DB2® Server for VSE & VM

Control Center Installation and Operations Guide for VM

Version 6 Release 1

GC09-2679-00

DB2® Server for VSE & VM

Control Center Installation and Operations Guide for VM

Version 6 Release 1

GC09-2679-00

Note!

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

| **First Edition (December 1998)**

This edition, GC09-2679, applies to Version 6 Release 1 of the Control Center feature of the IBM DATABASE 2™ Server for VSE & | VM Program 5648-A70, and to all subsequent releases of this product until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change or addition.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

A form for readers’ comments is provided at the back of this publication. If the form has been removed, address your comments to: IBM Canada Ltd. Laboratory, Information Development 2G/345/1150/TOR, 1150 Eglinton Ave East, North York, Ontario, Canada. M3C 1H7

You can also send your comments by facsimile to (416) 448-6161 addressed to the attention of the RCF Coordinator. If you have | access to Internet, you can send your comments electronically to **torrcf@ca.ibm.com**; IBMLink™, to **toribm(torrcf)**; IBM/PROFS®, to **torolab4(torrcf)**; IBMMAIL, to **ibmmail(caibmwt9)**; home page, on **http://www.software.ibm.com/data/db2/vse-vm/**

If you choose to respond through Internet, please include either your entire Internet network address, or a postal address.

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

Contents

Notices

Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent product, program or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Canada Ltd., Department 071, 1150 Eglinton Avenue East, North York, Ontario Canada M3C 1H7. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

This publication 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.

Trademarks

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

IBM
BookManager
CICS/VSE
DB2
DataPropagator
DATABASE 2
DRDA
IBMLink
IMS
Library Reader
OS/390
OS OPEN
PROFS
QMF
RACF
SQL/DS
VM/ESA
VSE/ESA

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

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

About This Manual

Who Should Use This Manual

IBM Control Center for VSE & VM is a set of database administration tools for IBM DATABASE 2 Server for VSE & VM. This manual is intended for people who want to learn about the Control Center product or who are involved in the evaluation, installation, maintenance, administration, and/or usage of Control Center in a VM environment.

Conventions Used in This Manual

Throughout this document and in the Control Center panel interface the terms *database, database manager, or database server* are used to refer to the DB2 Server for VM. The database or server is composed of a directory, log disk(s), and one or more storage pools.

The Control Center panel interface provides access to, and integrates the functions of, the other product components, along with the job scheduling facility. However, the Operator Commands, Group Authorization Tool (SQLADMIN), Data Restore Functions, DBSPACE Reorganization (SQLREORG), DBSPACE Reorganization Driver Tool (SQLREODR), TABLE Reorganization/Redefinition (SQLTABLE), DBSPACE Maintenance (SQLMAINT), Index Analysis and Maintenance Tool (SQLRINDX), Database Object Search and List (SQLDLIST), and Package Rebind (SQLRBIND) may also be invoked directly; see Appendix G, “Command Mode Interface” on page 523. Related tools are grouped together into chapters to facilitate use of the product and this manual.

The term *Control Center*, unless otherwise specified, refers to Control Center for VM Version 6 Release 1 Modification 0.

Organization of This Manual

The **“Summary of Changes for DB2 Version 6 Release 1” on page xxi** summarizes the changes included in Control Center Version 6 Release 1.0.

Chapter 1, “Introduction” on page 1 introduces the product and tool set and explains how it works with the DB2 Server in a VM environment.

Chapter 2, “Architecture” on page 13 describes the product architecture.

Chapter 3, “Installation and Migration Overview” on page 29 gives preliminary steps for installation and migration.

Chapter 4, “Installing the Service Machine” on page 37 presents the steps for new users to install the service machine.

Chapter 5, “Installing a Support Machine” on page 51 presents the steps for new and existing users to install a support machine. It assumes that the steps outlined in Chapter 4 have been successfully completed.

Chapter 6, "Database Setup" on page 61 presents the steps for new and existing users to set up a database manager to work with the product. It assumes that the steps outlined in Chapter 4 have been successfully completed and that the database has been successfully installed (refer to the *DB2 Server for VM System Administration* manual).

