now given an end time which is one microsecond less than the creation time of the new entry. Figure C-1 on page 307 shows an example of multiple object definition levels for an object as recorded in the SLR.

> **Note:** Establish each DB2 subsystem in the SLR by tailoring the ARYSJ001 for that subsystem and running it.

[1] ARYDDL7 for DB2 for z/OS Version 7
[2] ARYDDL71 for DB2 for z/OS Version 7
[3] ARYDDL72 for DB2 for z/OS Version 7
[4] Only when Data Definition Language (DDL) changes are made to the object over a period of time, and a recovery action is requested to a point-in-time such as an image copy that was taken when the DB2 object reflected an earlier DDL version of the object, or following image copies so that recovery of dropped objects can be performed.

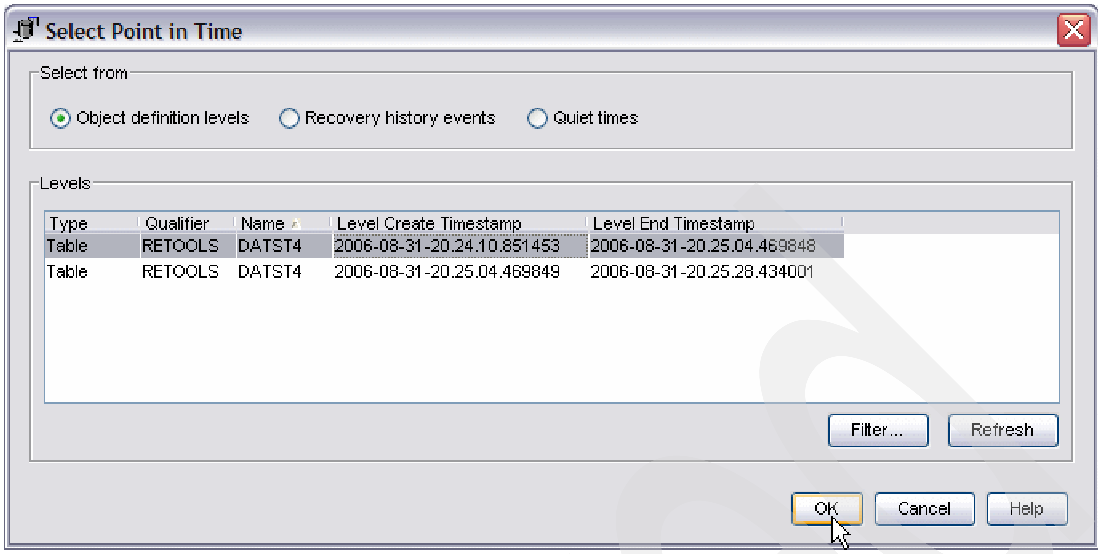

Figure C-1 Object definition level entries in the SLR

Recovery Expert provides a sample library member ARYSJ002, which contains the job control language (JCL) to update the SLR. The initial run of job ARYSJ002 might take multiple hours to copy the contents of the DB2 system catalog. Run times vary depending on the DB2 system catalog size.

The following steps list the tasks that are required to update the schema level repository:

1. Edit the ARY.V1R1M0.SARYSAMP member ARYSJ002.
2. Add the appropriate job card to ARYSJ002.
3. Change #SARYLOAD to the appropriate IBM DB2 Recovery Expert for Z/OS product libraries.
4. Change #SDSNLOAD to the appropriate DB2 load library.
5. Change #SSID to the target DB2 subsystem ID.
6. Change the DB2PARMS data set name to the product Virtual Storage Access Method (VSAM) control data set that is created by the sample JCL provided in member ARYSJ000.
7. Modify the CAPTURE PROFILES statement as instructed in the sample member ARYSJ002.
8. Run ARYSJ002.

Attention: The SLR update job must end with a return code of zero. If a non-zero return code occurs, review the job output for errors. Correct the problem and resubmit the JCL.

The Recovery Expert GUI client options tab, shown in Figure C-2 on page 308, must identify the SLR created and updated previously, to store the quiet times. The quiet times are then displayed in the quiet times window, as shown in Figure C-3 on page 308.

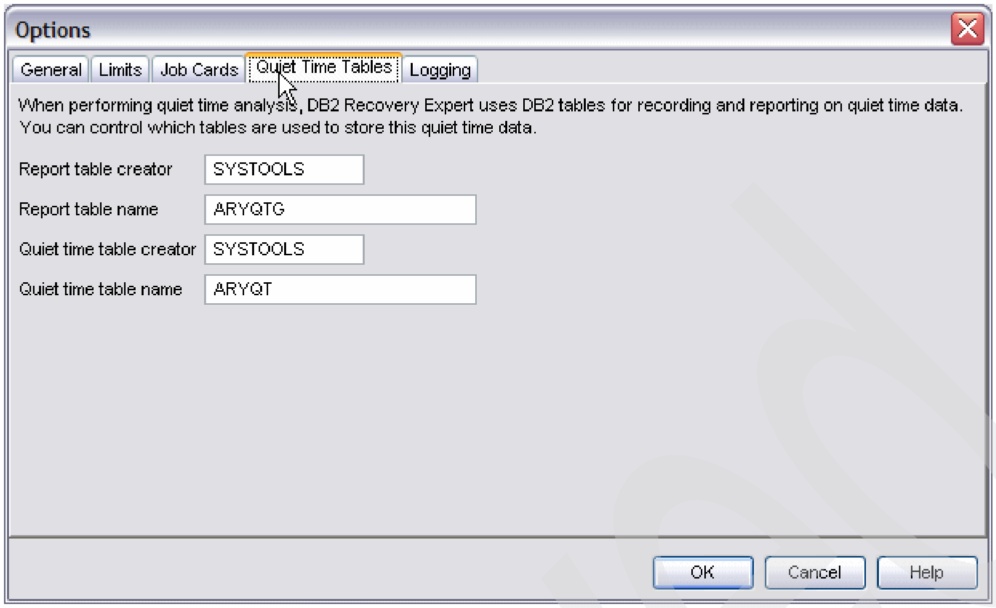

Figure C-2 SLR table details for quiet times

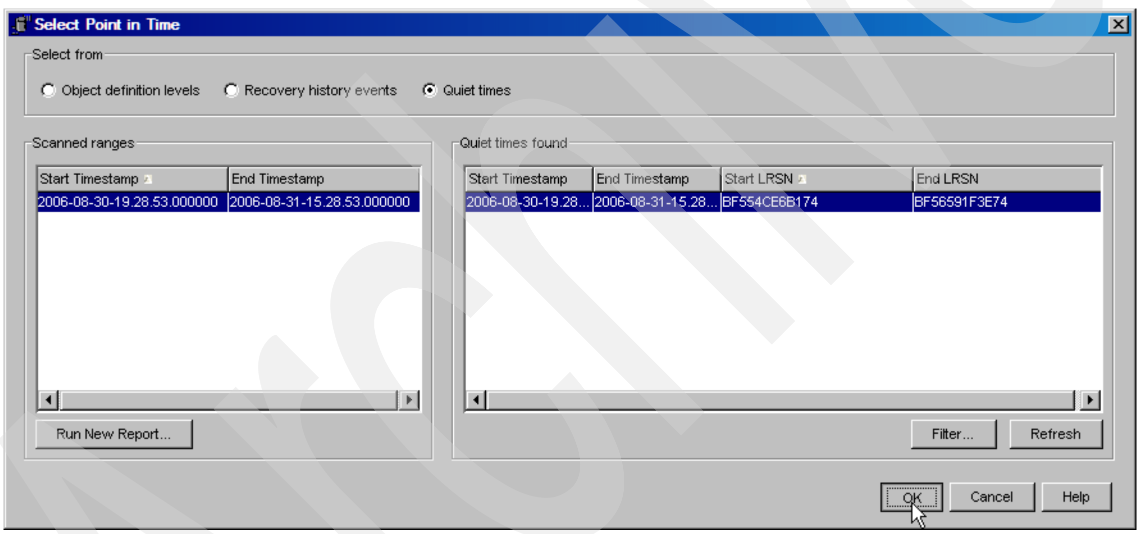

Figure C-3 Quiet times

C.2 SLR tables

Table C-1 on page 309 shows the list of SLR tables in the SYSTOOLS (default) database, and the corresponding DB2 catalog tables from which it populates these tables. A brief description of the SLR tables is included. The SLR tables have a number of indexes and views on them which are not shown here.The SLR tables also have additional columns as compared to their corresponding DB2 catalog tables.

Not all DB2 catalog tables have an equivalent SLR table and not all SLR tables have an equivalent DB2 catalog table. Eight SLR tables have no equivalent DB2 catalog tables because they contain information about the execution of the Recovery Expert for z/OS environment. These tables are ARYQT, ARYQTG, ARYSPEC_DATA, ARYSPEC_DIR, ARYVIEWSR, ARYVRLOG, ARYPROFILES, and ARYOBJECTS.

Table C-1 SLR tables in comparison with DB2 catalog tables

SLR tables (SYSTOOLS)	Corresponding DB2 tables (SYSIBM)	Description
ARYCOPY	SYSCOPY	Contains information required for recovery
ARYCOLAUTH	SYSCOLAUTH	Records the UPDATE or REFERENCES privileges that are held by users on individual columns of a table or view
ARYCOLUMNS	SYSCOLUMNS	Contains one row for every column of each table and view
ARYFIELDS	SYSFIELDS	Contains one row for every column that has a field procedure
ARYFOREIGNKEYS	SYSFOREIGNKEYS	Contains one row for every column of every foreign key
ARYINDEXES	SYSINDEXES	Contains one row for every index
ARYINDEXPART	SYSINDEXPART	Contains one row for each non-partitioned secondary index (NPSI) and one row for each partition of a partitioning index or a data-partitioned secondary index (DPSI)
ARYKEYS	SYSKEYS	Contains one row for each column of an index key
ARYRELS	SYSRELS	Contains one row for every referential constraint
ARYSYNONYMS	SYSSYNONYMS	Contains one row for each synonym of a table or view
ARYTABAUTH	SYSTABAUTH	Records the privileges that users hold on tables and views
ARYTABLEPART	SYSTABLEPART	Contains one row for each non-partitioned table space and one row for each partition of a partitioned table space
ARYTABLES	SYSTABLES	Contains one row for each table, view, or alias
ARYTABLESPACE	SYSTABLESPACE	Contains one row for each table space
ARYDATABASE	SYSDATABASE	Contains one row for each database, except for database DSNDB01
ARYDBAUTH	SYSDBAUTH	Records the privileges that are held by users over databases
ARYRESAUTH	SYSRESAUTH	Records CREATE IN and PACKADM ON privileges for collections; USAGE privileges for distinct types; and USE privileges for buffer pools, storage groups, and table spaces
ARYSTOGROUP	SYSSTOGROUP	Contains one row for each storage group
ARYVOLUMES	SYSVOLUMES	Contains one row for each volume of each storage group
ARYAUXRELS	SYSAUXRELS	Contains one row for each auxiliary table created for a large object (LOB) column; a base table space that is partitioned must have one auxiliary table for each partition of each LOB column
ARYDATATYPES	SYSDATATYPES	Contains one row for each distinct type defined to the system
ARYPARMS	SYSPARMS	Contains a row for each parameter of a routine or multiple rows for table parameters (one for each column of the table)
ARYROUTINEAUTH	SYSROUTINEAUTH	Records the privileges that are held by users on routines; a routine can be a user-defined function, cast function, or stored procedure
ARYROUTINES	SYSROUTINES	Contains a row for every routine; a routine can be a user-defined function, cast function, or stored procedure

SLR tables (SYSTOOLS)	Corresponding DB2 tables (SYSIBM)	Description
ARYSCHEMAAUTH	SYSSCHEMAAUTH	Contains one or more rows for each user who is granted a privilege on a particular schema in the database
ARYTRIGGERS	SYSTRIGGERS	Contains one row for each trigger
ARYSEQUENCEAUTH	SYSSEQUENCEAUTH	Records the privileges that are held by users over sequences
ARYSEQUENCES	SYSSEQUENCES	Contains one row for each identity column or user-defined sequence
ARYCHECKS	SYSCHECKS	Contains one row for each check constraint
ARYCHECKS2	SYSCHECKS2	Contains one row for each table check constraint for catalog tables created in or after Version 7; check that constraints for catalog tables created before Version 7 are not included in this table
ARYUSERAUTH	SYSUSERAUTH	Records the system privileges that are held by users
ARYVIEWDEP	SYSVIEWDEP	Records the dependencies of views on tables, functions, and other views
ARYVIEWS	SYSVIEWS	Contains one or more rows for each view, materialized query table, or user-defined Structured Query Language (SQL) function
ARYVIEWSR	SYSVIEWS	Contains user-defined function (UDF) and procedure-related information; the ARYVIEWSR table is loaded by the SLR update utility (ADHSJ002) with SQL text for SQ- based routines. In the DB2 catalog, this text is stored in SYSVIEWS table.
ARYVRLOG	No equivalent	Records utility execution statistics in multiple entries (one for each SLR table updated) for execution of the SLR update utility; Recovery Expert for z/OS also records some internal data in this table
ARYPACKAGE	SYSPACKAGE	Contains a row for every package
ARYPACKAUTH	SYSPACKAUTH	Records the privileges that are held by users over packages
ARYPACKDEP	SYSPACKDEP	Records the dependencies of packages on local tables, views, synonyms, table spaces, indexes, aliases, functions, and stored procedures
ARYPACKLIST	SYSPACKLIST	Contains one or more rows for every local application plan bound with a package list; each row represents a unique entry in the plan's package list
ARYPLSYSTEM	SYSPLSYSTEM	Contains zero or more rows for every plan; each row for a given plan represents one or more connections to an environment on which the plan can be used
ARYPKSYSTEM	SYSPKSYSTEM	Contains zero or more rows for every package; each row for a given package represents one or more connections to an environment on which the package can be run
ARYDBRM	SYSDBRM	Contains one row for each DBRM of each application plan
ARYPLAN	SYSPLAN	Contains one row for each application plan
ARYPLANAUTH	SYSPLANAUTH	Records the privileges that are held by users over application plans

SLR tables (SYSTOOLS)	Corresponding DB2 tables (SYSIBM)	Description
ARYPLANDEP	SYSPLANDEP	Records the dependencies of plans on tables, views, aliases, synonyms, table spaces, indexes, functions, and stored procedures
ARYPROFILES	No equivalent	Is used to archive DB2 Automation Tool object profiles
ARYOBJECTS	No equivalent	Is used to archive DB2 Automation Tool object profiles
ARYSPEC_DIR	No equivalent	Contains one row for each saved specification
ARYSPEC_DATA	No equivalent	Contains the detailed data associated with each specification
ARYQTG	No equivalent	Contains the group ID that relates multiple quiet times that are detected after running log analysis
ARYQT	No equivalent	Contains the quiet times detected after running log analysis

C.3 SLR maintenance considerations

As mentioned previously, the SLR is critical to the ability of Recovery Expert to recover dropped objects, and to synchronize data and DDL when an object is recovered to a prior point-in-time. To perform this function effectively, keep the SLR up-to-date with information about creation, alteration, and dropping of objects, and creation of recoverable resources such as full and incremental image copies, and also quiesce points.

The SLR is loaded initially with the contents of the DB2 catalog tables when Recovery Expert is installed using the ARYSJ002 job in the ARY.V1R1M0.SARYSAMP library. As mentioned previously, this process can take a long time depending on the size of the DB2 catalog. Subsequently, it is the user's responsibility to ensure that the SLR is updated at appropriate frequent intervals. We recommend that you perform these updates at least daily, and when important application changes occur in your DB2 environment.

Besides updating the SLR at frequent intervals, it is also important to manage the SLR database, which includes performing the following tasks:

- Take backups of DB2 objects of the SLR at frequent intervals. Recovery Expert cannot be used to recover the SLR database.
- Monitor and tune the SLR database including RUNSTATS and REORG, as appropriate.
- Prune[5] the contents of the SLR when its entries become outdated:
  - Entries that belong to objects that have been dropped and are no longer required
  - Entries that belong to existing objects but are no longer required, such as entries related to image copies, quiesce points, quiet times, and object definition levels
  - Entries that seem to belong to the same object, but in fact belong to different objects that accidentally shared the same name

    For example, this can occur when you create a table PROD with certain columns, perform updates, take image copies, and drop it some days later. A few days later, you create another table PROD with different attributes, and perform updates, take image copies, and so on. The SLR has information about both these tables and assumes they are different versions of the same object. This can cause unpredictable results when

[5] Recovery Expert for z/OS does not provide a graphical user interface (GUI) or batch facility to prune the contents of the SLR.

performing a point-in-time recovery. You must prune the entries of the first PROD table because, in practice, it does not represent an older version of the current PROD table.

Note: The use of a date or timestamp is not likely to properly prune or delete data from the SLR tables. The timestamp that is entered in the tables does not span all the SLR tables and is the timestamp when the SLR update was run.

Important: Before you delete any entries in any of the SLR tables, first run a dummy delete (SELECT statement with the same predicates as the DELETE statement) to ensure that *only* the required rows are deleted.

Example C-1 through Example C-12 on page 319 provide some examples of SQL that you can use to prune the SLR tables depending on the circumstances. An advisory is provided with each sample.

Note: Example C-12 on page 319 shows the deletion of entries in the ARYSPEC_DATA table. Rows also get deleted from this table when the delete function for a selected specification is started from the Recovery Expert for z/OS GUI client.

Example: C-1 When delete authority is removed for objects in DB2 for z/OS

```
//*****************************************************************************
//* TABLES: ARYDBAUTH, ARYRESAUTH, ARYSCHEMAAUTH and ARYUSERAUTH
//*
//*  FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED
//*  ON YOUR SPECIFIC ENVIRONMENT AND REQUIREMENTS
//*
//*  => CHANGE
//*       'DATABASE NAME' TO THE NAME OF THE DATABASE FOR THE AUTHORITY
//*       'GRANTED BY' TO THE GRANTOR OF THE AUTHORITY
//*       'PERSON GRANTED AUTH' TO THE PERSON GRANTED AUTHORITY
//*       'TABLESPACE NAME, BPOOL NAME, ETC' TO THE APPROPRIATE NAME
//*       'NAME OF SCHEMA' TO THE NAME OF THE SCHEMA
//*
//*****************************************************************************

   DELETE FROM SYSTOOLS.ARYDBAUTH WHERE NAME = 'DATABASE NAME'
     AND GRANTOR = 'GRANTED BY'
     AND GRANTEE = 'PERSON GRANTED AUTH';

   DELETE FROM SYSTOOLS.ARYRESAUTH WHERE
     NAME = 'TABLESPACE NAME, BPOOL NAME, ETC'
     AND GRANTOR = 'GRANTED BY'
     AND GRANTEE = 'PERSON GRANTED AUTH'
     AND QUALIFIER = 'DATABASE NAME' ;

   DELETE FROM SYSTOOLS.ARYSCHEMAAUTH WHERE
     SCHEMANAME = 'NAME OF SCHEMA'
     AND GRANTOR = 'GRANTED BY'
     AND GRANTEE = 'PERSON GRANTED AUTH';
```

```
  DELETE FROM SYSTOOLS.ARYUSERAUTH   WHERE
     AND GRANTOR = 'GRANTED BY'
     AND GRANTEE = 'PERSON GRANTED AUTH';
```

Example: C-2   When image copies are deleted for objects in DB2 for z/OS

```
//********************************************************************************
//*  FOLLOWING SQL PERFORMS A DELETE FROM ARYCOPY WHICH CONTAINS SYSCOPY
//*  INFORMATION.
//*
//*  YOU SHOULD UNDERSTAND AND EVALUATE CAREFULLY HOW LONG YOU WILL NEED TO KEEP
//*  THIS INFORMATION SINCE IT IS A KEY FACTOR IN THE ABILITY OF THE RECOVERY
//*  EXPERT TOOL TO RETRIEVE AND RECREATE OBJECTS (OBJECT LEVEL DEFINITION)
//*
//* ADDITIONAL COLUMNS SUCH AS ICTIME CAN BE USED TO FURTHER LIMIT
//* YOUR DELETION BASED ON YOUR INDIVIDUAL REQUIREMENTS
//********************************************************************************
//*  TABLE: ARYCOPY
//*
//*  FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR SPECIFIC
//*  REQUIREMENTS
//*
//*  => CHANGE
//*      'DATABASE NAME' TO THE DATABASE NAME
//*      'TABLESPACE NAME' TO THE TABLESPACE NAME
//*      'IMAGE DATE' TO THE SPECIFIC IMAGE COPY DATE
//*
//*     YOU COULD ALSO PERFORM A RANGE OF DATES BY CHANGING THE SQL STATEMENT
//*     TO USE THE BETWEEN STATEMENT
//********************************************************************************

   DELETE FROM SYSTOOLS.ARYCOPY WHERE DBNAME = 'DATABASE NAME'
    AND TSNAME = 'TABLESPACE NAME' AND ICDATE = 'IMAGE DATE';
```

Example: C-3   When objects are dropped in DB2 for z/OS

```
//********************************************************************************
//* WHEN A TABLE, TABLESPACE, DATABASE IS DROPPED AND
//* 1. YOU NO LONGER WISH TO RECOVER / RESTORE THE OBJECT
//* 2. BEFORE YOU CREATE ANOTHER OBJECT WITH THE SAME NAME
//*
//*  YOU SHOULD UNDERSTAND AND EVALUATE CAREFULLY HOW LONG YOU WILL NEED TO
//*  KEEP THIS INFORMATION SINCE IT IS A KEY FACTOR IN THE ABILITY OF THE
//*  RECOVERY EXPERT TOOL TO RECOVER OR RECREATE OBJECTS
//*
//********************************************************************************
//*  TABLES: ARYCOLAUTH, ARYCOLUMNS, ARYDATABASE, ARYINDEXES, ARYINDEXPART,
//*         ARYKEYS, ARYSYNONYMS, ARYTABAUTH, ARYTABLEPART, ARYTABLES,
//*         ARYTABLESPACE, ARYVIEWDEP, ARYVIEWS
//*
//*  WE RECOMMEND THAT YOU SHOULD DELETE FROM ALL TABLES
//*
//*  FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
//*  SPECIFIC ENVIRONMENT AND REQUIREMENTS
//*
//*  => CHANGE
```

```
//*       'TABLE NAME' TO THE TABLE OR VIEW NAME
//*       'TABLE CREATOR' TO THE CREATOR OF THE TABLE OR VIEW
//*       'GRANTED BY' TO THE GRANTOR OF THE COLUMN AUTHORITY
//*       'PERSON GRANTED AUTH' TO THE PERSON GRANTED COLUMN AUTHORITY
//*       'DATABASE NAME' TO THE DATABASE NAME
//*       'DATABASE CREATOR' TO THE CREATOR OF THE DATABASE NAME
//*       'INDEX NAME' TO THE NAME OF THE INDEX
//*       'INDEX SPACE NAME' TO THE SPACE NAME FOR THE INDEX
//*       'INDEX CREATOR' TO THE CREATOR OF THE INDEX
//*       'SYNONYM NAME' TO THE NAME OF THE SYNONYM
//*       'SYNONYM CREATOR' TO THE NAME OF THE CREATOR OF THE SYNONYM
//*       'TABLESPACE NAME' TO THE NAME OF THE TABLESPACE
//*       'TABLE(T) OR VIEW(V) NAME' TO CORRECT TYPE NAME (TYPICALLY T
//*                  OR V, BUT CAN BE SEVERAL OTHERS SO REVIEW CAREFULLY
//*       'TABLESPACE CREATOR' TO THE NAME OF THE CREATOR OF TABLESPACE
//*       'VIEW NAME' TO THE NAME OF THE VIEW
//*       'VIEW CREATOR' TO THE CREATOR OF THE VIEW
//*
//********************************************************************************
   DELETE FROM SYSTOOLS.ARYCOLAUTH WHERE TNAME = 'TABLE NAME'
     AND CREATOR = 'TABLE CREATOR' AND GRANTOR = 'GRANTED BY'
     AND GRANTEE = 'PERSON GRANTED AUTH';

   DELETE FROM SYSTOOLS.ARYCOLUMNS WHERE TBNAME = 'TABLE NAME'
     AND TBCREATOR = 'TABLE CREATOR' ;

   DELETE FROM SYSTOOLS.ARYDATABASE WHERE NAME = 'DATABASE NAME'
     AND CREATOR = 'DATABASE CREATOR';

   DELETE FROM SYSTOOLS.ARYINDEXES WHERE NAME = 'INDEX NAME'
     AND TBCREATOR = 'TABLE CREATOR' AND TBNAME = 'TABLE NAME'
     AND DBNAME = 'DATABASE NAME' AND
     INDEXSPACE = 'INDEX SPACE NAME' ;

   DELETE FROM SYSTOOLS.ARYINDEXPART WHERE IXNAME = 'INDEX SPACE NAME'
     AND IXCREATOR = 'INDEX CREATOR';

   DELETE FROM SYSTOOLS.ARYKEYS WHERE IXNAME = 'INDEX NAME'
     AND IXCREATOR = 'INDEX CREATOR';

   DELETE FROM SYSTOOLS.ARYSYNONYMS WHERE NAME = 'SYNONYM NAME'
     AND CREATOR = 'SYNONYM CREATOR'
     AND TBNAME  = 'TABLE NAME'
     AND TBCREATOR = 'TABLE CREATOR';

   DELETE FROM SYSTOOLS.ARYTABAUTH WHERE TTNAME = 'TABLE NAME'
     AND TCREATOR = 'TABLE CREATOR' AND GRANTOR = 'GRANTED BY'
     AND GRANTEE = 'PERSON GRANTED AUTH';
--   NOTE GRANTEE CAN BE PUBLIC OR PUBLIC* AS WELL AS A SPECIFIC NAME

   DELETE FROM SYSTOOLS.ARYTABLEPART WHERE DBNAME = 'DATABASE NAME'
      AND TSNAME = 'TABLESPACE NAME'
      AND IXNAME = 'INDEX NAME'
      AND IXCREATOR = 'INDEX CREATOR' ;
--   NOTE IXNAME AND IXCREATOR MAY BE NULL IF THERE IS NO PARTITIONING
```

```
DELETE FROM SYSTOOLS.ARYTABLES  WHERE NAME = 'TABLE NAME'
  AND CREATOR = 'TABLE CREATOR' AND DBNAME = 'DATABASE NAME'
  AND TSNAME = 'TABLESPACE NAME' AND
  TYPE = 'TABLE(T) OR VIEW(V) NAME' ;

DELETE FROM SYSTOOLS.ARYTABLESPACE WHERE NAME = 'TABLESPACE NAME'
  AND DBNAME = 'DATABASE NAME'
  AND CREATOR = 'TABLESPACE CREATOR' ;

DELETE FROM SYSTOOLS.ARYVIEWDEP WHERE BNAME = 'TABLE NAME'
  AND BCREATOR = 'TABLE CREATOR'
  AND DNAME = 'VIEW NAME'
  AND DCREATOR = 'VIEW CREATOR' ;

DELETE FROM SYSTOOLS.ARYVIEWS   WHERE NAME = 'VIEW NAME'
  AND CREATOR = 'VIEW CREATOR' ;
```

Example: C-4   When quiet time details are removed

```
//*****************************************************************************
//*  TABLES: ARYQT, ARYQTG
//*
//*  THE ARYQT AND ARYQTG TABLES ARE 'LINKED' BY GROUPID.
//*  YOU SHOULD THEREFORE DELETE FROM BOTH TABLES
//*
//*  FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
//*  SPECIFIC QUIET TIME REQUESTS WHILE USING RECOVERY EXPERT
//*
//*  => CHANGE THE 'N' FOR GROUPID TO AN APPROPRIATE GROUPID NUMBER
//*
//****************************************************************************
   DELETE FROM SYSTOOLS.ARYQT   WHERE GROUPID = N ;

   DELETE FROM SYSTOOLS.ARYQTG  WHERE GROUPID = N ;
```

Example: C-5   When stogroup or volume changes occur in DB2 for z/OS

```
//*****************************************************************************
//*  TABLES: ARYSTOGROUP, ARYVOLUMES
//*
//*  FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
//*  SPECIFIC ENVIRONMENT AND REQUIREMENTS
//*
//*  => CHANGE
//*       'STOGROUP NAME' TO THE NAME OF THE STOGROUP
//*       'CREATOR OF STOGROUP' TO THE NAME OF THE CREATOR OF STOGROUP
//*       'VCAT NAME' TO THE NAME OF THE HIGH LEVEL QUALIFIER / VCAT
//*       'VOLUME ID' TO THE NAME OF THE VOLUME OR * FOR SMS MANAGED
//*
//****************************************************************************

   DELETE FROM SYSTOOLS.ARYSTOGROUP WHERE NAME = 'STOGROUP NAME'
     AND CREATOR = 'CREATOR OF STOGROUP'
     AND VCATNAME = 'VCAT NAME';
```

```
   DELETE FROM SYSTOOLS.ARYVOLUMES  WHERE SGNAME = 'STOGROUP NAME'
     AND SGCREATOR = 'CREATOR OF STOGROUP'
     AND VOLID = 'VOLUME ID';
```

Example: C-6   Deleting rows in SLR table SRYVRLOG

```
//***************************************************************************
//*  SYSTOOLS TABLE:ARYVRLOG
//*
//*
//*  FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
//*  SPECIFIC ENVIRONMENT
//*
//*  => CHANGE
//*     'TIMESTAMP ASSOCIATED WITH ROW TO DELETE' TO TIMESTAMP DESIRED
//*     'JOB NAME OF SLR JOB RUN' TO JOB NAME ASSOCIATED WITH TIMESTAMP
//*
//***************************************************************************

   DELETE FROM SYSTOOLS.ARYVRLOG WHERE
      RUNDTS = 'TIMESTAMP ASSOCIATED WITH ROW TO DELETE'
      AND  JOBNAME = 'JOB NAME OF SLR JOB RUN' ;
//*
//*   DELETE FROM SYSTOOLS.ARYVRLOG WHERE
//*      RUNDTS = '2006-09-05-11.06.48.032638'
//*      AND  JOBNAME = 'W8SLR2' ;
```

Example: C-7   Deleting rows in ARYAUXRELS SLR table

```
//***************************************************************************
//*  TABLE: ARYAUXRELS
//*
//*  FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
//*  SPECIFIC ENVIRONMENT
//*
//*  => CHANGE
//*        'TBOWNER' TO THE AUXILLARY TABLE OWNER NAME
//*        'TABLENAME' TO THE AUXILLARY TABLE NAME
//*        'DBNAME' TO THE DATABASE NAME WHICH CONTAINS THE AUX TABLE
//*        'TSNAME' TO THE TABLESPACE NAME WHICH CONTAINS THE AUX TABLE
//*
//**************************************************************************
   DELETE FROM SYSTOOLS.ARYAUXRELS WHERE
      AUXTBOWNER = 'TBOWNER'
      AND  AUXTBNAME = 'TABLENAME'
      AND  AUXDBNAME = 'DBNAME'
      AND  AUXTSNAME = 'TSNAME' ;
```

Example: C-8   Deleting rows in ARYDATATYPES, ARYPARMS, ARYROUTINES, ARYVIEWSR SLR tables

```
//***************************************************************************
//* TABLES: ARYDATATYPES, ARYPARMS, ARYROUTINES, ARYVIEWSR
//*
//*  FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
//*  SPECIFIC ENVIRONMENT
//*
```

```
//*  => CHANGE
//*     'DATA TYPE OWNER' TO THE OWNER OF THE FUNCTION / DATA TYPE
//*     'FUNCTION OR DATA TYPE NAME' TO THE NAME OF FUNCTION / DATA TYPE
//*     'CREATOR OF THE FUNCTION OR DATA TYPE' TO CREATOR NAME
//*******************************************************************************

   DELETE FROM SYSTOOLS.ARYDATATYPES WHERE
      OWNER = 'DATA TYPE OWNER'
      AND  NAME = 'FUNCTION OR DATA TYPE NAME'
      AND  CREATEDBY = 'CREATOR OF THE FUNCTION OR DATA TYPE';

   DELETE FROM SYSTOOLS.ARYPARMS WHERE
      OWNER = 'DATA TYPE OWNER'
      AND  NAME = 'FUNCTION OR DATA TYPE NAME' ;

   DELETE FROM SYSTOOLS.ARYROUTINES WHERE
      OWNER = 'DATA TYPE OWNER'
      AND  NAME = 'FUNCTION OR DATA TYPE NAME'
      AND  CREATEDBY = 'CREATOR OF THE FUNCTION OR DATA TYPE';

   DELETE FROM SYSTOOLS.ARYVIEWSR    WHERE
      CREATOR = 'CREATOR OF THE FUNCTION OR DATA TYPE'
      AND  NAME = 'FUNCTION OR DATA TYPE NAME' ;
```

Example: C-9 Deleting row in ARYFIELDS SLR table

```
//*******************************************************************************
//* TABLES: ARYFIELDS
//*
//*  FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
//*  SPECIFIC ENVIRONMENT
//*
//*  => CHANGE
//*        'TABLE CREATOR' TO THE NAME OF THE TABLE CREATOR
//*        'TABLE NAME' TO THE NAME OF THE TABLE
//*        'FIELD PROC' TO THE NAME OF THE FIELD PROC
//*
//******************************************************************************

   DELETE FROM SYSTOOLS.ARYFIELDS  WHERE
      TBCREATOR = 'TABLE CREATOR'
      AND  TBNAME = 'TABLE NAME'
      AND  FLDPROC = 'FIELD PROC' ;
```

Example: C-10 Deleting DBRM, PLAN, and PACKAGES information in SLR tables

```
//*******************************************************************************
//* TABLES: (ALL DBRM, PLAN AND PACKAGES)
//*       ARYDBRM, ARYPACKAGE, ARYPACKAUTH, ARYPACKDEP, ARYPACKLIST, ARYPKSYSTEM,
//*       ARYPLAN, ARYPLANAUTH, ARYPLANDEP, ARYPLSYSTEM
//*
//*  THESE TABLES CONTAIN INFORMATION ABOUT DBRM'S, PLANS AND
//*  PACKAGES; WHEN YOU DELETE FROM ONE OF THE TABLES IT IS
//*  YOU SHOULD DELETE FROM ALL THESE SYSTOOLS TABLES
//*
//*  FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
```

```
//*  SPECIFIC ENVIRONMENT
//*
//*  => CHANGE
//*       'DBRM NAME' TO DBRM NAME
//*       'PDS WHERE DBRM RESIDES' TO NAME OF PDS WHERE DBRM RESIDES
//*       'PLAN NAME' TO THE NAME OF THE PLAN
//*       'PLAN CREATOR' TO THE NAME OF CREATOR OF THE PLAN
//*       'PACKAGE NAME' TO THE NAME OF PACKAGE
//*       'PACKAGE CREATOR' TO THE NAME OF CREATOR OF THE PACKAGE
//*       'PACKAGE QUALIFIER' TO THE NAME OF THE ASSIGNED PKG QUALIFIER
//*       'VERSION NAME ASSIGNED' TO THE NAME OF THE VERSION
//*       'GRANTED BY' IS THE GRANTOR OF THE AUTHORITY
//*       'PERSON GRANTED AUTH' TO THE PERSON GRANTED AUTHORITY
//*       'TABLE OR VIEW NAME' TO THE NAME OF THE TABLE OR VIEW
//*       'COLLECTION NAME' TO THE NAME OF THE COLLECTION
//*       'PLAN QUALIFIER' TO THE NAME OF THE ASSIGNED PLAN QUALIFIER
//*
//*******************************************************************************
   DELETE FROM SYSTOOLS.ARYDBRM  WHERE NAME = 'DBRM NAME'
     AND PDSNAME = 'PDS WHERE DBRM RESIDES'
     AND PLNAME = 'PLAN NAME'
     AND PLCREATOR = 'PLAN CREATOR' ;

    DELETE FROM SYSTOOLS.ARYPACKAGE WHERE NAME = 'PACKAGE NAME'
     AND OWNER = 'PACKAGE CREATOR' AND QUALIFIER = 'PACKAGE QUALIFIER'
     AND VERSION = 'VERSION NAME ASSIGNED'
     AND PDSNAME = 'PDS WHERE DBRM RESIDES' ;

   DELETE FROM SYSTOOLS.ARYPACKAUTH WHERE NAME = 'PACKAGE NAME'
     AND GRANTOR = 'GRANTED BY'
     AND GRANTEE = 'PERSON GRANTED AUTH'
     AND COLLIDX = 'COLLECTION NAME';

   DELETE FROM SYSTOOLS.ARYPACKDEP  WHERE BNAME = 'TABLE OR VIEW NAME'
     AND BQUALIFIER = 'PACKAGE QUALIFIER'
     AND DCOLLID = 'COLLECTION NAME';
     AND DOWNER = 'PACKAGE CREATOR' ;

   DELETE FROM SYSTOOLS.ARYPACKLIST  WHERE
     PLANNAME = 'PLAN NAME'
     AND COLLIDX = 'COLLECTION NAME';

   DELETE FROM SYSTOOLS.ARYPKSYSTEM  WHERE
     NAME = 'PACKAGE NAME'
     AND COLLIDX = 'COLLECTION NAME';

   DELETE FROM SYSTOOLS.ARYPLAN WHERE NAME = 'PLAN NAME'
     AND CREATOR = 'PLAN CREATOR' AND QUALIFIER = 'PLAN QUALIFIER' ;

   DELETE FROM SYSTOOLS.ARYPLANAUTH WHERE NAME = 'PLAN NAME'
     AND GRANTOR = 'GRANTED BY'
     AND GRANTEE = 'PERSON GRANTED AUTH';
```

```
   DELETE FROM SYSTOOLS.ARYPLANDEP  WHERE BNAME = 'PLAN NAME'
     AND BCREATOR = 'PLAN CREATOR' ;

   DELETE FROM SYSTOOLS.ARYPLSYSTEM  WHERE
     NAME = 'PLAN NAME' ;
```

Example: C-11 Deleting referential constraint information in SLR tables

```
//*****************************************************************************
//* TABLES: (CHECK CONSTRAINTS AND REFERENTIAL RELATIONSHIPS)
//*       ARYCHECKS, ARYCHECKS2, ARYFOREIGNKEYS, ARYRELS
//*  THESE TABLES CONTAIN INFORMATION ABOUT CHECK CONSTRAINTS
//*  AND REFERENTIAL INTEGRITY RELATIONSHIPS.
//*  WHEN A CONSTRAINT OR REFERENTIAL RELATIONSHIP IS REMOVED
//*  YOU SHOULD DELETE FROM ALL THESE SYSTOOLS TABLES
//*
//*  FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
//*  SPECIFIC ENVIRONMENT
//*
//*  => CHANGE
//*       'TABLE NAME' TO TABLE NAME WITH CONSTRAINT
//*       'OWNER NAME' TO TABLE OWNER NAME WITH THE CONSTRAINT
//*       'CONSTRAINT NAME' TO THE NAME OF THE CONSTRAINT
//*       'RI CREATOR' TO THE NAME OF THE CREATOR OF THE RI
//*       'RELATIONSHIP NAME' TO NAME OF FOREIGN KEY
//*
//*****************************************************************************

   DELETE FROM SYSTOOLS.ARYCHECKS  WHERE TBNAME = 'TABLE NAME'
     AND TBOWNER = 'OWNER NAME' AND CHECKNAME = 'CONSTRAINT NAME' ;

   DELETE FROM SYSTOOLS.ARYCHECKS2 WHERE TBNAME = 'TABLE NAME'
     AND TBOWNER = 'OWNER NAME' AND CHECKNAME = 'CONSTRAINT NAME' ;

   DELETE FROM SYSTOOLS.ARYFOREIGNKEYS WHERE CREATOR = 'RI CREATOR'
     AND TBNAME = 'TABLE NAME' AND RELNAME = 'RELATIONSHIP NAME';

   DELETE FROM SYSTOOLS.ARYRELS  WHERE CREATOR = 'RI CREATOR'
     AND TBNAME = 'TABLE NAME' AND RELNAME = 'RELATIONSHIP NAME';
```

Example: C-12 Deleting specifications in SLR tables

```
//*****************************************************************************
//*  TABLES: ARYSPEC_DATA, ARYSPEC_DIR
//*
//*  THE ARYSPEC_DATA AND ARYSPEC_DIR TABLES CONTAIN INFORMATION
//*  ABOUT SAVED SPECIFICATIONS.
//*  YOU SHOULD DELETE FROM BOTH TABLES
//*
//*  FOLLOWING SAMPLE DELETE STATEMENT SHOULD BE MODIFIED BASED ON YOUR
//*  SPECIFIC ENVIRONMENT
//*  => CHANGE
//*        'SPECOWNER' TO THE SPECIFICATION OWNER NAME
//*        'SPECNAME'  TO THE SAVED SPECIFICATION NAME
//*
```

```
//******************************************************************************
   DELETE FROM SYSTOOLS.ARYSPEC_DATA WHERE OWNER = 'SPECOWNER'
      AND NAME = 'SPECNAME' ;

   DELETE FROM SYSTOOLS.ARYSPEC_DIR WHERE OWNER = 'SPECOWNER'
      AND NAME = 'SPECNAME' ;
```

D

Potential Recovery Expert for z/OS recovery plans

In this appendix, we provide a list of the commonly generated recovery plans by Recovery Expert for z/OS, and the possible causes of a failure to generate a recovery plan.

D.1 Introduction

When you request Recovery Expert for z/OS to generate a recovery plan for one or more objects that you have identified in the Recovery Expert for z/OS graphical user interface (GUI) client, it considers the various resources of recovery available such as full and incremental image copies[1], and active and archive logs, and generates a number of potential recovery plans that can each individually perform the recovery action that is requested. These recovery plans are arranged in ascending sequence of efficiency as determined by Recovery Expert for z/OS. The number of recovery plans varies depending upon the required recovery action and the available recovery resources. In some cases, Recovery Expert for z/OS determines that adequate recovery resources are not available to generate any recovery plan.