Chapter 7, "Migration" on page 83 describes the steps to migrate from earlier releases of SQL Master to Control Center Version 6 Release 1.0.

Chapter 8, "Corrective Service" on page 91 describes service activities. Information contained in this section is not applicable for initial installation.

Chapter 9, "Data Restore Setup with Control Center" on page 93 describes the steps to set up Data Restore in a Control Center environment.

Chapter 10, "Version Tool" on page 107 describes the version tool which assists DBAs in determining the PTF and code level of options include generating reports and lists for local or remote machines.

Chapter 11, "Managing the Environment" on page 113 provides information necessary to properly manage and maintain the environment including explanations of the Control Center to DB2 Server communication interface and interrupt processing.

Chapter 12, "Getting Started" on page 135 provides you with enough information to get you started in using the product through the panel-mode interface.

Chapter 13, "Group Authorization Tool" on page 149 assists DBAs in managing the access to database objects, simplifies the process of authorization, and shortens the amount of time needed to grant or revoke privileges.

Chapter 14, "Job Scheduling Tool" on page 165 describes the job scheduling tool which may be used to schedule execution of Database Administration tools described in other chapters.

Chapter 15, "Tape Management Tool" on page 185 describes the database tape management tool which supports various tape management methodologies and can be configured to work with one of several available tape management products.

Chapter 16, "Database Operation Interface Tools" on page 197 describes how to use the operator interface tools to issue operator commands to any database server in your VM environment.

Chapter 17, "Database Status Tool" on page 203 describes tools for displaying and changing the status of one or more database servers in your VM environment.

Chapter 18, "Database Startup and Termination Tools" on page 213 describes tools for viewing and changing database startup parameters and for performing immediate or scheduled startup or termination of a database.

Chapter 19, "Database Archiving and Recovery Tools" on page 233 provides detailed information on the automated archiving and recovery tools of the product, including the various supported configurations and options.

Chapter 20, “Data Restore Support” on page 257 gives a general overview of the Data Restore together with Control Center, giving users the ability to automate, manage, track and schedule Data Restore functions on one or more database servers.

Chapter 21, “Data Restore BACKUP” on page 265 describes how to use the Data Restore BACKUP with Control Center to perform a user archive.

Chapter 22, “Data Restore RESTORE” on page 277 describes how to use the Data Restore RESTORE with Control Center to perform a user recovery from BACKUP.

Chapter 23, “Data Restore UNLOAD” on page 289 describes how to use the Data Restore UNLOAD with Control Center to selectively unload DBSPACES.

Chapter 24, “Data Restore RELOAD” on page 297 describes how to use the Data Restore RELOAD with Control Center to selectively restore a table or series of tables, together with forward log recovery.

Chapter 25, “Data Restore TRANSLATE” on page 307 describes how to use the Data Restore TRANSLATE with Control Center to convert a database archive to Data Restore BACKUP format.

Chapter 26, “Data Restore LISTLOG” on page 313 describes how to use the Data Restore LISTLOG with Control Center to list LUW statements from the active and archive logs for selected tables.

Chapter 27, “Data Restore APPLYLOG” on page 317 describes how to use the Data Restore APPLYLOG with Control Center to perform forward log recovery on selected tables.

Chapter 28, “Single User Mode Tools” on page 321 describes the database utility tools that execute on the database virtual machine in single user mode (SUM). The product will, if necessary, automatically change the database to SUM operation from multiple user mode (MUM) operation prior to executing the tool.

Chapter 29, “Database Monitoring Tools” on page 347 describes the automated database monitoring tools. Results from monitoring specific activities, such as resource usage or resource contention, may be gathered over a period of time and reported on or used to automatically invoke tools described in other chapters.

Chapter 30, “DBSPACE Reorganization Tools” on page 367 describes the automated tools for reorganizing DBSPACEs within a database.

Chapter 31, “DBSPACE Reorganization Driver Tool” on page 391 describes the tool which performs multiple DBSPACE reorganizations within a single job submission.

Chapter 32, “ Table Reorganization Tool” on page 405 describes the automated tools for performing table-level reorganization and redefinition. Table redefinition includes copying the table, moving the table to another DBSPACE, and column redefinition.

Chapter 33, “Index Reorganization Tool” on page 419 describes the automated tools for analysis and reorganization of multiple indexes within a database.

Chapter 34, “Automated DBSPACE Maintenance Tools” on page 427 describes the tools for automatically analyzing DBSPACEs and performing maintenance upon them to improve performance. Maintenance includes DBSPACE Reorganization and updating statistics.

Chapter 35, “Rebind Package Tool” on page 449 describes the tool for automatically rebinding one or more database packages.

Chapter 36, “Object Search and List Tools” on page 453 describes the tools for searching and listing DBSPACEs, tables, views, indexes, packages, and columns. Applicable tools described in other chapters may be executed directly against one or more listed objects.

Chapter 37, “Control Center Administration Tools” on page 485 describes a set of special tools to assist the administrator in managing the database server for VM environment. It includes a description of the tool for issuing any command to a database machine from the panel interface and having it execute as if it were entered on a terminal connected to that machine.

Appendix A, “Control Center Messages” on page 497 describes message processing and the *Control Center Message/List Report*.

Appendix B, “Changing Database Name” on page 499 describes the procedure for changing the name of a database machine set up to work with the product.

Appendix C, “Password Support” on page 501 describes the product's support for minidisk passwords.

Appendix D, “Master Scheduling Tool” on page 507 describes the Master Scheduling tool including how to use the Job Scheduling tool to schedule non-Control Center work.

Appendix E, “Authorizations” on page 511 describes the authorization levels and the procedure for changing the default settings.

Appendix F, “User Archiving” on page 515 describes the process for supporting user archives and the steps required for implementation.

Appendix G, “Command Mode Interface” on page 523 describes the command line interface, available commands, and command syntax.

Appendix H, “Tape Hopper Support” on page 545 describes the steps that must be taken for the product to work with a tape hopper.

Appendix I, “Parameter Support” on page 547 describes the product's handling of startup parameters when a database is started in single user mode, and the capability to override startup parameters, to take effect during specified database activities or conditions.

Appendix J, “Database Shutdown Procedures” on page 549 describes the DBFORCE tool, an alternative to SQLEND QUICK for removing active and inactive users.

“Bibliography” on page 553 lists manuals that contain information which may be beneficial when operating Control Center and DB2 Server in a VM environment.

Prerequisite IBM Publications

This manual assumes you have reviewed and understand the IBM manuals for the related VM and DB2 Server for VM products. A working technical knowledge of DB2 Server for VM system administration and database administration is also assumed.

For additional information, see the manuals list in the “Bibliography” on page 553.

Summary of Changes for DB2 Version 6 Release 1

This is a summary of the technical changes to the DB2 Server for VSE & VM Version 6 Release 1 database management system. All manuals are affected by some or all of the changes discussed here. This summary does not list incompatibilities between releases of the DB2 Server for VSE & VM product; see either the *DB2 Server for VSE & VM SQL Reference*, *DB2 Server for VM System Administration*, or the *DB2 Server for VSE System Administration* manuals for a discussion of incompatibilities. Version 6 Release 1 of the DB2 Server for VSE & VM database management system is intended to run on the Virtual Machine/Enterprise Systems Architecture (VM/ESA®) Version 2 Release 2 or later environment and on the Virtual Storage Extended/Enterprise Systems Architecture (VSE/ESA™) Version 2 Release 2 or later environment.

Enhancements, New Functions, and New Capabilities

DRDA® RUOW Application Requestor for VSE (Online)

DRDA Remote Unit of Work Application Requestor provides read and update capability in one location in a single unit of work.

This support provides CICS/VSE® online application programs with the ability to execute SQL statements to access and manipulate data managed by any remote application server that implements the DRDA architecture. Online application programs that access remote application servers need to be preprocessed to create a bind file and then bound (using CBND) to the remote application server. Online application programs that access a local application server are preprocessed as in previous releases.

See the following DB2 Server for VSE & VM manuals for further information:

- *DB2 Server for VSE System Administration*
- *DB2 Server for VSE & VM SQL Reference*
- *DB2 Server for VSE Database Administration*
- *DB2 Server for VSE Application Programming*
- *DB2 Server for VSE Installation*

Stored Procedures

The ability to use stored procedures provides distributed solutions that let more people access data faster.