Recovery Expert for z/OS uses the following tools to perform the recovery action in its recovery plans:

- RECOVER, REBUILD INDEX, LOAD, CHECK DATA, and DSN1COPY are DB2 utilities.
- RESTORE DATASET is a DFSMSdss function command.
- Log analysis is included with DB2 Recovery Expert for z/OS to generate redo and undo Structured Query Language (SQL).

A recovery plan includes multiple recovery methods that describe the following features:

- An initial (first) utility such as DSN1COPY that is used to acquire the beginning data
- The resource for the data that is used such as an image copy
- A second utility, if necessary, that is required to modify the data such as a RECOVER utility
- The direction of log processing that is required to modify the data, redo for forward processing, undo for backward processing
- The location of the starting point for the log range
- The location of the point-in-time (PIT) that you specified

In the following sections, we list some of the most frequent recovery plans considered by Recovery Expert for z/OS, and some of the possible causes for failing to generate any recovery plan for a requested recovery action for one or more DB2 objects.

[1] Image copies are recorded in the DB2 System Catalog (DSC) and optionally in the schema level repository (SLR) that is part of Recovery Expert for z/OS resources. Logs are recorded only in the bootstrap data set (BSDS). Regular SLR updates preserve information that is otherwise lost from the DSC when objects are dropped or altered.

D.2 Potential recovery plans generated

Figure D-1 lists some of the more frequent recovery plans considered by Recovery Expert for z/OS when requested to perform a particular recovery action.

First utility	Data resource	Second utility	Log	Starting location	PIT
DSN1COPY	Last Image Copy				= Last IC
DSN1COPY	Last IC	RECOVER	Redo	Last IC	> Last IC
DSN1COPY	Last IC	Log Analysis	Redo	Last IC	> Last IC
DSN1COPY	Last IC	Log Analysis	Undo	Last IC	< Last IC
DSN1COPY	Full IC	Log Analysis	Undo	Full IC	< Full IC
RECOVER	Last IC				= Last IC
RECOVER	Last IC		Redo	Last IC	> Last IC
RECOVER	Last IC	Log Analysis	Redo	Last IC	> Last IC
RECOVER	Last IC	Log Analysis	Undo	Last IC	< Last IC
RECOVER	Full IC	Log Analysis	Undo	Full IC	< Full IC
LOAD	Full IC				= Full IC
LOAD	Full IC	Log Analysis	Redo	Full IC	> Full IC
LOAD	Full IC	Log Analysis	Undo	Full IC	< Full IC
Log Analysis			Undo	Current	< Current
REBUILD					= Current
RESTORE	Full IC				= Full IC
RESTORE	Full IC	RECOVER	Redo	Full IC	> Full IC
RESTORE	Full IC	Log Analysis	Redo	Full IC	> Full IC
RESTORE	Full IC	Log Analysis	Undo	Full IC	< Full IC

Figure D-1 Some possible recovery plans

In Figure D-1, the highlighted entry shows a recovery action that is requested to a point beyond the last available image copy; this can be to a relative byte address (RBA) or log record sequence number (LRSN). The highlighted recovery plan involves using DSN1COPY to restore an image copy, followed by an execution of the RECOVER utility with a LOGONLY option to an RBA or LRSN or end of log.

The entry that follows the highlighted one has the same recovery action requested (such as to an RBA or LRSN or end of log), which has a different set of recovery methods. In this case, the DSN1COPY restore of the last image copy is followed by a log analysis generation of redo SQL, which is then applied to the target object.

Attention: This is not a comprehensive list of all the possible recovery methods, but it clearly indicates that Recovery Expert for z/OS requires a full image copy to be available in *all* its recovery method cases (see the data resource column in Figure D-1), except for the recovery actions using log analysis to perform an undo to a point-in-time before current, and REBUILD INDEX as highlighted.

The recovery PIT requested always affects which methods are relevant for each object, and is usually compared with the most recent image copies both before and after the PIT. If the last image copy is incremental, then a full image copy before the PIT is required.

The log range required for a recovery method cannot contain objects with any of the following features, because the log does not contain the changes required for recovery:

- CHECK DATA with LOG(NO)
- LOAD with LOG(NO)
- REORG with LOG(NO)
- RECOVER with TOCOPY or TOLOGPOINT

Some of the constraints related to recovery plan generation are:

- Recreated tables are not supported by RECOVER, RESTORE, or undo SQL from current
- DSN1COPY does not support user-managed spaces defined as NOREUSE
- Image copies created with the CONCURRENT option are supported only by RECOVER and RESTORE
- Image copies created without the CONCURRENT option are supported by every utility except RESTORE
- RECOVER might not require a last image copy if the object is newer than the oldest recorded log, otherwise the full image copy must be recorded in the DSC
- LOAD uses only full image copies to avoid duplicate rows from partially updated pages on incremental copies
- Log analysis does not support tables with:
  - Columns using distinct data types
  - Columns defined with GENERATED ALWAYS, which is the default for the ROWID data type
  - Columns defined with BLOBs, CLOBs, or DBCLOBs
  - Column names, table name, or creator that cannot be translated into EBCDIC (CCSID 37)
- REBUILD is used as the first utility only when the parent table is not being recovered, and only when recovering to current
- The utilities used for recovery are usually either unlogged or limit log processing. Frequent image copies help to ensure a recoverable environment.

D.3 Possible cause of recovery plan generation failure

Sometimes Recovery Expert for z/OS is unable to generate any recovery plan for the requested recovery action, and issues one of the following error messages:

- ARYZ017E No objects to recover were found

  When this message is received, you must ensure that you have selected objects that can be recovered. Certain object types such as plans, packages, grouper sets, and object profiles are used only to select other recoverable object types. These object types are never directly recovered. This type of indirect selection can result in "no objects to recover".

  Common issues with indirect selection include:

  – A plan that uses only a pack list that has no direct dependencies
  – A storage group that is not used by any table space partitions or index partitions

- ARYZ018E The following objects are unrecoverable: object-list

  There are many possible reasons that can cause Recovery Expert for z/OS to generate this message. To resolve this situation, you must have a good understanding of what Recovery Expert for z/OS is attempting to do, and determine the factor that is preventing Recovery Expert for z/OS from generating a recovery plan.

E

Configuration files used in the user scenarios

In this appendix, we provide the complete contents of the configuration files that are used in the user scenarios.

E.1 Introduction

The complete contents of the following configuration files are used in the scenarios:

- Recovery Expert for z/OS server configuration file on logical partition SC53 (LPAR SC53), as shown in Example E-1 on page 328
- Recovery Expert for z/OS server configuration file on LPAR SC59, as shown in Example E-2 on page 332
- Recovery Expert for z/OS Agent 53 agent configuration file on all the LPARs, as shown in Example E-3 on page 335; Recovery Expert for z/OS Agent 53 on all LPARs references Recovery Expert for z/OS Server 53 using an agent configuration file that identifies the host name (wtsc53.itso.ibm.com) and port number 9876 (default).
- Shared Recovery Expert for z/OS Agent 53 product control file on LPARs SC53 and SC67, as shown in Example E-4 on page 340; it identifies subsystems D8G1, D8G2, D8F1, and D8F2.
- Recovery Expert for z/OS Agent 53 product control file on LPAR SC59, as shown in Example E-5 on page 356; it identifies only subsystem DB8A.
- Recovery Expert for z/OS Agent 59 agent configuration file on all the LPARs, as shown in Example E-6 on page 361; Recovery Expert for z/OS Agent 59 on all LPARs references Recovery Expert for z/OS Server 59 using an agent configuration file that identifies the host name (wtsc59.itso.ibm.com) and port number 9876 (default). Recovery Expert for z/OS Agent 59 on LPARs SC53 and SC67 shares the *same* IBM Recovery Expert for z/OS product control file used by Recovery Expert for z/OS Agent 53.
- Shared Recovery Expert for z/OS Agent 59 product control file on LPARs SC53 and SC67, as shown in Example E-7 on page 365; it identifies subsystems D8G1, D8G2, D8F1, and D8F2.
- Recovery Expert for z/OS Agent 59 product control file on LPAR SC59, as shown in Example E-8 on page 382; it identifies only subsystem DB8W.

Example: E-1 Recovery Expert for z/OS server configuration file on LPAR SC53

```
<!--*****************************************************************
5697-H74
(c) Copyright Rocket Software, Inc. 2003, 2006 All Rights Reserved.
*****************************************************************-->

<!--*****************************************************************
This is a sample configuration file for the IBM DB2 Recovery Expert
for z/OS 1.1.0 server.

A copy of this file should be made, and customized as described below.
The job to run the server must include a DD definition referring to the
customized file.  (See sample member ARYSJSRV.)

The default values for all parameters may be used.

This file must have valid XML syntax.
*****************************************************************-->

<server-config>
```

```
<!--******************************************************************
  client-listener-port must specify the IP port on which the server
  listens for connections from IBM DB2 Recovery Expert clients.

  Valid values are integers between 1 and 65535, inclusive.
*******************************************************************-->

<client-listener-port>9875</client-listener-port>


<!--******************************************************************
  agent-listener-port must specify the IP port on which the server
  listens for connections from IBM DB2 Recovery Expert agents.

  Valid values are integers between 1 and 65535, inclusive.
*******************************************************************-->

<agent-listener-port>9876</agent-listener-port>


<!--******************************************************************
  description is optional, and specifies a descriptive string for the
  instance of the server.  This description is displayed to client
  users when they browse from available servers to which to connect.
*******************************************************************-->

<description></description>


<!--******************************************************************
  community-string is optional, and specifies an identifying string
  for the instance of the server.  In order for agents to discover
  and connect to this server, those agents must be configured with
  the same community-string value.
*******************************************************************-->

<community-string>wtsc53</community-string>


<!--******************************************************************
  multicast-address is optional, and specifies the UDP multicast
  address on which the server should make server announcements.

  In order for agents and clients to discover this server, they must
  be configured with the same multicast-address value.

  Valid values are IP addresses in dotted-decimal notation, in the
  range from 224.0.1.0 to 238.255.255.255, inclusive.
*******************************************************************-->

<multicast-address>236.1.2.3</multicast-address>
```

```
<!--*****************************************************************
  multicast-port is optional, and specifies the UDP multicast
  port on which the server should make server announcements.

  In order for agents and clients to discover this server, they must
  be configured with the same multicast-port value.

  Valid values are integers between 1 and 65535, inclusive.
*****************************************************************-->

<multicast-port>19875</multicast-port>


<!--*****************************************************************
  multicast-interface is optional, and specifies the local network
  interface address on which the server should make server
  announcements.  If omitted, the server makes the announcements on
  all interfaces.

  Valid values are IP addresses in dotted-decimal notation.
*****************************************************************-->

<multicast-interface></multicast-interface>


<!--*****************************************************************
  multicast-ttl is optional, and specifies the UDP multicast
  "time-to-live" value for the server announcements.  This value
  specifies the maximum number of subnets over which the announcements
  will be routed.  Consult your network configuration documentation
  for more information.

  Valid values are integers greater than 0.
*****************************************************************-->

<multicast-ttl>5</multicast-ttl>


<!--*****************************************************************
  multicast-delay is optional, and specifies the number of seconds the
  server should wait between making announcements of its presence on
  the network.  Smaller values result in more network traffic, but
  better responsiveness to agents and clients attempting to discover
  this server.

  Valid values are integers greater than 0.
*****************************************************************-->

<multicast-delay>5</multicast-delay>


<!--*****************************************************************
  log-level controls the amount of output log information that is
  generated by the server.
```

```
  Valid values are:
  0 - disable logging
  S - log severe error messages only
  E - log error and severe error messages
  W - log warning, error, and severe error messages
  I - log information, warning error, and severe error messages
          (recommended)
*******************************************************************-->

<log-level>I</log-level>

<!--*****************************************************************
  bind-retry-max specifies the maximum number of
  attempts the server should make to bind to thse specified client
  and agent listener ports, before exiting with an error.

  Typically, there is no delay in binding.  However, it is possible
  that one or both of the ports might be in use by another
  application.  Also, if a previous application was using one or both
  of the ports and failed, there may be a delay before the system
  releases the port(s) so that they can be used by the DB2 Recovery
  Expert server.

  Valid values are integers greater than or equal to 0.
*******************************************************************-->

<bind-retry-max>30</bind-retry-max>


<!--*****************************************************************
  bind-retry-delay specifies the number of seconds the
  server should wait between attempts to bind to the client and
  agent listener ports.

  The period of time (in seconds) that the server will continue
  attempts to bind is:
      bind-retry-max * bind-retry-delay

  Valid values are integers greater than 0.
*******************************************************************-->

<bind-retry-delay>10</bind-retry-delay>


<!--*****************************************************************
  The trace-* parameters cause additional tracing information to be
  logged, in order to diagnose errors that may occur during product
  execution.

  These parameters should not be enabled unless directed by product
  support, as there is a significant cost in performance when they
  are enabled.

  Valid values for each are true or false.
*******************************************************************-->
```

```
<trace-network>false</trace-network>
  <trace-xml>false</trace-xml>
  <trace-events>false</trace-events>
  <trace-config>false</trace-config>

</server-config>
```

Example: E-2 Recovery Expert for z/OS server configuration file on LPAR SC59

```
<!--*******************************************************************
5697-H74
(c) Copyright Rocket Software, Inc. 2003, 2006 All Rights Reserved.
*******************************************************************-->

<!--*******************************************************************
This is a sample configuration file for the IBM DB2 Recovery Expert
for z/OS 1.1.0 server.

A copy of this file should be made, and customized as described below.
The job to run the server must include a DD definition referring to the
customized file.  (See sample member ARYSJSRV.)

The default values for all parameters may be used.

This file must have valid XML syntax.
*******************************************************************-->

<server-config>


  <!--*****************************************************************
    client-listener-port must specify the IP port on which the server
    listens for connections from IBM DB2 Recovery Expert clients.

    Valid values are integers between 1 and 65535, inclusive.
  *****************************************************************-->

  <client-listener-port>9875</client-listener-port>


  <!--*****************************************************************
    agent-listener-port must specify the IP port on which the server
    listens for connections from IBM DB2 Recovery Expert agents.

    Valid values are integers between 1 and 65535, inclusive.
  *****************************************************************-->

  <agent-listener-port>9876</agent-listener-port>


  <!--*****************************************************************
    description is optional, and specifies a descriptive string for the
    instance of the server.  This description is displayed to client
    users when they browse from available servers to which to connect.
  *****************************************************************-->
```

```
<description></description>

  <!--*******************************************************************
    community-string is optional, and specifies an identifying string
    for the instance of the server.  In order for agents to discover
    and connect to this server, those agents must be configured with
    the same community-string value.
  *****************************************************************-->

  <community-string>wtsc59</community-string>


  <!--*******************************************************************
    multicast-address is optional, and specifies the UDP multicast
    address on which the server should make server announcements.

    In order for agents and clients to discover this server, they must
    be configured with the same multicast-address value.

    Valid values are IP addresses in dotted-decimal notation, in the
    range from 224.0.1.0 to 238.255.255.255, inclusive.
  *****************************************************************-->

  <multicast-address>236.1.2.3</multicast-address>


  <!--*******************************************************************
    multicast-port is optional, and specifies the UDP multicast
    port on which the server should make server announcements.

    In order for agents and clients to discover this server, they must
    be configured with the same multicast-port value.

    Valid values are integers between 1 and 65535, inclusive.
  *****************************************************************-->

  <multicast-port>19875</multicast-port>


  <!--*******************************************************************
    multicast-interface is optional, and specifies the local network
    interface address on which the server should make server
    announcements.  If omitted, the server makes the announcements on
    all interfaces.

    Valid values are IP addresses in dotted-decimal notation.
  *****************************************************************-->

  <multicast-interface></multicast-interface>


  <!--*******************************************************************
    multicast-ttl is optional, and specifies the UDP multicast
    "time-to-live" value for the server announcements.  This value
```

```
  specifies the maximum number of subnets over which the announcements
  will be routed.  Consult your network configuration documentation
  for more information.

  Valid values are integers greater than 0.
*******************************************************************-->

<multicast-ttl>5</multicast-ttl>


<!--*****************************************************************
  multicast-delay is optional, and specifies the number of seconds the
  server should wait between making announcements of its presence on
  the network.  Smaller values result in more network traffic, but
  better responsiveness to agents and clients attempting to discover
  this server.

  Valid values are integers greater than 0.
*******************************************************************-->

<multicast-delay>5</multicast-delay>


<!--*****************************************************************
  log-level controls the amount of output log information that is
  generated by the server.

  Valid values are:
  O - disable logging
  S - log severe error messages only
  E - log error and severe error messages
  W - log warning, error, and severe error messages
  I - log information, warning error, and severe error messages
        (recommended)
*******************************************************************-->

<log-level>I</log-level>

<!--*****************************************************************
  bind-retry-max specifies the maximum number of
  attempts the server should make to bind to thse specified client
  and agent listener ports, before exiting with an error.

  Typically, there is no delay in binding.  However, it is possible
  that one or both of the ports might be in use by another
  application.  Also, if a previous application was using one or both
  of the ports and failed, there may be a delay before the system
  releases the port(s) so that they can be used by the DB2 Recovery
  Expert server.

  Valid values are integers greater than or equal to 0.
*******************************************************************-->

<bind-retry-max>30</bind-retry-max>
```

```
  <!--******************************************************************
    bind-retry-delay specifies the number of seconds the
    server should wait between attempts to bind to the client and
    agent listener ports.

    The period of time (in seconds) that the server will continue
    attempts to bind is:
        bind-retry-max * bind-retry-delay

    Valid values are integers greater than 0.
  *******************************************************************-->

  <bind-retry-delay>10</bind-retry-delay>


  <!--******************************************************************
    The trace-* parameters cause additional tracing information to be
    logged, in order to diagnose errors that may occur during product
    execution.

    These parameters should not be enabled unless directed by product
    support, as there is a significant cost in performance when they
    are enabled.

    Valid values for each are true or false.
  *******************************************************************-->

  <trace-network>false</trace-network>
  <trace-xml>false</trace-xml>
  <trace-events>false</trace-events>
  <trace-config>false</trace-config>

</server-config>
```

Example: E-3 Recovery Expert for z/OS Agent 53 agent configuration file on all the LPARs

```
<!--******************************************************************
5697-H74
(c) Copyright Rocket Software, Inc. 2003, 2006 All Rights Reserved.
*******************************************************************-->

<!--******************************************************************
This is a sample configuration file for the IBM DB2 Recovery Expert
for z/OS 1.1.0 agent.

A copy of this file should be made, and customized as described below.
The job to run the agent must include a DD definition referring to the
customized file.  (See sample member ARYSJAGT.)

The default values for all parameters may be used, except for
<server-address> and <server-port>, which must be properly
configured to connect to the DB2 Recovery Expert server.

This file must have valid XML syntax.
*******************************************************************-->
```

```
<agent-config>

  <!--****************************************************************
    server-address must specify the host name or IP address (in
    dotted-decimal notation, e.g., 1.2.3.4) of the DB2 Recovery
    Expert server to which the agent should connect.  If server-address
    is omitted, the agent will attempt to automatically discover the
    address and port number of a server.  (See below for additional
    server discovery parameters.)
  ****************************************************************-->

   <server-address>wtsc53.itso.ibm.com</server-address>


  <!--****************************************************************
    server-port must specify the IP port number on the DB2 Recovery
    Expert server to which the agent should connect.

    Valid values are integers between 1 and 65535, inclusive.
  ****************************************************************-->

  <server-port>9876</server-port>


  <!--****************************************************************
    community-string is optional, and specifies the identification for
    which the agent listens when performing server discovery.  In order
    for the agent to discover and connect to a server, that server must
    be configured with the same community-string value.
  ****************************************************************-->

  <community-string>wtcs53</community-string>


  <!--****************************************************************
    multicast-address is optional, and specifies the UDP multicast
    address on which the agent should listen for server announcements.
    In order for the agent to discover and connect to a server, that
    server must be configured with the same multicast-address value.

    Server discovery is performed only if the server-address parameter
    is omitted.

    Valid values are IP addresses in dotted-decimal notation, in the
    range from 224.0.1.0 to 238.255.255.255, inclusive.
  ****************************************************************-->

  <multicast-address>236.1.2.3</multicast-address>


  <!--****************************************************************
    multicast-port is optional, and specifies the UDP multicast
    port on which the agent should listen for server announcements.
    In order for the agent to discover and connect to a server, that
    server must be configured with the same multicast-address value.
```

```
  Server discovery is performed only if the server-address parameter
  is omitted.

  Valid values are integers between 1 and 65535, inclusive.
*******************************************************************-->

<multicast-port>19875</multicast-port>


<!--*****************************************************************
  control-file-dd specifies the DD name allocated to the product
  control file.  The name specified must be allocated in the JCL used
  to run the agent.  The default value is DB2PARMS.

  Valid values are strings conforming to DD name syntax rules.
*******************************************************************-->

<control-file-dd>DB2PARMS</control-file-dd>


<!--*****************************************************************
  log-level controls the amount of output log information that is
  generated by the agent.

  Valid values are:
  O - disable logging
  S - log severe error messages only
  E - log error and severe error messages
  W - log warning, error, and severe error messages
  I - log information, warning error, and severe error messages
          (recommended)
*******************************************************************-->

<log-level>I</log-level>


<!--*****************************************************************
  server-connect-retry-max specifies the maximum number of
  attempts the agent should make to connect to the server, before
  exiting with an error.

  Typically, the server should be started and available before any
  agent is started, in which case the agent will immediately connect
  on the first attempt.  This parameter allows for the case when the
  server is not immediately available.  It also allows for the case
  where the server terminates for any reason.  In this case, the
  agent will cancel any pending requests, and attempt to reconnect
  to the server (presumably, after the server is restarted).

  Valid values are:
  0 - continue attempts to connect indefinitely
  integers greater than 0 - the maximum number of connection attempts
*******************************************************************-->

<server-connect-retry-max>30</server-connect-retry-max>
```

```
<!--*****************************************************************
  server-connect-retry-delay specifies the number of seconds the
  agent should wait between attempts to connect to the server.

  The period of time (in seconds) that the agent will continue
  attempts to connect is:
      server-connect-retry-max * server-connect-retry-delay

  Valid values are integers greater than 0.
*****************************************************************-->

<server-connect-retry-delay>10</server-connect-retry-delay>


<!--*****************************************************************
  request-thread-timeout specifies the number of seconds a
  thread/task created to do work for a specific user should remain
  idle before exiting.  Setting this value higher provides a better
  response to client requests, but consumes more resources (in the
  form of extra tasks that are not performing work).

  This value should be set high enough so that a task does not exit
  during a typical end-user client session, i.e., greater than the
  expected time between end-user actions in the client.

  Valid values are integers greater than 0.
*****************************************************************-->

<request-thread-timeout>300</request-thread-timeout>


<!--*****************************************************************
  uppercase-passwords specifies whether or not user IDs and
  passwords specified by end users should be folded to uppercase
  before using them to authenticate the user.

  Valid values are:
  true - uppercase user IDs and passwords
  false - use user IDs and passwords as entered by the user
*****************************************************************-->

<uppercase-passwords>true</uppercase-passwords>


<!--*****************************************************************
  job-poll-rate specifies the number of seconds the agent should wait
  before attempts to query the status of submitted jobs.  Lower values
  provide better response time to end users, but require more
  resources on the server.

  Valid values are integers greater than 0.
*****************************************************************-->

<job-poll-rate>5</job-poll-rate>
```

```
<!--******************************************************************
    job-cancel-timeout specifies the number of seconds the agent should
    wait after cancelling a job for the job to end.  If the job does not
    end within this time period, the agent "abandons" the job, allowing
    it to continue running, and reporting an error to the server.

    Valid values are integers greater than 0.
  ******************************************************************-->

  <job-cancel-timeout>5</job-cancel-timeout>


  <!--******************************************************************
    check-ownership-external specifies whether or not an external
    security manager (such as RACF) is used to manage DB2 security.
    If set to false, the agent will evaluate authorizations recorded in
    the DB2 catalog in order to determine whether or not a given user
    has ownership rights on a specification.  If an external security
    manager is used, this process is not sufficient to determine
    ownership.  If set to true, the agent will additionally attempt to
    SET CURRENT SQLID to the specification owner to determine if a user
    has ownership rights.  This will work, but may result in spurious
    errors with SQL code -553, which may be recorded in DB2 audit
    traces.  The default is true.

    Valid values are true or false.
  ******************************************************************-->

  <check-ownership-external>true</check-ownership-external>


  <!--******************************************************************
    The trace-* parameters cause additional tracing information to be
    logged, in order to diagnose errors that may occur during product
    execution.

    These parameters should not be enabled unless directed by product
    support, as there is a significant cost in performance when they
    are enabled.

    Valid values for each are true or false.
  ******************************************************************-->

  <trace-csi>false</trace-csi>
  <trace-db2-attachment>false</trace-db2-attachment>
  <trace-sql>false</trace-sql>
  <trace-ifi>false</trace-ifi>
  <trace-network>false</trace-network>
  <trace-xml>false</trace-xml>
  <trace-events>false</trace-events>
  <trace-config>false</trace-config>

</agent-config>
```