A stored procedure is a user-written application program compiled and stored at the server. When the database is running in multiple user mode, local applications or remote DRDA applications can invoke the stored procedure. SQL statements are local to the server and issued by a stored procedure so they do not incur the high network costs of distributed statements. Instead, a single network send and receive operation is used to invoke a series of SQL statements contained in a stored procedure.

See the following DB2 Server for VSE & VM manuals for further information:

- *DB2 Server for VM System Administration*
- *DB2 Server for VM Database Administration*
- *DB2 Server for VSE & VM SQL Reference*
- *DB2 Server for VSE & VM Operation*

TCP/IP Support for DB2 Server for VM

TCP/IP support allows:

- VM applications to use SQLDS-private protocol to connect to VM databases over TCP/IP.
- VM applications to use DRDA protocol to connect to DB2 family databases (and any other database that supports DRDA connections) over TCP/IP.
- non-VM applications to use DRDA-protocol to access VM database over TCP/IP.

TCP/IP support for DB2 Server for VM integrated with the DB2 Server for VM application server means a system easier to configure and maintain.

The database manager will optionally secure TCP/IP connections using any external security manager that supports the RACROUTE interface.

New Code Page and Euro Symbol Code Page Support

The following CCSIDs are now supported:

- 1112: Latvian/Lithuanian
- 1122: Estonian
- 1123: Ukrainian
- 1130: Vietnamese
- 1132: Lao
- 1148: E-International
- 1140: E-English
- 1141: E-German
- 1144: E-Italian
- 1147: E-French

Additional support has been added for conversions from Unicode (UCS-2) to host CCSIDs.

For a complete list of CCSIDs supported refer to the *DB2 Server for VM System Administration* and *DB2 Server for VSE System Administration* manuals.

DataPropagator™ Capture

DataPropagator Capture is part of the DB2 Family of DataPropagator products. DataPropagator Capture is updated for Version 6 Release 1 compatibility.

QMF for VM, QMF for VSE, and QMF for Windows®

IBM Query Management Facility (QMF™) is now an separately priced feature of DB2 Server for VSE & VM. QMF is a tightly integrated, powerful, and reliable tool that performs query and reporting for IBM's DB2 relational database Management System Family. It offers an easy-to-learn, interactive interface. Users with little or no data processing experience can easily retrieve, create, update, insert, or delete data that is stored in DB2.

QMF offers a total solution that includes accessing large amounts of data and sharing central repositories of queries and enterprise reports. It also allows you to implement tightly-controlled, distributed, or client-server solutions. In addition, you can use QMF to publish reports to the World Wide Web that you can view with your favorite web browser.

Using QMF, users can access a wide variety of data sources, including operational or warehouse data from many platforms: DB2 for VSE, VM, OS/390® and Windows. Via IBM Data Joiner, you can access non-relational data, such as IMS™ and VSAM, as well as data from other vendor platforms.

RDS Above the Line

The RDS component will load and execute above the 16 megabyte line. This support frees up approximately 1.5 megabytes of storage below the 16 megabyte line (or approximately 2.5 megabytes, if DRDA is installed) when compared to Version 5 Release 1. No installation or migration changes are required for this support to be used (except for the definition of VM Shared Segments and for users who execute the database server with AMODE(24)). If sufficient storage is available, the RDS component will be automatically loaded above the 16 megabyte line. When using VM Shared Segments, the RDS Segment should be defined above the 16 megabyte line.

VM users who wish to run the database server in 24-bit addressing mode (i.e. use the AMODE(24) parameter) **must** use a virtual storage size no greater than 16 megabytes. See the *DB2 Server for VM System Administration* or *DB2 Server for VSE System Administration* for release to release incompatibility information.

Combining of NLS Feature Installation Tapes with Base Product Installation Tape

All available NLS features for DB2 Server for VSE, DB2 Server for VM, Control Center for VSE and REXX SQL for VM have been combined with the respective base product installation tape. Customers interested in an NLS feature language will no longer need to order an additional NLS feature tape because all NLS languages will be available to all customers. In all cases, the default language as shipped is American English. The installation and migration processes have been changed to allow you to choose the default language. Refer to the *DB2 Server for VM Program Directory*, *DB2 Server for VSE Installation*, *DB2 for VSE Control Center Installation and Operations Guide*, and *DB2 REXX SQL for VM/ESA Installation* for the details of how these changes affect the installation process and how you can choose to have a different default language.

Control Center Feature

DB2 Server for VSE & VM Version 6 Release 1 enhances the new Control Center feature as follows:

For both VM/ESA and VSE/ESA:

- Access to the Query Management Facility (QMF)

For VM/ESA:

- Compatibility with DB2 Server for VM Version 6 Release 1 initialization parameters and operator commands
- Shared File System Support (SFS) in a VM/ESA environment
- CA-DYNAM/T Interface Support in a VM/ESA environment
- Data Restore Incremental Backup Support in a VM/ESA environment

For VSE/ESA:

- Control Center code installation on any library
- Ability to use while viewing a list of tables online
- Ability to create, reorganize, unload, reload, move and copy tables in batch mode
- Ability to update table statistics in batch mode
- Ability to drop tables online

Data Restore Feature

The Data Restore feature provides archiving and recovery functions in addition to those provided in DB2 for VSE & VM. Data Restore is enhanced in Version 6 Release 1 with incremental database archiving support. The support allows you to archive only the areas of the database that have been updated since the last database archive, instead of having to archive the entire database. This can provide significant savings for customers with large databases which are updated infrequently, or where only a small fraction of the database is updated frequently.

DB2 REXX SQL Feature

The DB2 REXX SQL feature provides a REXX interface for VM customers to allow SQL calls to be executed from REXX programs. The DB2 REXX SQL feature is updated for Version 6 Release 1 compatibility.

Reliability, Availability, and Serviceability Improvements

Migration Considerations

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

Library Enhancements

Some general library enhancements include:

- The following books have been removed from the library:
  - *DB2 Server for VM Operation*
  - *DB2 Server for VSE Operation*
  - *DB2 Server for VM Interactive SQL Guide and Reference*
  - *DB2 Server for VSE Interactive SQL Guide and Reference*
  - *DB2 Server for VM Database Services Utility*
  - *DB2 Server for VSE Database Services Utility*
- The following books have been added to the library:
  - *DB2 Server for VSE & VM Operation*
  - *DB2 Server for VSE & VM Interactive SQL Guide and Reference*
  - *DB2 Server for VSE & VM Database Services Utility*

Refer to the new *DB2 Server for VSE & VM Overview* for a better understanding of the benefits DB2 Server for VSE & VM can provide.

Chapter 1. Introduction

About Control Center

This section introduces Control Center and its tool set, and provides an overview of the Control Center in a DB2 Server for VM environment.

Product Overview

Control Center is an IBM licensed program which works with the DB2 Server for VM licensed program to automate many of the manual Database Administrator (DBA) functions required to support databases within VM/ESA environments. It automates DBA functions such as database archiving and recovery, adding and deleting dbextents, adding dbspaces, database startup and shutdown, startup parameter changing, dbspace-level reorganization (with candidate selection), table-level reorganization and redefinition, catalog index reorganizations, and database monitoring. Each of these functions can be initiated immediately by an authorized user (local or remote), or they can be scheduled to execute at any specified date and time, or repetitive execution interval.

Control Center simplifies the task of supporting databases by automating the complex steps required to perform many DBA activities. These functions can be scheduled and performed automatically during periods of low system usage, improving the operational productivity of the entire system. Control Center also allows these functions to be performed in a consistent and repeatable manner with a high degree of security and control, thereby reducing the amount of support time.

Once the Control Center environment has been established, new databases can be easily incorporated. Due to the automated control provided by the service machine, the workload impact of additional databases to Database Administrators can be minimized.

Product Benefits

Access Control

Four levels of authorization can be specified controlling access to and use of Control Center. Additionally, these levels can be customized, allowing unique user authorization requirements to be implemented.

Externalized Database Startup Parameters

All database startup parameters are maintained by Control Center and can be updated at any time by the DBA using its user interface. Modified parameters become operational during the next database startup.