Example: E-4 Shared Recovery Expert for z/OS Agent 53 product control file on LPARs SC53 and SC67

```
//ARYSJ002 JOB (999,POK),'CONWAY',CLASS=A,MSGCLASS=H,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
/*JOBPARM L=999,SYSAFF=SC53
//*******************************************************************
//*
//* 5697-H74
//* (c) Copyright Rocket Software, Inc. 2003 - 2006 All Rights
//* Reserved.
//*
//*******************************************************************
//* IBM DB2 Recovery Expert for z/OS 1.1.0
//*
//* THIS JCL WILL LOAD THE CONTROL FILE AS DEFINED IN SAMPLIB MEMBER
//* ARYSJ000.
//*
//* CAUTION: THIS IS NEITHER A JCL PROCEDURE NOR A COMPLETE JOB.
//*
//* Before running this job, you will have to make the following
//* modifications:
//*
//* 1. Change the job statement to meet your system requirements
//*
//* 2. Change 'ARY.V0110.CONTROL' below to the name of the VSAM
//*    control data set created in member ARYSJ000.
//*
//* 3. Change ARY.V1R1M0.SARYLOAD below to the name of the product
//*    LOADLIB used for IBM DB2 Recovery Expert Z/OS Tool.
//*
//* 4. Change ARY.V1R1M0.SARYMLIB below to the name of the product
//*    message library used for IBM DB2 Recovery Expert Z/OS Tool.
//*
//* 5. Look at the SYSIN DD * statements for the job. You must change
//*    all parameters prefixed with an '&'. At the bottom of this JCL
//*    is a sample SYSIN statement to help clarify the modifications
//*    needed. It shows the necessary formats for these parameters
//*    as well. Any number of subsystems may be initially defined.
//*    New information will be added and existing information will
//*    be updated.
//*
//*    -any optional statements can be commented out or deleted. To
//*     comment out, simply put a '*' in Column 1 of the control
//*     statement.
//*
//*    -the 'SET PRODUCT CFG' AND 'SET PRODUCT VER' control statements
//*     are used when defining multiple product configurations to
//*     support the execution of multiple versions of the product and
//*     multiple copies of the schema level repository on a DB2
//*     subsystem. Refer to the product user guide for information on
//*     how to use these control statements. If multiple product
//*     execution configurations are not required then remove or
//*     comment the 'SET PRODUCT' control statements or provide the
//*     value of NULL.
//*-----------------------------------------------------------------*
//*
```

```
//*
//PCFUPD1  EXEC PGM=ARY#UTIL,PARM='SETUP',REGION=4M
//STEPLIB  DD   DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//SYSUDUMP DD   SYSOUT=*
//DB2PARMS DD   DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//*
//* REPORTS
//*
//SYSOUT   DD   SYSOUT=*,RECFM=FBA,LRECL=133    SYSIN REPORT
//SYSPRINT DD   SYSOUT=*,RECFM=FBA,LRECL=133    PCF REPORT
//*
//* CONTROLS
//*
//SYSIN    DD   *
*
*-------------------------------------------------------------------
* Sample statements to add/update GDG model data set name.
*-------------------------------------------------------------------
*
* UPDATE GDG MODEL       = &gdg_model_dsname
*
*-------------------------------------------------------------------
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*-------------------------------------------------------------------
*
 SET DB2 SSID           = D8G1
 UPDATE DB2 ZPARMS      = DSNZPAG1
 UPDATE DB2 BOOTSTRAP1  = DB8GU.D8G1.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8GU.D8G1.BSDS02
 UPDATE DB2 LOADLIB1    = DB8G8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8G8.SDSNLOAD
*UPDATE DB2 LOADLIB3    =
*UPDATE DB2 LOADLIB4    =
*UPDATE DB2 LOADLIB5    =
*
*-------------------------------------------------------------------
* Sample statements to add/update ARY product plans
*-------------------------------------------------------------------
*
 SET DB2 SSID           = D8G1
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
*
*
*-------------------------------------------------------------------
* Sample statements to add/update product message library
*-------------------------------------------------------------------
```

```
*
 UPDATE ARY MSGLIBRARY  = ARY.V1R1M0.SARYMENU
*
*--------------------------------------------------------------------
* Sample statements to add/update log services options
*--------------------------------------------------------------------
*
 UPDATE ARY ARCHLOG1    = Y       USE ARCHIVE LOG 1
 UPDATE ARY ARCHLOG2    = N       USE ARCHIVE LOG 2
 UPDATE ARY ACTLOGPRI   = Y       ACTIVE LOG PRIORITY
*
*--------------------------------------------------------------------
* Sample statements to add/update data set prefix generation
*--------------------------------------------------------------------
*
*  The DSN PREFIX maximum length is 17 characters. If NULL
*  is specified then user id is used as data set prefix. Use &USERID
*  in the prefix to insert user id. Example: TEST.&USERID will
*  generate a data set prefix of 'TEST.MYID' where the user id is
*  'MYID'.
*
 UPDATE ARY DSN PREFIX  = RETOOLS
*
*--------------------------------------------------------------------
*
*--------------------------------------------------------------------
* Sample statements to add/update schema level repository data
* capture options.
*--------------------------------------------------------------------
*
 UPDATE SLR LOAD BINDS  = Y         N = package/plan versions are
*                                       not saved in schema level
*                                       repository.
 UPDATE SLR LOAD AUTHS  = Y         N = DB2 authorizations are
*                                       not saved in schema level
*                                       repository.
*
*--------------------------------------------------------------------
* Sample statements to add/update interproduct communication
* options.
*--------------------------------------------------------------------
*
 UPDATE IPC IPC_GROUPER = N         Y = Enable Grouper-related
*                                       table recovery.
*
*--------------------------------------------------------------------
* Sample statements to add/update table activity quiet time
* repository names.
*--------------------------------------------------------------------
*
*  EACH QT OWNER/NAME IS A 45 CHAR MAXIMUM LENGTH. THESE OBJECTS WILL
*  BE CREATED AUTOMATICALLY WHEN THE QUIET TIME REPORT/CAPTURE JCL
*  IS RUN IF THEY DO NOT ALREADY EXIST. DDL TO CREATE THESE OBJECTS
*  IS PROVIDED IN ARYDDL7 AND ARYDDL8 SAMPLE DDL MEMBERS.
*
```

```
*                         XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 UPDATE QT GRP TBOWNER  = SYSTOOLS
 UPDATE QT GRP TBNAME   = ARYQTG
 UPDATE QT GRP IXOWNER  = SYSTOOLS
 UPDATE QT GRP IXNAME   = ARYQTGX
 UPDATE QT ENTRY OWNER  = SYSTOOLS
 UPDATE QT ENTRY NAME   = ARYQT
*                         XXXXXXXX (MAX LENGTH FOR DB AND TS IS 8)
 UPDATE QT DATABASE     = SYSTOOLS
 UPDATE QT TABLESPACE   = ARYTSQT
*
*-------------------------------------------------------------------
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*-------------------------------------------------------------------
*
*  The ROWDATA VSAM data set is dynamically created by the log
*  analysis services when creating SQL from the log.
*
*  The DSN PREFIX maximum length is 21 characters. The following
*  set of product controls are required and must be properly set
*  to ensure proper log data recoveries. The VOLSERS statement
*  value can be set to blanks, if required. A maximum of 3 volsers
*  can be specified.
*
*UPDATE LAS DSN PREFIX  = xxxxxxxxxxxxxxxxxxxxx
*UPDATE LAS VOLSERS     = TST02B,TST02B,TST02B
*UPDATE LAS VOLSERS     = aaaaa,bbbbbb,cccccc
 UPDATE LAS DSN PREFIX  = NALUR
 UPDATE LAS VOLSERS     = TST02B,TST026,TST021
 UPDATE LAS DATA AUNIT  = C
 UPDATE LAS DATA PQTY   = 00005
 UPDATE LAS DATA SQTY   = 00005
 UPDATE LAS INDEX AUNIT = C
 UPDATE LAS INDEX PQTY  = 00005
 UPDATE LAS INDEX SQTY  = 00005
*
*-------------------------------------------------------------------
* Sample statements to add/update character conversion information
*-------------------------------------------------------------------
*
*  Schema Level Repository Unicode data conversion information.
*  These values should not be changed. The IBM DB2 Recovery Expert
*  z/OS components require the following CCSID conversions to be
*  defined on the target systems.
*
 UPDATE CCS SLR TECHNQ  = ER            CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS SLR SBCS    = 00037         EBCDIC SBCS CCSID
 UPDATE CCS SLR DBCS    = 01200         UNICODE UT-8 DBCS CCSID
 UPDATE CCS SLR MIXED   = 01208         UNICODE UT-8 MIXED CCSID
*
*  Product output Unicode data conversion information.
*
 UPDATE CCS ARY TECHNQ  = ER            CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS ARY SBCS    = 00037         EBCDIC SBCS CCISD
```

```
 UPDATE CCS ARY DBCS    = 01200         UNICODE UT-8 DBCS CCSID
 UPDATE CCS ARY MIXED   = 01208         UNICODE UT-8 MIXED CCSID
*
*-------------------------------------------------------------------
* Sample statements to add/update default data set information
*-------------------------------------------------------------------
*
*  File tailoring work data set allocation.
*
 UPDATE FTW DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE FTW ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE FTW PQTY        = 00001         PRIMARY QTY
 UPDATE FTW SQTY        = 00001         SECONDARY QTY
*UPDATE FTW SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE FTW SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE FTW SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*
*  Image copy output data set allocation defaults.
*
 UPDATE ICF DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE ICF ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE ICF PQTY        = 00001         PRIMARY QTY
 UPDATE ICF SQTY        = 00001         SECONDARY QTY
*UPDATE ICF SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE ICF SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE ICF SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE ICF MULTIVOL    = xxx
*UPDATE ICF EXPIREDT    = xxxxxxx       EXPIRATION DATE
*UPDATE ICF RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE ICF FILENUM     = xxxx          LABEL FILE NUMBER
*
*  Recovery output data set allocation defaults.
*
 UPDATE RDA DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE RDA ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE RDA PQTY        = 00001         PRIMARY QTY
 UPDATE RDA SQTY        = 00001         SECONDARY QTY
*UPDATE RDA SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE RDA SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE RDA SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE RDA MULTIVOL    = xxx
*UPDATE RDA EXPIREDT    = xxxxxxx       EXPIRATION DATE
*UPDATE RDA RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE RDA FILENUM     = xxxx          LABEL FILE NUMBER
*
/*
//*
//PCFUPD2  EXEC PGM=ARY#UTIL,PARM='SETUP',REGION=4M
//STEPLIB  DD   DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//SYSUDUMP DD   SYSOUT=*
//DB2PARMS DD   DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//*
//* REPORTS
//*
//SYSOUT   DD   SYSOUT=*,RECFM=FBA,LRECL=133    SYSIN REPORT
```

```
//SYSPRINT DD   SYSOUT=*,RECFM=FBA,LRECL=133    PCF REPORT
//*
//* CONTROLS
//*
//SYSIN    DD   *
*
*--------------------------------------------------------------------
* Sample statements to add/update GDG model data set name.
*--------------------------------------------------------------------
*
* UPDATE GDG MODEL       = &gdg_model_dsname
*
*--------------------------------------------------------------------
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*--------------------------------------------------------------------
*
 SET DB2 SSID           = D8G2
 UPDATE DB2 ZPARMS      = DSNZPAG2
 UPDATE DB2 BOOTSTRAP1  = DB8GU.D8G2.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8GU.D8G2.BSDS02
 UPDATE DB2 LOADLIB1    = DB8G8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8G8.SDSNLOAD
*UPDATE DB2 LOADLIB3    =
*UPDATE DB2 LOADLIB4    =
*UPDATE DB2 LOADLIB5    =
*
*--------------------------------------------------------------------
* Sample statements to add/update ARY product plans
*--------------------------------------------------------------------
*
 SET DB2 SSID           = D8G2
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
*
*--------------------------------------------------------------------
* Sample statements to add/update product message library
*--------------------------------------------------------------------
*
 UPDATE ARY MSGLIBRARY  = ARY.V1R1M0.SARYMENU
*
*--------------------------------------------------------------------
* Sample statements to add/update log services options
*--------------------------------------------------------------------
*
 UPDATE ARY ARCHLOG1    = Y       USE ARCHIVE LOG 1
 UPDATE ARY ARCHLOG2    = N       USE ARCHIVE LOG 2
 UPDATE ARY ACTLOGPRI   = Y       ACTIVE LOG PRIORITY
```

```
*
*--------------------------------------------------------------------
* Sample statements to add/update data set prefix generation
*--------------------------------------------------------------------
*
*  The DSN PREFIX maximum length is 17 characters. If NULL
*  is specified then user id is used as data set prefix. Use &USERID
*  in the prefix to insert user id. Example: TEST.&USERID will
*  generate a data set prefix of 'TEST.MYID' where the user id is
*  'MYID'.
*
 UPDATE ARY DSN PREFIX  = RETOOLS
*
*--------------------------------------------------------------------
*
*--------------------------------------------------------------------
* Sample statements to add/update schema level repository data
* capture options.
*--------------------------------------------------------------------
*
 UPDATE SLR LOAD BINDS  = Y         N = package/plan versions are
*                                       not saved in schema level
*                                       repository.
 UPDATE SLR LOAD AUTHS  = Y         N = DB2 authorizations are
*                                       not saved in schema level
*                                       repository.
*
*--------------------------------------------------------------------
* Sample statements to add/update interproduct communication
* options.
*--------------------------------------------------------------------
*
 UPDATE IPC IPC_GROUPER = N         Y = Enable Grouper-related
*                                       table recovery.
*
*--------------------------------------------------------------------
* Sample statements to add/update table activity quiet time
* repository names.
*--------------------------------------------------------------------
*
*  EACH QT OWNER/NAME IS A 45 CHAR MAXIMUM LENGTH. THESE OBJECTS WILL
*  BE CREATED AUTOMATICALLY WHEN THE QUIET TIME REPORT/CAPTURE JCL
*  IS RUN IF THEY DO NOT ALREADY EXIST. DDL TO CREATE THESE OBJECTS
*  IS PROVIDED IN ARYDDL7 AND ARYDDL8 SAMPLE DDL MEMBERS.
*
*                         XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 UPDATE QT GRP TBOWNER  = SYSTOOLS
 UPDATE QT GRP TBNAME   = ARYQTG
 UPDATE QT GRP IXOWNER  = SYSTOOLS
 UPDATE QT GRP IXNAME   = ARYQTGX
 UPDATE QT ENTRY OWNER  = SYSTOOLS
 UPDATE QT ENTRY NAME   = ARYQT
*                         XXXXXXXX (MAX LENGTH FOR DB AND TS IS 8)
 UPDATE QT DATABASE     = SYSTOOLS
 UPDATE QT TABLESPACE   = ARYTSQT
```

```
*
*-------------------------------------------------------------------
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*-------------------------------------------------------------------
*
*  The ROWDATA VSAM data set is dynamically created by the log
*  analysis services when creating SQL from the log.
*
*  The DSN PREFIX maximum length is 21 characters. The following
*  set of product controls are required and must be properly set
*  to ensure proper log data recoveries. The VOLSERS statement
*  value can be set to blanks, if required. A maximum of 3 volsers
*  can be specified.
*
*UPDATE LAS DSN PREFIX  = xxxxxxxxxxxxxxxxxxxxx
*UPDATE LAS VOLSERS     = aaaaaa,bbbbbb,cccccc
*UPDATE LAS VOLSERS     = TST02B,TST02B,TST02B
 UPDATE LAS DSN PREFIX  = NALUR
 UPDATE LAS VOLSERS     = TST02B,TST026,TST021
 UPDATE LAS DATA AUNIT  = C
 UPDATE LAS DATA PQTY   = 00005
 UPDATE LAS DATA SQTY   = 00005
 UPDATE LAS INDEX AUNIT = C
 UPDATE LAS INDEX PQTY  = 00005
 UPDATE LAS INDEX SQTY  = 00005
*
*-------------------------------------------------------------------
* Sample statements to add/update character conversion information
*-------------------------------------------------------------------
*
*  Schema Level Repository Unicode data conversion information.
*  These values should not be changed. The IBM DB2 Recovery Expert
*  z/OS components require the following CCSID conversions to be
*  defined on the target systems.
*
 UPDATE CCS SLR TECHNQ  = ER           CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS SLR SBCS    = 00037        EBCDIC SBCS CCSID
 UPDATE CCS SLR DBCS    = 01200        UNICODE UT-8 DBCS CCSID
 UPDATE CCS SLR MIXED   = 01208        UNICODE UT-8 MIXED CCSID
*
*  Product output Unicode data conversion information.
*
 UPDATE CCS ARY TECHNQ  = ER           CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS ARY SBCS    = 00037        EBCDIC SBCS CCISD
 UPDATE CCS ARY DBCS    = 01200        UNICODE UT-8 DBCS CCSID
 UPDATE CCS ARY MIXED   = 01208        UNICODE UT-8 MIXED CCSID
*
*-------------------------------------------------------------------
* Sample statements to add/update default data set information
*-------------------------------------------------------------------
*
*  File tailoring work data set allocation.
*
 UPDATE FTW DEVICE      = SYSALLDA     DEVICE TYPE
```

```
 UPDATE FTW ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE FTW PQTY        = 00001         PRIMARY QTY
 UPDATE FTW SQTY        = 00001         SECONDARY QTY
*UPDATE FTW SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE FTW SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE FTW SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*
*  Image copy output data set allocation defaults.
*
 UPDATE ICF DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE ICF ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE ICF PQTY        = 00001         PRIMARY QTY
 UPDATE ICF SQTY        = 00001         SECONDARY QTY
*UPDATE ICF SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE ICF SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE ICF SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE ICF MULTIVOL    = xxx
*UPDATE ICF EXPIREDT    = xxxxxxx       EXPIRATION DATE
*UPDATE ICF RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE ICF FILENUM     = xxxx          LABEL FILE NUMBER
*
*  Recovery output data set allocation defaults.
*
 UPDATE RDA DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE RDA ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE RDA PQTY        = 00001         PRIMARY QTY
 UPDATE RDA SQTY        = 00001         SECONDARY QTY
*UPDATE RDA SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE RDA SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE RDA SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE RDA MULTIVOL    = xxx
*UPDATE RDA EXPIREDT    = xxxxxxx       EXPIRATION DATE
*UPDATE RDA RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE RDA FILENUM     = xxxx          LABEL FILE NUMBER
*
/*
//PCFUPD3  EXEC PGM=ARY#UTIL,PARM='SETUP',REGION=4M
//STEPLIB  DD   DISP=SHR,DSN=ARY.V1R1MO.SARYLOAD
//SYSUDUMP DD   SYSOUT=*
//DB2PARMS DD   DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//*
//* REPORTS
//*
//SYSOUT   DD   SYSOUT=*,RECFM=FBA,LRECL=133    SYSIN REPORT
//SYSPRINT DD   SYSOUT=*,RECFM=FBA,LRECL=133    PCF REPORT
//*
//* CONTROLS
//*
//SYSIN    DD   *
*
*------------------------------------------------------------------
* Sample statements to add/update GDG model data set name.
*------------------------------------------------------------------
*
* UPDATE GDG MODEL       = &gdg_model_dsname
```

```
*
*------------------------------------------------------------------
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*------------------------------------------------------------------
*
*
 SET DB2 SSID          = D8F1
 UPDATE DB2 ZPARMS     = DSNZPAF1
 UPDATE DB2 BOOTSTRAP1  = DB8FU.D8F1.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8FU.D8F1.BSDS02
 UPDATE DB2 LOADLIB1    = DB8F8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8F8.SDSNLOAD
*UPDATE DB2 LOADLIB3    =
*UPDATE DB2 LOADLIB4    =
*UPDATE DB2 LOADLIB5    =
*
*------------------------------------------------------------------
* Sample statements to add/update ARY product plans
*------------------------------------------------------------------
*
 SET DB2 SSID          = D8F1
 SET PRODUCT CFG       = NULL
 SET PRODUCT VER       = NULL
*
 UPDATE ARY PLAN1      = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2      = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3      = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4      = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5      = ARYPLAN5    LOG ANALYSIS SERVICES
*
*------------------------------------------------------------------
* Sample statements to add/update product message library
*------------------------------------------------------------------
*
 UPDATE ARY MSGLIBRARY  = ARY.V1R1M0.SARYMENU
*
*------------------------------------------------------------------
* Sample statements to add/update log services options
*------------------------------------------------------------------
*
 UPDATE ARY ARCHLOG1    = Y       USE ARCHIVE LOG 1
 UPDATE ARY ARCHLOG2    = N       USE ARCHIVE LOG 2
 UPDATE ARY ACTLOGPRI   = Y       ACTIVE LOG PRIORITY
*
*------------------------------------------------------------------
* Sample statements to add/update data set prefix generation
*------------------------------------------------------------------
*
*  The DSN PREFIX maximum length is 17 characters. If NULL
*  is specified then user id is used as data set prefix. Use &USERID
*  in the prefix to insert user id. Example: TEST.&USERID will
*  generate a data set prefix of 'TEST.MYID' where the user id is
*  'MYID'.
```

```
*
 UPDATE ARY DSN PREFIX  = RETOOLS
*
*-------------------------------------------------------------------
*
*-------------------------------------------------------------------
* Sample statements to add/update schema level repository data
* capture options.
*-------------------------------------------------------------------
*
 UPDATE SLR LOAD BINDS  = Y         N = package/plan versions are
*                                       not saved in schema level
*                                       repository.
 UPDATE SLR LOAD AUTHS  = Y         N = DB2 authorizations are
*                                       not saved in schema level
*                                       repository.
*
*-------------------------------------------------------------------
* Sample statements to add/update interproduct communication
* options.
*-------------------------------------------------------------------
*
 UPDATE IPC IPC_GROUPER = N         Y = Enable Grouper-related
*                                       table recovery.
*
*-------------------------------------------------------------------
* Sample statements to add/update table activity quiet time
* repository names.
*-------------------------------------------------------------------
*
*  EACH QT OWNER/NAME IS A 45 CHAR MAXIMUM LENGTH. THESE OBJECTS WILL
*  BE CREATED AUTOMATICALLY WHEN THE QUIET TIME REPORT/CAPTURE JCL
*  IS RUN IF THEY DO NOT ALREADY EXIST. DDL TO CREATE THESE OBJECTS
*  IS PROVIDED IN ARYDDL7 AND ARYDDL8 SAMPLE DDL MEMBERS.
*
*                         XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 UPDATE QT GRP TBOWNER  = SYSTOOLS
 UPDATE QT GRP TBNAME   = ARYQTG
 UPDATE QT GRP IXOWNER  = SYSTOOLS
 UPDATE QT GRP IXNAME   = ARYQTGX
 UPDATE QT ENTRY OWNER  = SYSTOOLS
 UPDATE QT ENTRY NAME   = ARYQT
*                         XXXXXXXX (MAX LENGTH FOR DB AND TS IS 8)
 UPDATE QT DATABASE     = SYSTOOLS
 UPDATE QT TABLESPACE   = ARYTSQT
*
*-------------------------------------------------------------------
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*-------------------------------------------------------------------
*
*  The ROWDATA VSAM data set is dynamically created by the log
*  analysis services when creating SQL from the log.
*
```

```
*  The DSN PREFIX maximum length is 21 characters. The following
*  set of product controls are required and must be properly set
*  to ensure proper log data recoveries. The VOLSERS statement
*  value can be set to blanks, if required. A maximum of 3 volsers
*  can be specified.
*
*UPDATE LAS DSN PREFIX  = xxxxxxxxxxxxxxxxxxxx
*UPDATE LAS VOLSERS     = aaaaaa,bbbbbb,cccccc
*UPDATE LAS VOLSERS     = TST02B,TST02B,TST02B
 UPDATE LAS DSN PREFIX  = NALUR
 UPDATE LAS VOLSERS     = TST02B,TST026,TST021
 UPDATE LAS DATA AUNIT  = C
 UPDATE LAS DATA PQTY   = 00005
 UPDATE LAS DATA SQTY   = 00005
 UPDATE LAS INDEX AUNIT = C
 UPDATE LAS INDEX PQTY  = 00005
 UPDATE LAS INDEX SQTY  = 00005
*
*-------------------------------------------------------------------
* Sample statements to add/update character conversion information
*-------------------------------------------------------------------
*
*  Schema Level Repository Unicode data conversion information.
*  These values should not be changed. The IBM DB2 Recovery Expert
*  z/OS components require the following CCSID conversions to be
*  defined on the target systems.
*
 UPDATE CCS SLR TECHNQ  = ER            CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS SLR SBCS    = 00037         EBCDIC SBCS CCSID
 UPDATE CCS SLR DBCS    = 01200         UNICODE UT-8 DBCS CCSID
 UPDATE CCS SLR MIXED   = 01208         UNICODE UT-8 MIXED CCSID
*
*  Product output Unicode data conversion information.
*
 UPDATE CCS ARY TECHNQ  = ER            CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS ARY SBCS    = 00037         EBCDIC SBCS CCISD
 UPDATE CCS ARY DBCS    = 01200         UNICODE UT-8 DBCS CCSID
 UPDATE CCS ARY MIXED   = 01208         UNICODE UT-8 MIXED CCSID
*
*-------------------------------------------------------------------
* Sample statements to add/update default data set information
*-------------------------------------------------------------------
*
*  File tailoring work data set allocation.
*
 UPDATE FTW DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE FTW ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE FTW PQTY        = 00001         PRIMARY QTY
 UPDATE FTW SQTY        = 00001         SECONDARY QTY
*UPDATE FTW SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE FTW SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE FTW SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*
*  Image copy output data set allocation defaults.
*
```

```
 UPDATE ICF DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE ICF ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE ICF PQTY        = 00001         PRIMARY QTY
 UPDATE ICF SQTY        = 00001         SECONDARY QTY
*UPDATE ICF SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE ICF SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE ICF SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE ICF MULTIVOL    = xxx
*UPDATE ICF EXPIREDT    = xxxxxxx       EXPIRATION DATE
*UPDATE ICF RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE ICF FILENUM     = xxxx          LABEL FILE NUMBER
*
*  Recovery output data set allocation defaults.
*
 UPDATE RDA DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE RDA ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE RDA PQTY        = 00001         PRIMARY QTY
 UPDATE RDA SQTY        = 00001         SECONDARY QTY
*UPDATE RDA SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE RDA SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE RDA SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE RDA MULTIVOL    = xxx
*UPDATE RDA EXPIREDT    = xxxxxxx       EXPIRATION DATE
*UPDATE RDA RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE RDA FILENUM     = xxxx          LABEL FILE NUMBER
*
/*
//*
//PCFUPD4  EXEC PGM=ARY#UTIL,PARM='SETUP',REGION=4M
//STEPLIB  DD   DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//SYSUDUMP DD   SYSOUT=*
//DB2PARMS DD   DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//*
//* REPORTS
//*
//SYSOUT   DD   SYSOUT=*,RECFM=FBA,LRECL=133    SYSIN REPORT
//SYSPRINT DD   SYSOUT=*,RECFM=FBA,LRECL=133    PCF REPORT
//*
//* CONTROLS
//*
//SYSIN    DD   *
*
*-------------------------------------------------------------------
* Sample statements to add/update GDG model data set name.
*-------------------------------------------------------------------
*
* UPDATE GDG MODEL       = &gdg_model_dsname
*
*-------------------------------------------------------------------
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*-------------------------------------------------------------------
*
```

```
 SET DB2 SSID           = D8F2
 UPDATE DB2 ZPARMS      = DSNZPAF2
 UPDATE DB2 BOOTSTRAP1  = DB8FU.D8F2.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8FU.D8F2.BSDS02
 UPDATE DB2 LOADLIB1    = DB8F8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8F8.SDSNLOAD
*UPDATE DB2 LOADLIB3    =
*UPDATE DB2 LOADLIB4    =
*UPDATE DB2 LOADLIB5    =
*
*-------------------------------------------------------------------
* Sample statements to add/update ARY product plans
*-------------------------------------------------------------------
*
 SET DB2 SSID           = D8F2
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
*
*-------------------------------------------------------------------
* Sample statements to add/update product message library
*-------------------------------------------------------------------
*
 UPDATE ARY MSGLIBRARY  = ARY.V1R1M0.SARYMENU
*
*-------------------------------------------------------------------
* Sample statements to add/update log services options
*-------------------------------------------------------------------
*
 UPDATE ARY ARCHLOG1    = Y       USE ARCHIVE LOG 1
 UPDATE ARY ARCHLOG2    = N       USE ARCHIVE LOG 2
 UPDATE ARY ACTLOGPRI   = Y       ACTIVE LOG PRIORITY
*
*-------------------------------------------------------------------
* Sample statements to add/update data set prefix generation
*-------------------------------------------------------------------
*
*  The DSN PREFIX maximum length is 17 characters. If NULL
*  is specified then user id is used as data set prefix. Use &USERID
*  in the prefix to insert user id. Example: TEST.&USERID will
*  generate a data set prefix of 'TEST.MYID' where the user id is
*  'MYID'.
*
 UPDATE ARY DSN PREFIX  = RETOOLS
*
*-------------------------------------------------------------------
*
*-------------------------------------------------------------------
* Sample statements to add/update schema level repository data
* capture options.
```

```
*------------------------------------------------------------------
*
 UPDATE SLR LOAD BINDS  = Y         N = package/plan versions are
*                                       not saved in schema level
*                                       repository.
 UPDATE SLR LOAD AUTHS  = Y         N = DB2 authorizations are
*                                       not saved in schema level
*                                       repository.
*
*------------------------------------------------------------------
* Sample statements to add/update interproduct communication
* options.
*------------------------------------------------------------------
*
 UPDATE IPC IPC_GROUPER = N         Y = Enable Grouper-related
*                                       table recovery.
*
*------------------------------------------------------------------
* Sample statements to add/update table activity quiet time
* repository names.
*------------------------------------------------------------------
*
*  EACH QT OWNER/NAME IS A 45 CHAR MAXIMUM LENGTH. THESE OBJECTS WILL
*  BE CREATED AUTOMATICALLY WHEN THE QUIET TIME REPORT/CAPTURE JCL
*  IS RUN IF THEY DO NOT ALREADY EXIST. DDL TO CREATE THESE OBJECTS
*  IS PROVIDED IN ARYDDL7 AND ARYDDL8 SAMPLE DDL MEMBERS.
*
*                         XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 UPDATE QT GRP TBOWNER  = SYSTOOLS
 UPDATE QT GRP TBNAME   = ARYQTG
 UPDATE QT GRP IXOWNER  = SYSTOOLS
 UPDATE QT GRP IXNAME   = ARYQTGX
 UPDATE QT ENTRY OWNER  = SYSTOOLS
 UPDATE QT ENTRY NAME   = ARYQT
*                         XXXXXXXX (MAX LENGTH FOR DB AND TS IS 8)
 UPDATE QT DATABASE     = SYSTOOLS
 UPDATE QT TABLESPACE   = ARYTSQT
*
*------------------------------------------------------------------
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*------------------------------------------------------------------
*
*  The ROWDATA VSAM data set is dynamically created by the log
*  analysis services when creating SQL from the log.
*
*  The DSN PREFIX maximum length is 21 characters. The following
*  set of product controls are required and must be properly set
*  to ensure proper log data recoveries. The VOLSERS statement
*  value can be set to blanks, if required. A maximum of 3 volsers
*  can be specified.
*
*UPDATE LAS DSN PREFIX  = xxxxxxxxxxxxxxxxxxxx
*UPDATE LAS VOLSERS     = aaaaaa,bbbbbb,cccccc
*UPDATE LAS VOLSERS     = TST02B,TST02B,TST02B
```

```
 UPDATE LAS DSN PREFIX  = NALUR
 UPDATE LAS VOLSERS     = TST02B,TST026,TST021
 UPDATE LAS DATA AUNIT  = C
 UPDATE LAS DATA PQTY   = 00005
 UPDATE LAS DATA SQTY   = 00005
 UPDATE LAS INDEX AUNIT = C
 UPDATE LAS INDEX PQTY  = 00005
 UPDATE LAS INDEX SQTY  = 00005
*
*-------------------------------------------------------------------
* Sample statements to add/update character conversion information
*-------------------------------------------------------------------
*
*  Schema Level Repository Unicode data conversion information.
*  These values should not be changed. The IBM DB2 Recovery Expert
*  z/OS components require the following CCSID conversions to be
*  defined on the target systems.
*
 UPDATE CCS SLR TECHNQ  = ER            CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS SLR SBCS    = 00037         EBCDIC SBCS CCSID
 UPDATE CCS SLR DBCS    = 01200         UNICODE UT-8 DBCS CCSID
 UPDATE CCS SLR MIXED   = 01208         UNICODE UT-8 MIXED CCSID
*
*  Product output Unicode data conversion information.
*
 UPDATE CCS ARY TECHNQ  = ER            CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS ARY SBCS    = 00037         EBCDIC SBCS CCISD
 UPDATE CCS ARY DBCS    = 01200         UNICODE UT-8 DBCS CCSID
 UPDATE CCS ARY MIXED   = 01208         UNICODE UT-8 MIXED CCSID
*
*-------------------------------------------------------------------
* Sample statements to add/update default data set information
*-------------------------------------------------------------------
*
*  File tailoring work data set allocation.
*
 UPDATE FTW DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE FTW ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE FTW PQTY        = 00001         PRIMARY QTY
 UPDATE FTW SQTY        = 00001         SECONDARY QTY
*UPDATE FTW SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE FTW SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE FTW SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*
*  Image copy output data set allocation defaults.
*
 UPDATE ICF DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE ICF ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE ICF PQTY        = 00001         PRIMARY QTY
 UPDATE ICF SQTY        = 00001         SECONDARY QTY
*UPDATE ICF SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE ICF SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE ICF SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE ICF MULTIVOL    = xxx
*UPDATE ICF EXPIREDT    = xxxxxxx       EXPIRATION DATE
```

```
*UPDATE ICF RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE ICF FILENUM     = xxxx          LABEL FILE NUMBER
*
*  Recovery output data set allocation defaults.
*
 UPDATE RDA DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE RDA ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE RDA PQTY        = 00001         PRIMARY QTY
 UPDATE RDA SQTY        = 00001         SECONDARY QTY
*UPDATE RDA SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE RDA SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE RDA SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE RDA MULTIVOL    = xxx
*UPDATE RDA EXPIREDT    = xxxxxxx       EXPIRATION DATE
*UPDATE RDA RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE RDA FILENUM     = xxxx          LABEL FILE NUMBER
*
/*
//
```

Example: E-5 Recovery Expert for z/OS Agent 53 product control file on LPAR SC59

```
//ARYSJA53 JOB (999,POK),'CONWAY',CLASS=A,MSGCLASS=H,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
//*******************************************************************
//*
//* 5697-H74
//* (c) Copyright Rocket Software, Inc. 2003 - 2006 All Rights
//* Reserved.
//*
//*******************************************************************
//* IBM DB2 Recovery Expert for z/OS 1.1.0
//*
//* THIS JCL WILL LOAD THE CONTROL FILE AS DEFINED IN SAMPLIB MEMBER
//* ARYSJ000.
//*
//* CAUTION: THIS IS NEITHER A JCL PROCEDURE NOR A COMPLETE JOB.
//*
//* Before running this job, you will have to make the following
//* modifications:
//*
//* 1. Change the job statement to meet your system requirements
//*
//* 2. Change 'ARY.V0110.CONTROL' below to the name of the VSAM
//*    control data set created in member ARYSJ000.
//*
//* 3. Change ARY.V1R1M0.SARYLOAD below to the name of the product
//*    LOADLIB used for IBM DB2 Recovery Expert Z/OS Tool.
//*
//* 4. Change ARY.V1R1M0.SARYMLIB below to the name of the product
//*    message library used for IBM DB2 Recovery Expert Z/OS Tool.
//*
//* 5. Look at the SYSIN DD * statements for the job. You must change
//*    all parameters prefixed with an '&'. At the bottom of this JCL
//*    is a sample SYSIN statement to help clarify the modifications
```

```
//*    needed. It shows the necessary formats for these parameters
//*    as well. Any number of subsystems may be initially defined.
//*    New information will be added and existing information will
//*    be updated.
//*
//*    -any optional statements can be commented out or deleted. To
//*     comment out, simply put a '*' in Column 1 of the control
//*     statement.
//*
//*    -the 'SET PRODUCT CFG' AND 'SET PRODUCT VER' control statements
//*     are used when defining multiple product configurations to
//*     support the execution of multiple versions of the product and
//*     multiple copies of the schema level repository on a DB2
//*     subsystem. Refer to the product user guide for information on
//*     how to use these control statements. If multiple product
//*     execution configurations are not required then remove or
//*     comment the 'SET PRODUCT' control statements or provide the
//*     value of NULL.
//*-------------------------------------------------------------------*
//*
//PCFUPDT  EXEC PGM=ARY#UTIL,PARM='SETUP',REGION=4M
//STEPLIB  DD   DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//SYSUDUMP DD   SYSOUT=*
//DB2PARMS DD   DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT53
//*
//* REPORTS
//*
//SYSOUT   DD   SYSOUT=*,RECFM=FBA,LRECL=133    SYSIN REPORT
//SYSPRINT DD   SYSOUT=*,RECFM=FBA,LRECL=133    PCF REPORT
//*
//* CONTROLS
//*
//SYSIN    DD   *
*
*-------------------------------------------------------------------
* Sample statements to add/update GDG model data set name.
*-------------------------------------------------------------------
*
* UPDATE GDG MODEL       = &gdg_model_dsname
*
*-------------------------------------------------------------------
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*-------------------------------------------------------------------
*
*
 SET DB2 SSID           = DB8A
 UPDATE DB2 ZPARMS      = DSNZPARM
 UPDATE DB2 BOOTSTRAP1  = DB8AU.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8AU.BSDS02
 UPDATE DB2 LOADLIB1    = DB8A8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8A8.SDSNLOAD
*UPDATE DB2 LOADLIB3    =
*UPDATE DB2 LOADLIB4    =
```

```
*UPDATE DB2 LOADLIB5    =
*
*
*--------------------------------------------------------------------
* Sample statements to add/update ARY product plans
*--------------------------------------------------------------------
*
 SET DB2 SSID           = DB8A
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
*
*--------------------------------------------------------------------
* Sample statements to add/update product message library
*--------------------------------------------------------------------
*
 UPDATE ARY MSGLIBRARY  = ARY.V1R1M0.SARYMENU
*
*--------------------------------------------------------------------
* Sample statements to add/update log services options
*--------------------------------------------------------------------
*
 UPDATE ARY ARCHLOG1    = Y       USE ARCHIVE LOG 1
 UPDATE ARY ARCHLOG2    = N       USE ARCHIVE LOG 2
 UPDATE ARY ACTLOGPRI   = Y       ACTIVE LOG PRIORITY
*
*--------------------------------------------------------------------
* Sample statements to add/update data set prefix generation
*--------------------------------------------------------------------
*
*  The DSN PREFIX maximum length is 17 characters. If NULL
*  is specified then user id is used as data set prefix. Use &USERID
*  in the prefix to insert user id. Example: TEST.&USERID will
*  generate a data set prefix of 'TEST.MYID' where the user id is
*  'MYID'.
*
*UPDATE ARY DSN PREFIX  = xxxxxxxx.xxxxxxxx
*
 UPDATE ARY DSN PREFIX  = NALUR
*
*--------------------------------------------------------------------
*--------------------------------------------------------------------
* Sample statements to add/update schema level repository data
* capture options.
*--------------------------------------------------------------------
*
 UPDATE SLR LOAD BINDS  = Y         N = package/plan versions are
*                                       not saved in schema level
*                                       repository.
```

```
 UPDATE SLR LOAD AUTHS  = Y         N = DB2 authorizations are
*                                       not saved in schema level
*                                       repository.
*
*-------------------------------------------------------------------
* Sample statements to add/update interproduct communication
* options.
*-------------------------------------------------------------------
*
 UPDATE IPC IPC_GROUPER = N         Y = Enable Grouper-related
*                                       table recovery.
*
*-------------------------------------------------------------------
* Sample statements to add/update table activity quiet time
* repository names.
*-------------------------------------------------------------------
*
*  EACH QT OWNER/NAME IS A 45 CHAR MAXIMUM LENGTH. THESE OBJECTS WILL
*  BE CREATED AUTOMATICALLY WHEN THE QUIET TIME REPORT/CAPTURE JCL
*  IS RUN IF THEY DO NOT ALREADY EXIST. DDL TO CREATE THESE OBJECTS
*  IS PROVIDED IN ARYDDL7 AND ARYDDL8 SAMPLE DDL MEMBERS.
*
*                         XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 UPDATE QT GRP TBOWNER  = SYSTOOLS
 UPDATE QT GRP TBNAME   = ARYQTG
 UPDATE QT GRP IXOWNER  = SYSTOOLS
 UPDATE QT GRP IXNAME   = ARYQTGX
 UPDATE QT ENTRY OWNER  = SYSTOOLS
 UPDATE QT ENTRY NAME   = ARYQT
*                         XXXXXXXX (MAX LENGTH FOR DB AND TS IS 8)
 UPDATE QT DATABASE     = SYSTOOLS
 UPDATE QT TABLESPACE   = ARYTSQT
*
*-------------------------------------------------------------------
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*-------------------------------------------------------------------
*
*  The ROWDATA VSAM data set is dynamically created by the log
*  analysis services when creating SQL from the log.
*
*  The DSN PREFIX maximum length is 21 characters. The following
*  set of product controls are required and must be properly set
*  to ensure proper log data recoveries. The VOLSERS statement
*  value can be set to blanks, if required. A maximum of 3 volsers
*  can be specified.
*
*UPDATE LAS DSN PREFIX  = xxxxxxxxxxxxxxxxxxxx
 UPDATE LAS DSN PREFIX  = NALUR
 UPDATE LAS VOLSERS     = OP1TSA,OP1TSB,OP1TSC
 UPDATE LAS DATA AUNIT  = C
 UPDATE LAS DATA PQTY   = 00005
 UPDATE LAS DATA SQTY   = 00005
 UPDATE LAS INDEX AUNIT = C
 UPDATE LAS INDEX PQTY  = 00005
```

```
 UPDATE LAS INDEX SQTY  = 00005
*
*-------------------------------------------------------------------
* Sample statements to add/update character conversion information
*-------------------------------------------------------------------
*
*  Schema Level Repository Unicode data conversion information.
*  These values should not be changed. The IBM DB2 Recovery Expert
*  z/OS components require the following CCSID conversions to be
*  defined on the target systems.
*
 UPDATE CCS SLR TECHNQ  = ER            CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS SLR SBCS    = 00037         EBCDIC SBCS CCSID
 UPDATE CCS SLR DBCS    = 01200         UNICODE UT-8 DBCS CCSID
 UPDATE CCS SLR MIXED   = 01208         UNICODE UT-8 MIXED CCSID
*
*  Product output Unicode data conversion information.
*
 UPDATE CCS ARY TECHNQ  = ER            CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS ARY SBCS    = 00037         EBCDIC SBCS CCISD
 UPDATE CCS ARY DBCS    = 01200         UNICODE UT-8 DBCS CCSID
 UPDATE CCS ARY MIXED   = 01208         UNICODE UT-8 MIXED CCSID
*
*-------------------------------------------------------------------
* Sample statements to add/update default data set information
*-------------------------------------------------------------------
*
*  File tailoring work data set allocation.
*
 UPDATE FTW DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE FTW ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE FTW PQTY        = 00001         PRIMARY QTY
 UPDATE FTW SQTY        = 00001         SECONDARY QTY
*UPDATE FTW SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE FTW SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE FTW SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*
*  Image copy output data set allocation defaults.
*
 UPDATE ICF DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE ICF ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE ICF PQTY        = 00001         PRIMARY QTY
 UPDATE ICF SQTY        = 00001         SECONDARY QTY
*UPDATE ICF SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE ICF SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE ICF SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE ICF MULTIVOL    = xxx
*UPDATE ICF EXPIREDT    = xxxxxxx       EXPIRATION DATE
*UPDATE ICF RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE ICF FILENUM     = xxxx          LABEL FILE NUMBER
*
*  Recovery output data set allocation defaults.
*
 UPDATE RDA DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE RDA ALCUNIT     = C             C=CYLS, T=TRACKS
```

```
 UPDATE RDA PQTY        = 00001         PRIMARY QTY
 UPDATE RDA SQTY        = 00001         SECONDARY QTY
*UPDATE RDA SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE RDA SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE RDA SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE RDA MULTIVOL    = xxx
*UPDATE RDA EXPIREDT    = xxxxxxx       EXPIRATION DATE
*UPDATE RDA RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE RDA FILENUM     = xxxx          LABEL FILE NUMBER
*
/*
```

Example: E-6 Recovery Expert for z/OS Agent 59 agent configuration file on all the LPARs

```
<!--*******************************************************************
5697-H74
(c) Copyright Rocket Software, Inc. 2003, 2006 All Rights Reserved.
*******************************************************************-->

<!--*******************************************************************
This is a sample configuration file for the IBM DB2 Recovery Expert
for z/OS 1.1.0 agent.

A copy of this file should be made, and customized as described below.
The job to run the agent must include a DD definition referring to the
customized file.  (See sample member ARYSJAGT.)

The default values for all parameters may be used, except for
<server-address> and <server-port>, which must be properly
configured to connect to the DB2 Recovery Expert server.

This file must have valid XML syntax.
*******************************************************************-->

<agent-config>

  <!--*****************************************************************
    server-address must specify the host name or IP address (in
    dotted-decimal notation, e.g., 1.2.3.4) of the DB2 Recovery
    Expert server to which the agent should connect.  If server-address
    is omitted, the agent will attempt to automatically discover the
    address and port number of a server.  (See below for additional
    server discovery parameters.)
  *****************************************************************-->

   <server-address>wtsc59.itso.ibm.com</server-address>


  <!--*****************************************************************
    server-port must specify the IP port number on the DB2 Recovery
    Expert server to which the agent should connect.

    Valid values are integers between 1 and 65535, inclusive.
  *****************************************************************-->
```

```
<server-port>9876</server-port>


<!--******************************************************************
  community-string is optional, and specifies the identification for
  which the agent listens when performing server discovery.  In order
  for the agent to discover and connect to a server, that server must
  be configured with the same community-string value.
*******************************************************************-->

<community-string>wtcs59</community-string>


<!--******************************************************************
  multicast-address is optional, and specifies the UDP multicast
  address on which the agent should listen for server announcements.
  In order for the agent to discover and connect to a server, that
  server must be configured with the same multicast-address value.

  Server discovery is performed only if the server-address parameter
  is omitted.

  Valid values are IP addresses in dotted-decimal notation, in the
  range from 224.0.1.0 to 238.255.255.255, inclusive.
*******************************************************************-->

<multicast-address>236.1.2.3</multicast-address>


<!--******************************************************************
  multicast-port is optional, and specifies the UDP multicast
  port on which the agent should listen for server announcements.
  In order for the agent to discover and connect to a server, that
  server must be configured with the same multicast-address value.

  Server discovery is performed only if the server-address parameter
  is omitted.

  Valid values are integers between 1 and 65535, inclusive.
*******************************************************************-->

<multicast-port>19875</multicast-port>


<!--******************************************************************
  control-file-dd specifies the DD name allocated to the product
  control file.  The name specified must be allocated in the JCL used
  to run the agent.  The default value is DB2PARMS.

  Valid values are strings conforming to DD name syntax rules.
*******************************************************************-->

<control-file-dd>DB2PARMS</control-file-dd>
```

```
<!--*****************************************************************
  log-level controls the amount of output log information that is
  generated by the agent.

  Valid values are:
  O - disable logging
  S - log severe error messages only
  E - log error and severe error messages
  W - log warning, error, and severe error messages
  I - log information, warning error, and severe error messages
        (recommended)
**************************************************************-->

<log-level>I</log-level>


<!--*****************************************************************
  server-connect-retry-max specifies the maximum number of
  attempts the agent should make to connect to the server, before
  exiting with an error.

  Typically, the server should be started and available before any
  agent is started, in which case the agent will immediately connect
  on the first attempt.  This parameter allows for the case when the
  server is not immediately available.  It also allows for the case
  where the server terminates for any reason.  In this case, the
  agent will cancel any pending requests, and attempt to reconnect
  to the server (presumably, after the server is restarted).

  Valid values are:
  0 - continue attempts to connect indefinitely
  integers greater than 0 - the maximum number of connection attempts
**************************************************************-->

<server-connect-retry-max>30</server-connect-retry-max>


<!--*****************************************************************
  server-connect-retry-delay specifies the number of seconds the
  agent should wait between attempts to connect to the server.

  The period of time (in seconds) that the agent will continue
  attempts to connect is:
      server-connect-retry-max * server-connect-retry-delay

  Valid values are integers greater than 0.
**************************************************************-->

<server-connect-retry-delay>10</server-connect-retry-delay>


<!--*****************************************************************
  request-thread-timeout specifies the number of seconds a
  thread/task created to do work for a specific user should remain
  idle before exiting.  Setting this value higher provides a better
```

```
  response to client requests, but consumes more resources (in the
  form of extra tasks that are not performing work).

  This value should be set high enough so that a task does not exit
  during a typical end-user client session, i.e., greater than the
  expected time between end-user actions in the client.

  Valid values are integers greater than 0.
*******************************************************************-->

<request-thread-timeout>300</request-thread-timeout>


<!--*******************************************************************
  uppercase-passwords specifies whether or not user IDs and
  passwords specified by end users should be folded to uppercase
  before using them to authenticate the user.

  Valid values are:
  true - uppercase user IDs and passwords
  false - use user IDs and passwords as entered by the user
*******************************************************************-->

<uppercase-passwords>true</uppercase-passwords>


<!--*******************************************************************
  job-poll-rate specifies the number of seconds the agent should wait
  before attempts to query the status of submitted jobs.  Lower values
  provide better response time to end users, but require more
  resources on the server.

  Valid values are integers greater than 0.
*******************************************************************-->

<job-poll-rate>5</job-poll-rate>


<!--*******************************************************************
  job-cancel-timeout specifies the number of seconds the agent should
  wait after cancelling a job for the job to end.  If the job does not
  end within this time period, the agent "abandons" the job, allowing
  it to continue running, and reporting an error to the server.

  Valid values are integers greater than 0.
*******************************************************************-->

<job-cancel-timeout>5</job-cancel-timeout>


<!--*******************************************************************
  check-ownership-external specifies whether or not an external
  security manager (such as RACF) is used to manage DB2 security.
  If set to false, the agent will evaluate authorizations recorded in
  the DB2 catalog in order to determine whether or not a given user
```

```
    has ownership rights on a specification.  If an external security
    manager is used, this process is not sufficient to determine
    ownership.  If set to true, the agent will additionally attempt to
    SET CURRENT SQLID to the specification owner to determine if a user
    has ownership rights.  This will work, but may result in spurious
    errors with SQL code -553, which may be recorded in DB2 audit
    traces.  The default is true.

    Valid values are true or false.
  *******************************************************************-->

  <check-ownership-external>true</check-ownership-external>


  <!--*******************************************************************
    The trace-* parameters cause additional tracing information to be
    logged, in order to diagnose errors that may occur during product
    execution.

    These parameters should not be enabled unless directed by product
    support, as there is a significant cost in performance when they
    are enabled.

    Valid values for each are true or false.
  *******************************************************************-->

  <trace-csi>false</trace-csi>
  <trace-db2-attachment>false</trace-db2-attachment>
  <trace-sql>false</trace-sql>
  <trace-ifi>false</trace-ifi>
  <trace-network>false</trace-network>
  <trace-xml>false</trace-xml>
  <trace-events>false</trace-events>
  <trace-config>false</trace-config>
</agent-config>
```