Local and Remote User Interface Capability

The user interface was designed to communicate with multiple service machines, both locally on a single CPU and on remote CPUs. The user interface provides both panel mode and command mode capabilities for maximum flexibility.

Database Operator Command Interface

Control Center provides the DBA with the capability to perform all operator commands for a database, either in command mode or through the panel mode interface. These functions include all SHOW commands, the FORCE command, COUNTER command, and RESET command.

Group Authorization Tool

The Group Authorization Tool assists users in managing the access to database objects, simplifies the process of authorization and shortens the amount of time needed to grant or revoke privileges. For more details, see Chapter 13, “Group Authorization Tool” on page 149.

Automated Archiving and Database Recovery

Archives are supported under all logmodes. Special archive features supported by Control Center include multivolume database archives (including User Archives), multiple log archives between database archives, implicit database and log archives, and archives (log and database) to disk.

Recovery functions provide the DBA with all available recovery sets derived from database archive and user archive events, and enable the DBA to select which recovery set should be used and how it should be applied.

The automated interface to the DB2 Server for VM Data Restore Feature lets users automate, track, and schedule the Data Restore functions on one or more databases, see Chapter 22, “Data Restore RESTORE” on page 277. This capability lets users automate, manage, track and schedule the Data Restore functions on one or more databases.

Handling Logmode Changes

When the database is started, all logmode switches are automatically handled. This includes any coldlogs, archives, and log archives that are required to properly restart the database with a new logmode.

Database Monitoring

Many of the periodic monitoring activities performed by the DBA are now automated with the database operator commands. This feature can also be instructed to automatically react to detected conditions. For example, a DBA can enable a log check monitor that will check, at scheduled intervals, the space remaining in a database's log. If the monitor detects that an implicit archive is likely, then it can schedule an explicit log archive for a convenient time in an effort to avoid a possible implicit archive during peak utilization time. Control Center's database monitoring feature also has a report generation capability, that logs database monitor information each time a monitor runs.

DBSPACE Reorganization

Control Center provides a data reorganization tool with the dual advantages of fast execution and high space savings through the DROP DBSPACE command. Execution options also allow for data to be moved within a database (such as to a different storage pool or larger dbspace) or to a different database (for migration or regeneration purposes). Externally stored dbspace data created during the reorganization process can be used for backup purposes. A PAUSE option allows the backup tool to be utilized separately from reorganization.

DBSPACE Reorganization Candidate Selection

Databases are analyzed for DBSPACES that need maintenance, and will optionally perform whatever maintenance is required (reorganization or Update Statistics).

Table Reorganization and Redefinition

A table-level reorganization tool is provided with many options for maximum flexibility. The data-only reorganization option performs a DATAUNLOAD and DATALOAD without dropping the table, which eliminates the need for DBA authority. Additional options allow the table to be dropped and recreated, along with all associated indexes, views, and grants, and for tables to be moved or copied, either within a database or between databases.

This tool also provides the capability to redefine a table definition, allowing columns to be added or deleted, and column characteristics to be changed, including column names, column datatypes, and null attributes.

Index Analysis and Reorganization

The index analysis and reorganization tool provides for automated maintenance of indexes. You can choose analysis only, analysis and reorganization, or reorganization based on a previous analysis. You can also define the scope of analysis and reorganization by specifying a DBSPACE owner, DBSPACE name, or range of pages. You can limit run time by specifying a time or quantity limit.

Object Search and List

This tool is used for searching for and listing database objects such as DBSPACES, tables, views, indexes, packages, and columns. A search can be conducted using a variety of search arguments. Applicable product tools can then be invoked against objects displayed in a list.

Cascading lists of dependent objects can be generated for an object within a currently displayed list. For example, a list of indexes can be displayed for any table selected within a list of tables.

Job Scheduling

The job scheduling tool provides the capability to schedule concurrent or single-threaded processing events against one or more databases. Detailed event control information provides the capability to establish dependencies between jobs, a dependency on database status (up or down), a dependency on a particular service machine and a dependency on probable completion within a designated timeframe. This allows for better control and automation of both System Administration tools (such as archiving and adding dbextents) and Database Administration tools (such as DBSPACE reorganization).