Example: E-7 Shared Recovery Expert for z/OS Agent 59 product control file on LPARs SC53 and SC67

```
//ARYSJ002 JOB (999,POK),'CONWAY',CLASS=A,MSGCLASS=H,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
/*JOBPARM L=999,SYSAFF=SC53
//*******************************************************************
//*
//* 5697-H74
//* (c) Copyright Rocket Software, Inc. 2003 - 2006 All Rights
//* Reserved.
//*
//*******************************************************************
//* IBM DB2 Recovery Expert for z/OS 1.1.0
//*
//* THIS JCL WILL LOAD THE CONTROL FILE AS DEFINED IN SAMPLIB MEMBER
//* ARYSJ000.
//*
//* CAUTION: THIS IS NEITHER A JCL PROCEDURE NOR A COMPLETE JOB.
//*
```

```
//* Before running this job, you will have to make the following
//* modifications:
//*
//* 1. Change the job statement to meet your system requirements
//*
//* 2. Change 'ARY.V0110.CONTROL' below to the name of the VSAM
//*    control data set created in member ARYSJ000.
//*
//* 3. Change ARY.V1R1M0.SARYLOAD below to the name of the product
//*    LOADLIB used for IBM DB2 Recovery Expert Z/OS Tool.
//*
//* 4. Change ARY.V1R1M0.SARYMLIB below to the name of the product
//*    message library used for IBM DB2 Recovery Expert Z/OS Tool.
//*
//* 5. Look at the SYSIN DD * statements for the job. You must change
//*    all parameters prefixed with an '&'. At the bottom of this JCL
//*    is a sample SYSIN statement to help clarify the modifications
//*    needed. It shows the necessary formats for these parameters
//*    as well. Any number of subsystems may be initially defined.
//*    New information will be added and existing information will
//*    be updated.
//*
//*    -any optional statements can be commented out or deleted. To
//*     comment out, simply put a '*' in Column 1 of the control
//*     statement.
//*
//*    -the 'SET PRODUCT CFG' AND 'SET PRODUCT VER' control statements
//*     are used when defining multiple product configurations to
//*     support the execution of multiple versions of the product and
//*     multiple copies of the schema level repository on a DB2
//*     subsystem. Refer to the product user guide for information on
//*     how to use these control statements. If multiple product
//*     execution configurations are not required then remove or
//*     comment the 'SET PRODUCT' control statements or provide the
//*     value of NULL.
//*-----------------------------------------------------------------*
//*
//*
//PCFUPD1  EXEC PGM=ARY#UTIL,PARM='SETUP',REGION=4M
//STEPLIB  DD   DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//SYSUDUMP DD   SYSOUT=*
//DB2PARMS DD   DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//*
//* REPORTS
//*
//SYSOUT   DD   SYSOUT=*,RECFM=FBA,LRECL=133    SYSIN REPORT
//SYSPRINT DD   SYSOUT=*,RECFM=FBA,LRECL=133    PCF REPORT
//*
//* CONTROLS
//*
//SYSIN    DD   *
*
*-----------------------------------------------------------------
* Sample statements to add/update GDG model data set name.
*-----------------------------------------------------------------
```

```
*
* UPDATE GDG MODEL       = &gdg_model_dsname
*
*-------------------------------------------------------------------
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*-------------------------------------------------------------------
*
 SET DB2 SSID           = D8G1
 UPDATE DB2 ZPARMS      = DSNZPAG1
 UPDATE DB2 BOOTSTRAP1  = DB8GU.D8G1.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8GU.D8G1.BSDS02
 UPDATE DB2 LOADLIB1    = DB8G8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8G8.SDSNLOAD
*UPDATE DB2 LOADLIB3    =
*UPDATE DB2 LOADLIB4    =
*UPDATE DB2 LOADLIB5    =
*
*-------------------------------------------------------------------
* Sample statements to add/update ARY product plans
*-------------------------------------------------------------------
*
 SET DB2 SSID           = D8G1
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
*
*
*-------------------------------------------------------------------
* Sample statements to add/update product message library
*-------------------------------------------------------------------
*
 UPDATE ARY MSGLIBRARY  = ARY.V1R1M0.SARYMENU
*
*-------------------------------------------------------------------
* Sample statements to add/update log services options
*-------------------------------------------------------------------
*
 UPDATE ARY ARCHLOG1    = Y       USE ARCHIVE LOG 1
 UPDATE ARY ARCHLOG2    = N       USE ARCHIVE LOG 2
 UPDATE ARY ACTLOGPRI   = Y       ACTIVE LOG PRIORITY
*
*-------------------------------------------------------------------
* Sample statements to add/update data set prefix generation
*-------------------------------------------------------------------
*
*  The DSN PREFIX maximum length is 17 characters. If NULL
*  is specified then user id is used as data set prefix. Use &USERID
*  in the prefix to insert user id. Example: TEST.&USERID will
```

```
*  generate a data set prefix of 'TEST.MYID' where the user id is
*  'MYID'.
*
 UPDATE ARY DSN PREFIX  = RETOOLS
*
*-------------------------------------------------------------------
*
*-------------------------------------------------------------------
* Sample statements to add/update schema level repository data
* capture options.
*-------------------------------------------------------------------
*
 UPDATE SLR LOAD BINDS  = Y         N = package/plan versions are
*                                       not saved in schema level
*                                       repository.
 UPDATE SLR LOAD AUTHS  = Y         N = DB2 authorizations are
*                                       not saved in schema level
*                                       repository.
*
*-------------------------------------------------------------------
* Sample statements to add/update interproduct communication
* options.
*-------------------------------------------------------------------
*
 UPDATE IPC IPC_GROUPER = N         Y = Enable Grouper-related
*                                       table recovery.
*
*-------------------------------------------------------------------
* Sample statements to add/update table activity quiet time
* repository names.
*-------------------------------------------------------------------
*
*  EACH QT OWNER/NAME IS A 45 CHAR MAXIMUM LENGTH. THESE OBJECTS WILL
*  BE CREATED AUTOMATICALLY WHEN THE QUIET TIME REPORT/CAPTURE JCL
*  IS RUN IF THEY DO NOT ALREADY EXIST. DDL TO CREATE THESE OBJECTS
*  IS PROVIDED IN ARYDDL7 AND ARYDDL8 SAMPLE DDL MEMBERS.
*
*                         XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 UPDATE QT GRP TBOWNER  = SYSTOOLS
 UPDATE QT GRP TBNAME   = ARYQTG
 UPDATE QT GRP IXOWNER  = SYSTOOLS
 UPDATE QT GRP IXNAME   = ARYQTGX
 UPDATE QT ENTRY OWNER  = SYSTOOLS
 UPDATE QT ENTRY NAME   = ARYQT
*                         XXXXXXXX (MAX LENGTH FOR DB AND TS IS 8)
 UPDATE QT DATABASE     = SYSTOOLS
 UPDATE QT TABLESPACE   = ARYTSQT
*
*-------------------------------------------------------------------
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*-------------------------------------------------------------------
*
*  The ROWDATA VSAM data set is dynamically created by the log
*  analysis services when creating SQL from the log.
```

```
*
*  The DSN PREFIX maximum length is 21 characters. The following
*  set of product controls are required and must be properly set
*  to ensure proper log data recoveries. The VOLSERS statement
*  value can be set to blanks, if required. A maximum of 3 volsers
*  can be specified.
*
*UPDATE LAS DSN PREFIX  = xxxxxxxxxxxxxxxxxxxx
*UPDATE LAS VOLSERS     = TST02B,TST02B,TST02B
*UPDATE LAS VOLSERS     = aaaaa,bbbbbb,cccccc
 UPDATE LAS DSN PREFIX  = NALUR
 UPDATE LAS VOLSERS     = TST02B,TST026,TST021
 UPDATE LAS DATA AUNIT  = C
 UPDATE LAS DATA PQTY   = 00005
 UPDATE LAS DATA SQTY   = 00005
 UPDATE LAS INDEX AUNIT = C
 UPDATE LAS INDEX PQTY  = 00005
 UPDATE LAS INDEX SQTY  = 00005
*
*-------------------------------------------------------------------
* Sample statements to add/update character conversion information
*-------------------------------------------------------------------
*
*  Schema Level Repository Unicode data conversion information.
*  These values should not be changed. The IBM DB2 Recovery Expert
*  z/OS components require the following CCSID conversions to be
*  defined on the target systems.
*
 UPDATE CCS SLR TECHNQ  = ER            CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS SLR SBCS    = 00037         EBCDIC SBCS CCSID
 UPDATE CCS SLR DBCS    = 01200         UNICODE UT-8 DBCS CCSID
 UPDATE CCS SLR MIXED   = 01208         UNICODE UT-8 MIXED CCSID
*
*  Product output Unicode data conversion information.
*
 UPDATE CCS ARY TECHNQ  = ER            CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS ARY SBCS    = 00037         EBCDIC SBCS CCISD
 UPDATE CCS ARY DBCS    = 01200         UNICODE UT-8 DBCS CCSID
 UPDATE CCS ARY MIXED   = 01208         UNICODE UT-8 MIXED CCSID
*
*-------------------------------------------------------------------
* Sample statements to add/update default data set information
*-------------------------------------------------------------------
*
*  File tailoring work data set allocation.
*
 UPDATE FTW DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE FTW ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE FTW PQTY        = 00001         PRIMARY QTY
 UPDATE FTW SQTY        = 00001         SECONDARY QTY
*UPDATE FTW SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE FTW SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE FTW SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*
*  Image copy output data set allocation defaults.
```

```
*
 UPDATE ICF DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE ICF ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE ICF PQTY        = 00001         PRIMARY QTY
 UPDATE ICF SQTY        = 00001         SECONDARY QTY
*UPDATE ICF SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE ICF SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE ICF SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE ICF MULTIVOL    = xxx
*UPDATE ICF EXPIREDT    = xxxxxxx       EXPIRATION DATE
*UPDATE ICF RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE ICF FILENUM     = xxxx          LABEL FILE NUMBER
*
*  Recovery output data set allocation defaults.
*
 UPDATE RDA DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE RDA ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE RDA PQTY        = 00001         PRIMARY QTY
 UPDATE RDA SQTY        = 00001         SECONDARY QTY
*UPDATE RDA SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE RDA SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE RDA SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE RDA MULTIVOL    = xxx
*UPDATE RDA EXPIREDT    = xxxxxxx       EXPIRATION DATE
*UPDATE RDA RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE RDA FILENUM     = xxxx          LABEL FILE NUMBER
*
/*
//*
//PCFUPD2  EXEC PGM=ARY#UTIL,PARM='SETUP',REGION=4M
//STEPLIB  DD   DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//SYSUDUMP DD   SYSOUT=*
//DB2PARMS DD   DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//*
//* REPORTS
//*
//SYSOUT   DD   SYSOUT=*,RECFM=FBA,LRECL=133    SYSIN REPORT
//SYSPRINT DD   SYSOUT=*,RECFM=FBA,LRECL=133    PCF REPORT
//*
//* CONTROLS
//*
//SYSIN    DD   *
*
*-------------------------------------------------------------------
* Sample statements to add/update GDG model data set name.
*-------------------------------------------------------------------
*
* UPDATE GDG MODEL       = &gdg_model_dsname
*
*-------------------------------------------------------------------
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*-------------------------------------------------------------------
*
```

```
 SET DB2 SSID           = D8G2
 UPDATE DB2 ZPARMS      = DSNZPAG2
 UPDATE DB2 BOOTSTRAP1  = DB8GU.D8G2.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8GU.D8G2.BSDS02
 UPDATE DB2 LOADLIB1    = DB8G8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8G8.SDSNLOAD
*UPDATE DB2 LOADLIB3    =
*UPDATE DB2 LOADLIB4    =
*UPDATE DB2 LOADLIB5    =
*
*-------------------------------------------------------------------
* Sample statements to add/update ARY product plans
*-------------------------------------------------------------------
*
 SET DB2 SSID           = D8G2
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
*
*-------------------------------------------------------------------
* Sample statements to add/update product message library
*-------------------------------------------------------------------
*
 UPDATE ARY MSGLIBRARY  = ARY.V1R1M0.SARYMENU
*
*-------------------------------------------------------------------
* Sample statements to add/update log services options
*-------------------------------------------------------------------
*
 UPDATE ARY ARCHLOG1    = Y        USE ARCHIVE LOG 1
 UPDATE ARY ARCHLOG2    = N        USE ARCHIVE LOG 2
 UPDATE ARY ACTLOGPRI   = Y        ACTIVE LOG PRIORITY
*
*-------------------------------------------------------------------
* Sample statements to add/update data set prefix generation
*-------------------------------------------------------------------
*
*  The DSN PREFIX maximum length is 17 characters. If NULL
*  is specified then user id is used as data set prefix. Use &USERID
*  in the prefix to insert user id. Example: TEST.&USERID will
*  generate a data set prefix of 'TEST.MYID' where the user id is
*  'MYID'.
*
 UPDATE ARY DSN PREFIX  = RETOOLS
*
*-------------------------------------------------------------------
*
*-------------------------------------------------------------------
* Sample statements to add/update schema level repository data
* capture options.
```

```
*-------------------------------------------------------------------
*
 UPDATE SLR LOAD BINDS  = Y         N = package/plan versions are
*                                       not saved in schema level
*                                       repository.
 UPDATE SLR LOAD AUTHS  = Y         N = DB2 authorizations are
*                                       not saved in schema level
*                                       repository.
*
*-------------------------------------------------------------------
* Sample statements to add/update interproduct communication
* options.
*-------------------------------------------------------------------
*
 UPDATE IPC IPC_GROUPER = N         Y = Enable Grouper-related
*                                       table recovery.
*
*-------------------------------------------------------------------
* Sample statements to add/update table activity quiet time
* repository names.
*-------------------------------------------------------------------
*
*  EACH QT OWNER/NAME IS A 45 CHAR MAXIMUM LENGTH. THESE OBJECTS WILL
*  BE CREATED AUTOMATICALLY WHEN THE QUIET TIME REPORT/CAPTURE JCL
*  IS RUN IF THEY DO NOT ALREADY EXIST. DDL TO CREATE THESE OBJECTS
*  IS PROVIDED IN ARYDDL7 AND ARYDDL8 SAMPLE DDL MEMBERS.
*
*                         XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 UPDATE QT GRP TBOWNER  = SYSTOOLS
 UPDATE QT GRP TBNAME   = ARYQTG
 UPDATE QT GRP IXOWNER  = SYSTOOLS
 UPDATE QT GRP IXNAME   = ARYQTGX
 UPDATE QT ENTRY OWNER  = SYSTOOLS
 UPDATE QT ENTRY NAME   = ARYQT
*                         XXXXXXXX (MAX LENGTH FOR DB AND TS IS 8)
 UPDATE QT DATABASE     = SYSTOOLS
 UPDATE QT TABLESPACE   = ARYTSQT
*
*-------------------------------------------------------------------
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*-------------------------------------------------------------------
*
*  The ROWDATA VSAM data set is dynamically created by the log
*  analysis services when creating SQL from the log.
*
*  The DSN PREFIX maximum length is 21 characters. The following
*  set of product controls are required and must be properly set
*  to ensure proper log data recoveries. The VOLSERS statement
*  value can be set to blanks, if required. A maximum of 3 volsers
*  can be specified.
*
*UPDATE LAS DSN PREFIX  = xxxxxxxxxxxxxxxxxxxxx
*UPDATE LAS VOLSERS     = aaaaaa,bbbbbb,cccccc
*UPDATE LAS VOLSERS     = TSTO2B,TSTO2B,TSTO2B
```

```
 UPDATE LAS DSN PREFIX  = NALUR
 UPDATE LAS VOLSERS     = TSTO2B,TSTO26,TSTO21
 UPDATE LAS DATA AUNIT  = C
 UPDATE LAS DATA PQTY   = 00005
 UPDATE LAS DATA SQTY   = 00005
 UPDATE LAS INDEX AUNIT = C
 UPDATE LAS INDEX PQTY  = 00005
 UPDATE LAS INDEX SQTY  = 00005
*
*------------------------------------------------------------------
* Sample statements to add/update character conversion information
*------------------------------------------------------------------
*
*  Schema Level Repository Unicode data conversion information.
*  These values should not be changed. The IBM DB2 Recovery Expert
*  z/OS components require the following CCSID conversions to be
*  defined on the target systems.
*
 UPDATE CCS SLR TECHNQ  = ER            CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS SLR SBCS    = 00037         EBCDIC SBCS CCSID
 UPDATE CCS SLR DBCS    = 01200         UNICODE UT-8 DBCS CCSID
 UPDATE CCS SLR MIXED   = 01208         UNICODE UT-8 MIXED CCSID
*
*  Product output Unicode data conversion information.
*
 UPDATE CCS ARY TECHNQ  = ER            CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS ARY SBCS    = 00037         EBCDIC SBCS CCISD
 UPDATE CCS ARY DBCS    = 01200         UNICODE UT-8 DBCS CCSID
 UPDATE CCS ARY MIXED   = 01208         UNICODE UT-8 MIXED CCSID
*
*------------------------------------------------------------------
* Sample statements to add/update default data set information
*------------------------------------------------------------------
*
*  File tailoring work data set allocation.
*
 UPDATE FTW DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE FTW ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE FTW PQTY        = 00001         PRIMARY QTY
 UPDATE FTW SQTY        = 00001         SECONDARY QTY
*UPDATE FTW SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE FTW SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE FTW SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*
*  Image copy output data set allocation defaults.
*
 UPDATE ICF DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE ICF ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE ICF PQTY        = 00001         PRIMARY QTY
 UPDATE ICF SQTY        = 00001         SECONDARY QTY
*UPDATE ICF SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE ICF SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE ICF SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE ICF MULTIVOL    = xxx
*UPDATE ICF EXPIREDT    = xxxxxxx       EXPIRATION DATE
```

```
*UPDATE ICF RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE ICF FILENUM     = xxxx          LABEL FILE NUMBER
*
*  Recovery output data set allocation defaults.
*
 UPDATE RDA DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE RDA ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE RDA PQTY        = 00001         PRIMARY QTY
 UPDATE RDA SQTY        = 00001         SECONDARY QTY
*UPDATE RDA SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE RDA SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE RDA SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE RDA MULTIVOL    = xxx
*UPDATE RDA EXPIREDT    = xxxxxxx       EXPIRATION DATE
*UPDATE RDA RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE RDA FILENUM     = xxxx          LABEL FILE NUMBER
*
/*
//PCFUPD3  EXEC PGM=ARY#UTIL,PARM='SETUP',REGION=4M
//STEPLIB  DD   DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//SYSUDUMP DD   SYSOUT=*
//DB2PARMS DD   DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//*
//* REPORTS
//*
//SYSOUT   DD   SYSOUT=*,RECFM=FBA,LRECL=133    SYSIN REPORT
//SYSPRINT DD   SYSOUT=*,RECFM=FBA,LRECL=133    PCF REPORT
//*
//* CONTROLS
//*
//SYSIN    DD   *
*
*------------------------------------------------------------------
* Sample statements to add/update GDG model data set name.
*------------------------------------------------------------------
*
* UPDATE GDG MODEL       = &gdg_model_dsname
*
*------------------------------------------------------------------
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*------------------------------------------------------------------
*
*
 SET DB2 SSID           = D8F1
 UPDATE DB2 ZPARMS      = DSNZPAF1
 UPDATE DB2 BOOTSTRAP1  = DB8FU.D8F1.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8FU.D8F1.BSDS02
 UPDATE DB2 LOADLIB1    = DB8F8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8F8.SDSNLOAD
*UPDATE DB2 LOADLIB3    =
*UPDATE DB2 LOADLIB4    =
*UPDATE DB2 LOADLIB5    =
*
```

```
*-----------------------------------------------------------------
* Sample statements to add/update ARY product plans
*-----------------------------------------------------------------
*
 SET DB2 SSID           = D8F1
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
*
*-----------------------------------------------------------------
* Sample statements to add/update product message library
*-----------------------------------------------------------------
*
 UPDATE ARY MSGLIBRARY  = ARY.V1R1M0.SARYMENU
*
*-----------------------------------------------------------------
* Sample statements to add/update log services options
*-----------------------------------------------------------------
*
 UPDATE ARY ARCHLOG1    = Y       USE ARCHIVE LOG 1
 UPDATE ARY ARCHLOG2    = N       USE ARCHIVE LOG 2
 UPDATE ARY ACTLOGPRI   = Y       ACTIVE LOG PRIORITY
*
*-----------------------------------------------------------------
* Sample statements to add/update data set prefix generation
*-----------------------------------------------------------------
*
*  The DSN PREFIX maximum length is 17 characters. If NULL
*  is specified then user id is used as data set prefix. Use &USERID
*  in the prefix to insert user id. Example: TEST.&USERID will
*  generate a data set prefix of 'TEST.MYID' where the user id is
*  'MYID'.
*
 UPDATE ARY DSN PREFIX  = RETOOLS
*
*-----------------------------------------------------------------
*
*-----------------------------------------------------------------
* Sample statements to add/update schema level repository data
* capture options.
*-----------------------------------------------------------------
*
 UPDATE SLR LOAD BINDS  = Y         N = package/plan versions are
*                                       not saved in schema level
*                                       repository.
 UPDATE SLR LOAD AUTHS  = Y         N = DB2 authorizations are
*                                       not saved in schema level
*                                       repository.
*
*-----------------------------------------------------------------
```

```
* Sample statements to add/update interproduct communication
* options.
*-------------------------------------------------------------------
*
 UPDATE IPC IPC_GROUPER = N         Y = Enable Grouper-related
*                                       table recovery.
*
*-------------------------------------------------------------------
* Sample statements to add/update table activity quiet time
* repository names.
*-------------------------------------------------------------------
*
*  EACH QT OWNER/NAME IS A 45 CHAR MAXIMUM LENGTH. THESE OBJECTS WILL
*  BE CREATED AUTOMATICALLY WHEN THE QUIET TIME REPORT/CAPTURE JCL
*  IS RUN IF THEY DO NOT ALREADY EXIST. DDL TO CREATE THESE OBJECTS
*  IS PROVIDED IN ARYDDL7 AND ARYDDL8 SAMPLE DDL MEMBERS.
*
*                         XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 UPDATE QT GRP TBOWNER  = SYSTOOLS
 UPDATE QT GRP TBNAME   = ARYQTG
 UPDATE QT GRP IXOWNER  = SYSTOOLS
 UPDATE QT GRP IXNAME   = ARYQTGX
 UPDATE QT ENTRY OWNER  = SYSTOOLS
 UPDATE QT ENTRY NAME   = ARYQT
*                         XXXXXXXX (MAX LENGTH FOR DB AND TS IS 8)
 UPDATE QT DATABASE     = SYSTOOLS
 UPDATE QT TABLESPACE   = ARYTSQT
*
*-------------------------------------------------------------------
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*-------------------------------------------------------------------
*
*  The ROWDATA VSAM data set is dynamically created by the log
*  analysis services when creating SQL from the log.
*
*  The DSN PREFIX maximum length is 21 characters. The following
*  set of product controls are required and must be properly set
*  to ensure proper log data recoveries. The VOLSERS statement
*  value can be set to blanks, if required. A maximum of 3 volsers
*  can be specified.
*
*UPDATE LAS DSN PREFIX  = xxxxxxxxxxxxxxxxxxxxx
*UPDATE LAS VOLSERS     = aaaaaa,bbbbbb,cccccc
*UPDATE LAS VOLSERS     = TST02B,TST02B,TST02B
 UPDATE LAS DSN PREFIX  = NALUR
 UPDATE LAS VOLSERS     = TST02B,TST026,TST021
 UPDATE LAS DATA AUNIT  = C
 UPDATE LAS DATA PQTY   = 00005
 UPDATE LAS DATA SQTY   = 00005
 UPDATE LAS INDEX AUNIT = C
 UPDATE LAS INDEX PQTY  = 00005
 UPDATE LAS INDEX SQTY  = 00005
*
*-------------------------------------------------------------------
```

```
* Sample statements to add/update character conversion information
*-------------------------------------------------------------------
*
*  Schema Level Repository Unicode data conversion information.
*  These values should not be changed. The IBM DB2 Recovery Expert
*  z/OS components require the following CCSID conversions to be
*  defined on the target systems.
*
 UPDATE CCS SLR TECHNQ  = ER            CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS SLR SBCS    = 00037         EBCDIC SBCS CCSID
 UPDATE CCS SLR DBCS    = 01200         UNICODE UT-8 DBCS CCSID
 UPDATE CCS SLR MIXED   = 01208         UNICODE UT-8 MIXED CCSID
*
*  Product output Unicode data conversion information.
*
 UPDATE CCS ARY TECHNQ  = ER            CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS ARY SBCS    = 00037         EBCDIC SBCS CCISD
 UPDATE CCS ARY DBCS    = 01200         UNICODE UT-8 DBCS CCSID
 UPDATE CCS ARY MIXED   = 01208         UNICODE UT-8 MIXED CCSID
*
*-------------------------------------------------------------------
* Sample statements to add/update default data set information
*-------------------------------------------------------------------
*
*  File tailoring work data set allocation.
*
 UPDATE FTW DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE FTW ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE FTW PQTY        = 00001         PRIMARY QTY
 UPDATE FTW SQTY        = 00001         SECONDARY QTY
*UPDATE FTW SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE FTW SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE FTW SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*
*  Image copy output data set allocation defaults.
*
 UPDATE ICF DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE ICF ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE ICF PQTY        = 00001         PRIMARY QTY
 UPDATE ICF SQTY        = 00001         SECONDARY QTY
*UPDATE ICF SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE ICF SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE ICF SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE ICF MULTIVOL    = xxx
*UPDATE ICF EXPIREDT    = xxxxxxx       EXPIRATION DATE
*UPDATE ICF RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE ICF FILENUM     = xxxx          LABEL FILE NUMBER
*
*  Recovery output data set allocation defaults.
*
 UPDATE RDA DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE RDA ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE RDA PQTY        = 00001         PRIMARY QTY
 UPDATE RDA SQTY        = 00001         SECONDARY QTY
*UPDATE RDA SMSDC       = xxxxxxxx      SMS DATA CLASS
```

```
*UPDATE RDA SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE RDA SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE RDA MULTIVOL    = xxx
*UPDATE RDA EXPIREDT    = xxxxxxx       EXPIRATION DATE
*UPDATE RDA RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE RDA FILENUM     = xxxx          LABEL FILE NUMBER
*
/*
//*
//PCFUPD4  EXEC PGM=ARY#UTIL,PARM='SETUP',REGION=4M
//STEPLIB  DD   DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//SYSUDUMP DD   SYSOUT=*
//DB2PARMS DD   DISP=SHR,DSN=ARY.DB2PARMS.CONTROL
//*
//* REPORTS
//*
//SYSOUT   DD   SYSOUT=*,RECFM=FBA,LRECL=133    SYSIN REPORT
//SYSPRINT DD   SYSOUT=*,RECFM=FBA,LRECL=133    PCF REPORT
//*
//* CONTROLS
//*
//SYSIN    DD   *
*
*-------------------------------------------------------------------
* Sample statements to add/update GDG model data set name.
*-------------------------------------------------------------------
*
* UPDATE GDG MODEL       = &gdg_model_dsname
*
*-------------------------------------------------------------------
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*-------------------------------------------------------------------
*
 SET DB2 SSID           = D8F2
 UPDATE DB2 ZPARMS      = DSNZPAF2
 UPDATE DB2 BOOTSTRAP1  = DB8FU.D8F2.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8FU.D8F2.BSDS02
 UPDATE DB2 LOADLIB1    = DB8F8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8F8.SDSNLOAD
*UPDATE DB2 LOADLIB3    =
*UPDATE DB2 LOADLIB4    =
*UPDATE DB2 LOADLIB5    =
*
*-------------------------------------------------------------------
* Sample statements to add/update ARY product plans
*-------------------------------------------------------------------
*
 SET DB2 SSID           = D8F2
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
```

```
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
*
*--------------------------------------------------------------------
* Sample statements to add/update product message library
*--------------------------------------------------------------------
*
 UPDATE ARY MSGLIBRARY  = ARY.V1R1M0.SARYMENU
*
*--------------------------------------------------------------------
* Sample statements to add/update log services options
*--------------------------------------------------------------------
*
 UPDATE ARY ARCHLOG1    = Y       USE ARCHIVE LOG 1
 UPDATE ARY ARCHLOG2    = N       USE ARCHIVE LOG 2
 UPDATE ARY ACTLOGPRI   = Y       ACTIVE LOG PRIORITY
*
*--------------------------------------------------------------------
* Sample statements to add/update data set prefix generation
*--------------------------------------------------------------------
*
*  The DSN PREFIX maximum length is 17 characters. If NULL
*  is specified then user id is used as data set prefix. Use &USERID
*  in the prefix to insert user id. Example: TEST.&USERID will
*  generate a data set prefix of 'TEST.MYID' where the user id is
*  'MYID'.
*
 UPDATE ARY DSN PREFIX  = RETOOLS
*
*--------------------------------------------------------------------
*
*--------------------------------------------------------------------
* Sample statements to add/update schema level repository data
* capture options.
*--------------------------------------------------------------------
*
 UPDATE SLR LOAD BINDS  = Y         N = package/plan versions are
*                                       not saved in schema level
*                                       repository.
 UPDATE SLR LOAD AUTHS  = Y         N = DB2 authorizations are
*                                       not saved in schema level
*                                       repository.
*
*--------------------------------------------------------------------
* Sample statements to add/update interproduct communication
* options.
*--------------------------------------------------------------------
*
 UPDATE IPC IPC_GROUPER = N         Y = Enable Grouper-related
*                                       table recovery.
*
*--------------------------------------------------------------------
* Sample statements to add/update table activity quiet time
* repository names.
```

```
*-------------------------------------------------------------------
*
*  EACH QT OWNER/NAME IS A 45 CHAR MAXIMUM LENGTH. THESE OBJECTS WILL
*  BE CREATED AUTOMATICALLY WHEN THE QUIET TIME REPORT/CAPTURE JCL
*  IS RUN IF THEY DO NOT ALREADY EXIST. DDL TO CREATE THESE OBJECTS
*  IS PROVIDED IN ARYDDL7 AND ARYDDL8 SAMPLE DDL MEMBERS.
*
*                       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 UPDATE QT GRP TBOWNER  = SYSTOOLS
 UPDATE QT GRP TBNAME   = ARYQTG
 UPDATE QT GRP IXOWNER  = SYSTOOLS
 UPDATE QT GRP IXNAME   = ARYQTGX
 UPDATE QT ENTRY OWNER  = SYSTOOLS
 UPDATE QT ENTRY NAME   = ARYQT
*                       XXXXXXXX (MAX LENGTH FOR DB AND TS IS 8)
 UPDATE QT DATABASE     = SYSTOOLS
 UPDATE QT TABLESPACE   = ARYTSQT
*
*-------------------------------------------------------------------
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*-------------------------------------------------------------------
*
*  The ROWDATA VSAM data set is dynamically created by the log
*  analysis services when creating SQL from the log.
*
*  The DSN PREFIX maximum length is 21 characters. The following
*  set of product controls are required and must be properly set
*  to ensure proper log data recoveries. The VOLSERS statement
*  value can be set to blanks, if required. A maximum of 3 volsers
*  can be specified.
*
*UPDATE LAS DSN PREFIX  = xxxxxxxxxxxxxxxxxxxxx
*UPDATE LAS VOLSERS     = aaaaaa,bbbbbb,cccccc
*UPDATE LAS VOLSERS     = TST02B,TST02B,TST02B
 UPDATE LAS DSN PREFIX  = NALUR
 UPDATE LAS VOLSERS     = TST02B,TST026,TST021
 UPDATE LAS DATA AUNIT  = C
 UPDATE LAS DATA PQTY   = 00005
 UPDATE LAS DATA SQTY   = 00005
 UPDATE LAS INDEX AUNIT = C
 UPDATE LAS INDEX PQTY  = 00005
 UPDATE LAS INDEX SQTY  = 00005
*
*-------------------------------------------------------------------
* Sample statements to add/update character conversion information
*-------------------------------------------------------------------
*
*  Schema Level Repository Unicode data conversion information.
*  These values should not be changed. The IBM DB2 Recovery Expert
*  z/OS components require the following CCSID conversions to be
*  defined on the target systems.
*
 UPDATE CCS SLR TECHNQ  = ER            CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS SLR SBCS    = 00037         EBCDIC SBCS CCSID
```

```
 UPDATE CCS SLR DBCS    = 01200         UNICODE UT-8 DBCS CCSID
 UPDATE CCS SLR MIXED   = 01208         UNICODE UT-8 MIXED CCSID
*
*  Product output Unicode data conversion information.
*
 UPDATE CCS ARY TECHNQ  = ER            CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS ARY SBCS    = 00037         EBCDIC SBCS CCISD
 UPDATE CCS ARY DBCS    = 01200         UNICODE UT-8 DBCS CCSID
 UPDATE CCS ARY MIXED   = 01208         UNICODE UT-8 MIXED CCSID
*
*-------------------------------------------------------------------
* Sample statements to add/update default data set information
*-------------------------------------------------------------------
*
*  File tailoring work data set allocation.
*
 UPDATE FTW DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE FTW ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE FTW PQTY        = 00001         PRIMARY QTY
 UPDATE FTW SQTY        = 00001         SECONDARY QTY
*UPDATE FTW SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE FTW SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE FTW SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*
*  Image copy output data set allocation defaults.
*
 UPDATE ICF DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE ICF ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE ICF PQTY        = 00001         PRIMARY QTY
 UPDATE ICF SQTY        = 00001         SECONDARY QTY
*UPDATE ICF SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE ICF SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE ICF SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE ICF MULTIVOL    = xxx
*UPDATE ICF EXPIREDT    = xxxxxxx       EXPIRATION DATE
*UPDATE ICF RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE ICF FILENUM     = xxxx          LABEL FILE NUMBER
*
*  Recovery output data set allocation defaults.
*
 UPDATE RDA DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE RDA ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE RDA PQTY        = 00001         PRIMARY QTY
 UPDATE RDA SQTY        = 00001         SECONDARY QTY
*UPDATE RDA SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE RDA SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE RDA SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE RDA MULTIVOL    = xxx
*UPDATE RDA EXPIREDT    = xxxxxxx       EXPIRATION DATE
*UPDATE RDA RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE RDA FILENUM     = xxxx          LABEL FILE NUMBER
*
/*
//
```

Example: E-8 Recovery Expert for z/OS Agent 59 product control file on LPAR SC59

```
//ARYSJA59 JOB (999,POK),'CONWAY',CLASS=A,MSGCLASS=H,
// NOTIFY=&SYSUID,TIME=1440,REGION=0M
//*********************************************************************
//*
//* 5697-H74
//* (c) Copyright Rocket Software, Inc. 2003 - 2006 All Rights
//* Reserved.
//*
//*********************************************************************
//* IBM DB2 Recovery Expert for z/OS 1.1.0
//*
//* THIS JCL WILL LOAD THE CONTROL FILE AS DEFINED IN SAMPLIB MEMBER
//* ARYSJ000.
//*
//* CAUTION: THIS IS NEITHER A JCL PROCEDURE NOR A COMPLETE JOB.
//*
//* Before running this job, you will have to make the following
//* modifications:
//*
//* 1. Change the job statement to meet your system requirements
//*
//* 2. Change 'ARY.V0110.CONTROL' below to the name of the VSAM
//*    control data set created in member ARYSJ000.
//*
//* 3. Change ARY.V1R1M0.SARYLOAD below to the name of the product
//*    LOADLIB used for IBM DB2 Recovery Expert Z/OS Tool.
//*
//* 4. Change ARY.V1R1M0.SARYMLIB below to the name of the product
//*    message library used for IBM DB2 Recovery Expert Z/OS Tool.
//*
//* 5. Look at the SYSIN DD * statements for the job. You must change
//*    all parameters prefixed with an '&'. At the bottom of this JCL
//*    is a sample SYSIN statement to help clarify the modifications
//*    needed. It shows the necessary formats for these parameters
//*    as well. Any number of subsystems may be initially defined.
//*    New information will be added and existing information will
//*    be updated.
//*
//*    -any optional statements can be commented out or deleted. To
//*     comment out, simply put a '*' in Column 1 of the control
//*     statement.
//*
//*    -the 'SET PRODUCT CFG' AND 'SET PRODUCT VER' control statements
//*     are used when defining multiple product configurations to
//*     support the execution of multiple versions of the product and
//*     multiple copies of the schema level repository on a DB2
//*     subsystem. Refer to the product user guide for information on
//*     how to use these control statements. If multiple product
//*     execution configurations are not required then remove or
//*     comment the 'SET PRODUCT' control statements or provide the
//*     value of NULL.
//*-------------------------------------------------------------------*
//*
```

```
//PCFUPDT  EXEC PGM=ARY#UTIL,PARM='SETUP',REGION=4M
//STEPLIB  DD   DISP=SHR,DSN=ARY.V1R1M0.SARYLOAD
//SYSUDUMP DD   SYSOUT=*
//DB2PARMS DD   DISP=SHR,DSN=ARY.DB2PARMS.CONTROL.AGT59
//*
//* REPORTS
//*
//SYSOUT   DD   SYSOUT=*,RECFM=FBA,LRECL=133    SYSIN REPORT
//SYSPRINT DD   SYSOUT=*,RECFM=FBA,LRECL=133    PCF REPORT
//*
//* CONTROLS
//*
//SYSIN    DD   *
*
*-------------------------------------------------------------------
* Sample statements to add/update GDG model data set name.
*-------------------------------------------------------------------
*
* UPDATE GDG MODEL       = &gdg_model_dsname
*
*-------------------------------------------------------------------
* Sample statements to add/update DB2 subsystem information.
* Multiple sets of following DB2 information control statements
* can be created and run in a single setup run.
*-------------------------------------------------------------------
*
*
 SET DB2 SSID           = DB8W
 UPDATE DB2 ZPARMS      = DSNZPARM
 UPDATE DB2 BOOTSTRAP1  = DB8WU.DB2.BSDS01
 UPDATE DB2 BOOTSTRAP2  = DB8WU.DB2.BSDS02
 UPDATE DB2 LOADLIB1    = DB8W8.SDSNEXIT
 UPDATE DB2 LOADLIB2    = DB8W8.SDSNLOAD
*UPDATE DB2 LOADLIB3    =
*UPDATE DB2 LOADLIB4    =
*UPDATE DB2 LOADLIB5    =
*
*-------------------------------------------------------------------
* Sample statements to add/update ARY product plans
*-------------------------------------------------------------------
*
 SET DB2 SSID           = DB8W
 SET PRODUCT CFG        = NULL
 SET PRODUCT VER        = NULL
*
 UPDATE ARY PLAN1       = ARYPLAN1    DISPLAY DATA EXTRACT
 UPDATE ARY PLAN2       = ARYPLAN2    SCHEMA LEVEL REPOSITORY LOAD
 UPDATE ARY PLAN3       = ARYPLAN3    RECOVERY PLAN GENERATION
 UPDATE ARY PLAN4       = ARYPLAN4    JCL GENERATION AND SQL EXEC
 UPDATE ARY PLAN5       = ARYPLAN5    LOG ANALYSIS SERVICES
*
*-------------------------------------------------------------------
* Sample statements to add/update product message library
*-------------------------------------------------------------------
*
```

```
 UPDATE ARY MSGLIBRARY  = ARY.V1R1M0.SARYMENU
*
*-------------------------------------------------------------------
* Sample statements to add/update log services options
*-------------------------------------------------------------------
*
 UPDATE ARY ARCHLOG1    = Y       USE ARCHIVE LOG 1
 UPDATE ARY ARCHLOG2    = N       USE ARCHIVE LOG 2
 UPDATE ARY ACTLOGPRI   = Y       ACTIVE LOG PRIORITY
*
*-------------------------------------------------------------------
* Sample statements to add/update data set prefix generation
*-------------------------------------------------------------------
*
*  The DSN PREFIX maximum length is 17 characters. If NULL
*  is specified then user id is used as data set prefix. Use &USERID
*  in the prefix to insert user id. Example: TEST.&USERID will
*  generate a data set prefix of 'TEST.MYID' where the user id is
*  'MYID'.
*
*UPDATE ARY DSN PREFIX  = xxxxxxxx.xxxxxxxx
*
 UPDATE ARY DSN PREFIX  = NALUR
*
*-------------------------------------------------------------------
*-------------------------------------------------------------------
* Sample statements to add/update schema level repository data
* capture options.
*-------------------------------------------------------------------
*
 UPDATE SLR LOAD BINDS  = Y         N = package/plan versions are
*                                       not saved in schema level
*                                       repository.
 UPDATE SLR LOAD AUTHS  = Y         N = DB2 authorizations are
*                                       not saved in schema level
*                                       repository.
*
*-------------------------------------------------------------------
* Sample statements to add/update interproduct communication
* options.
*-------------------------------------------------------------------
*
 UPDATE IPC IPC_GROUPER = N         Y = Enable Grouper-related
*                                       table recovery.
*
*-------------------------------------------------------------------
* Sample statements to add/update table activity quiet time
* repository names.
*-------------------------------------------------------------------
*
*  EACH QT OWNER/NAME IS A 45 CHAR MAXIMUM LENGTH. THESE OBJECTS WILL
*  BE CREATED AUTOMATICALLY WHEN THE QUIET TIME REPORT/CAPTURE JCL
*  IS RUN IF THEY DO NOT ALREADY EXIST. DDL TO CREATE THESE OBJECTS
*  IS PROVIDED IN ARYDDL7 AND ARYDDL8 SAMPLE DDL MEMBERS.
*
```

```
*                         XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 UPDATE QT GRP TBOWNER  = SYSTOOLS
 UPDATE QT GRP TBNAME   = ARYQTG
 UPDATE QT GRP IXOWNER  = SYSTOOLS
 UPDATE QT GRP IXNAME   = ARYQTGX
 UPDATE QT ENTRY OWNER  = SYSTOOLS
 UPDATE QT ENTRY NAME   = ARYQT
*                         XXXXXXXX (MAX LENGTH FOR DB AND TS IS 8)
 UPDATE QT DATABASE     = SYSTOOLS
 UPDATE QT TABLESPACE   = ARYTSQT
*
*-------------------------------------------------------------------
* Sample statements to add/update log analysis services ROWDATA
* VSAM data set attributes.
*-------------------------------------------------------------------
*
*  The ROWDATA VSAM data set is dynamically created by the log
*  analysis services when creating SQL from the log.       O
*                                                          P
*  The DSN PREFIX maximum length is 21 characters. The following
*  set of product controls are required and must be properly set
*  to ensure proper log data recoveries. The VOLSERS statement
*  value can be set to blanks, if required. A maximum of 3 volsers
*  can be specified.
*
 UPDATE LAS DSN PREFIX  = NALUR
 UPDATE LAS VOLSERS     = OP1TSA,OP1TSB,OP1TSC
 UPDATE LAS DATA AUNIT  = C
 UPDATE LAS DATA PQTY   = 00005
 UPDATE LAS DATA SQTY   = 00005
 UPDATE LAS INDEX AUNIT = C
 UPDATE LAS INDEX PQTY  = 00005
 UPDATE LAS INDEX SQTY  = 00005
*
*-------------------------------------------------------------------
* Sample statements to add/update character conversion information
*-------------------------------------------------------------------
*
*  Schema Level Repository Unicode data conversion information.
*  These values should not be changed. The IBM DB2 Recovery Expert
*  z/OS components require the following CCSID conversions to be
*  defined on the target systems.
*
 UPDATE CCS SLR TECHNQ  = ER            CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS SLR SBCS    = 00037         EBCDIC SBCS CCSID
 UPDATE CCS SLR DBCS    = 01200         UNICODE UT-8 DBCS CCSID
 UPDATE CCS SLR MIXED   = 01208         UNICODE UT-8 MIXED CCSID
*
*  Product output Unicode data conversion information.
*
 UPDATE CCS ARY TECHNQ  = ER            CHARACTER CONVERSION TECHNIQUE
 UPDATE CCS ARY SBCS    = 00037         EBCDIC SBCS CCISD
 UPDATE CCS ARY DBCS    = 01200         UNICODE UT-8 DBCS CCSID
 UPDATE CCS ARY MIXED   = 01208         UNICODE UT-8 MIXED CCSID
*
```

```
*-------------------------------------------------------------------
* Sample statements to add/update default data set information
*-------------------------------------------------------------------
*
*  File tailoring work data set allocation.
*
 UPDATE FTW DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE FTW ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE FTW PQTY        = 00001         PRIMARY QTY
 UPDATE FTW SQTY        = 00001         SECONDARY QTY
*UPDATE FTW SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE FTW SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE FTW SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*
*  Image copy output data set allocation defaults.
*
 UPDATE ICF DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE ICF ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE ICF PQTY        = 00001         PRIMARY QTY
 UPDATE ICF SQTY        = 00001         SECONDARY QTY
*UPDATE ICF SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE ICF SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE ICF SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE ICF MULTIVOL    = xxx
*UPDATE ICF EXPIREDT    = xxxxxxx       EXPIRATION DATE
*UPDATE ICF RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE ICF FILENUM     = xxxx          LABEL FILE NUMBER
*
*  Recovery output data set allocation defaults.
*
 UPDATE RDA DEVICE      = SYSALLDA      DEVICE TYPE
 UPDATE RDA ALCUNIT     = C             C=CYLS, T=TRACKS
 UPDATE RDA PQTY        = 00001         PRIMARY QTY
 UPDATE RDA SQTY        = 00001         SECONDARY QTY
*UPDATE RDA SMSDC       = xxxxxxxx      SMS DATA CLASS
*UPDATE RDA SMSSC       = xxxxxxxx      SMS STORAGE CLASS
*UPDATE RDA SMSMC       = xxxxxxxx      SMS MANAGEMENT CLASS
*UPDATE RDA MULTIVOL    = xxx
*UPDATE RDA EXPIREDT    = xxxxxxx       EXPIRATION DATE
*UPDATE RDA RETPERIOD   = xxxxxxx       RETENTION PERIOD
*UPDATE RDA FILENUM     = xxxx          LABEL FILE NUMBER
*
/*
//
```

F

Additional material

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

Locating the Web material

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

ftp://www.redbooks.ibm.com/redbooks/SG247226

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

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the redbook form number, SG247226.

Using the Web material

The additional Web material that accompanies this redbook includes the following files:

File name	*Description*
SG247226code.zip	Zipped code and configuration files

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space: 2 MB minimum
Operating System: Windows

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder.

Related publications

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

Other publications

These publications are also relevant as further information sources:

- *DB2 UDB for z/OS V8 Administration Guide*, SC18-7413
- *DB2 UDB for z/OS V8 Utility Guide and Reference*, SC18-7427
- *DB2 Recovery Expert for z/OS, V1R1, User's Guide*, SC18-9822
- *z/OS V1R5.0 DFSMSdfp Storage Administration Reference*, SC26-7402-02

Online resources

These Web sites are also relevant as further information sources:

- DB2 Recovery Expert for z/OS

  `http://www.ibm.com/software/data/db2imstools/db2tools/db2re-zos/`

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

V

IBM DB2 Recovery Expert for z/OS User Scenarios

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

IBM DB2 Recovery Expert for z/OS User Scenarios

DB2 for z/OS recovery overview

Recovery Expert for z/OS architecture

User scenarios

This IBM Redbook describes the procedures for using IBM DB2 Recovery Expert for z/OS to manage the recovery of DB2 for z/OS objects in a data sharing and non-data sharing environment. IBM DB2 Recovery Expert for z/OS is an easy-to-use, automated recovery solution that enables database recovery operations with minimal disruption. Its graphical user interface (GUI) provides powerful reporting and automated recovery capabilities for productive database maintenance and high availability. This book is written for an audience of DB2 for z/OS database administrators (DBA).

This book gives you an overview of DB2 Recovery Expert for z/OS, its main features, architecture, recovery semantics, and processing flow. It also describes some of the key considerations in choosing a particular topology to address a business requirement. We provide user scenarios with step-by-step approaches to recover single DB2 for z/OS objects, multiple DB2 for z/OS objects, dropped DB2 for z/OS tables, and DB2 for z/OS subsystems using the IBM Recovery Expert for z/OS tool.

We discuss backup and recovery in a DB2 for z/OS environment, the types of failures that might occur, the types of recovery that are supported, the types of objects that can be recovered, the elements and tools of recovery, and the recovery flow. We also provide a brief tutorial of the DB2 Recovery Expert for z/OS GUI client and an overview of the schema level repository (SLR) with recommendations for its maintenance.

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

SG24-7226-00 ISBN 0738494526