Product History

Control Center for VM, originally called SQL Master for VM (5684-136), available since June 1990, was designed by database administrators responsible for managing and maintaining large SQL/DS databases for IBM.

Product Requirements

Control Center requires an environment provided by these operating system(s):

- IBM Virtual Machine/Enterprise Systems Architecture (VM/ESA) Version 2 Release 2 (5684-030), or later

The following level of DB2 Server for VM is required:

- DB2 Server for VM Version 6 Release 1 (5648-A70)
  - DB2 Server for VM Version 5 Release 1 (5648-158) is also a supported environment
  - SQL/DS Version 3 Release 5 (5688-103) is also a supported environment

Control Center supports environments without any tape management system, sending tape mount request messages directly to a defined tape operator user ID. It also provides support for CA-DYNAM/T (or DYNAM/T) , VMTAPE, and EPIC, if they are used on the target system. In addition, with minor customization, it can easily be adapted to other tape management products.

Control Center and the Database Environment

A product installation includes a Control Center service machine, which can manage one or more databases, and optionally, one or more support machines. User machines linked to the service machine are given proper database authority. This enables communication with the databases managed by the service machine whether on the same CPU node or a remote node.

Service Machine

The service machine executes on a disconnected virtual machine (referred to as the Control Center service machine) which communicates directly with one or more databases. Normally in an idle mode, the service machine will, upon interruption, initiate work such as a database archive; respond properly to specific database, CP and CMS prompts and messages; track the status of all databases; and communicate with one or more DBAs in your information systems organization. It has the authority and know-how to perform simple and quick one-step operations as well as complex multi-step operations which can span several hours.

Distributing Work with Support Machines

Database Administration tools such as Multiple User Mode DBSPACE Reorganization execute on a virtual machine separate from the database machine and can be initiated for immediate execution or scheduled to execute at any specified date and time, or repetitive execution interval. Depending on the tool, the size of the database, and other factors, the activity may take several hours to complete.

Control Center is able to distribute work by scheduling Database Administration tools to execute on one or more support machines where they will execute independent of the service machine and independent of each other. Overall performance and throughput (Database Administration tools can execute

concurrently against different databases) is improved, and it eliminates the need to have separately installed service machines for each installed database.[1]

Sample Environments

The following two scenarios help to illustrate how Control Center manages local and remote databases in a VM environment.

Control Center in a Single-CPU/Multi-Database Environment

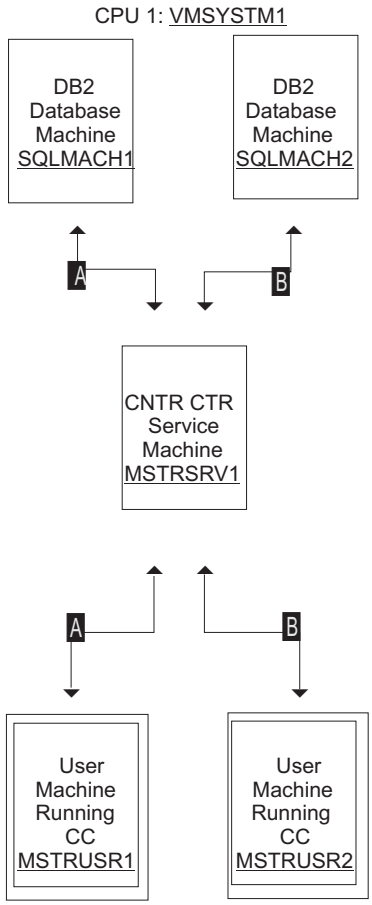


Figure 1. Control Center in a Single-CPU/Multi-Database Environment

Shown in Figure 1 is a single-CPU VM environment where two databases are being managed by a single service machine. Two users linked to the service machine and having proper authority are able to perform functions against the database machines as if they were logged on to the machines directly.

For example, user MSTRUSR1 may be issuing operator commands to SQLMACH1 and reviewing the results (communication path **A**) while user MSTRUSR2 is

[1] In practical terms there is a limit to the number of databases a single service machine can manage.

initiating a single user mode DBSPACE reorganization on SQLMACH2 (communication path **B**).

Control Center with Support Machine in a Multi-CPU/Multi-Database Environment

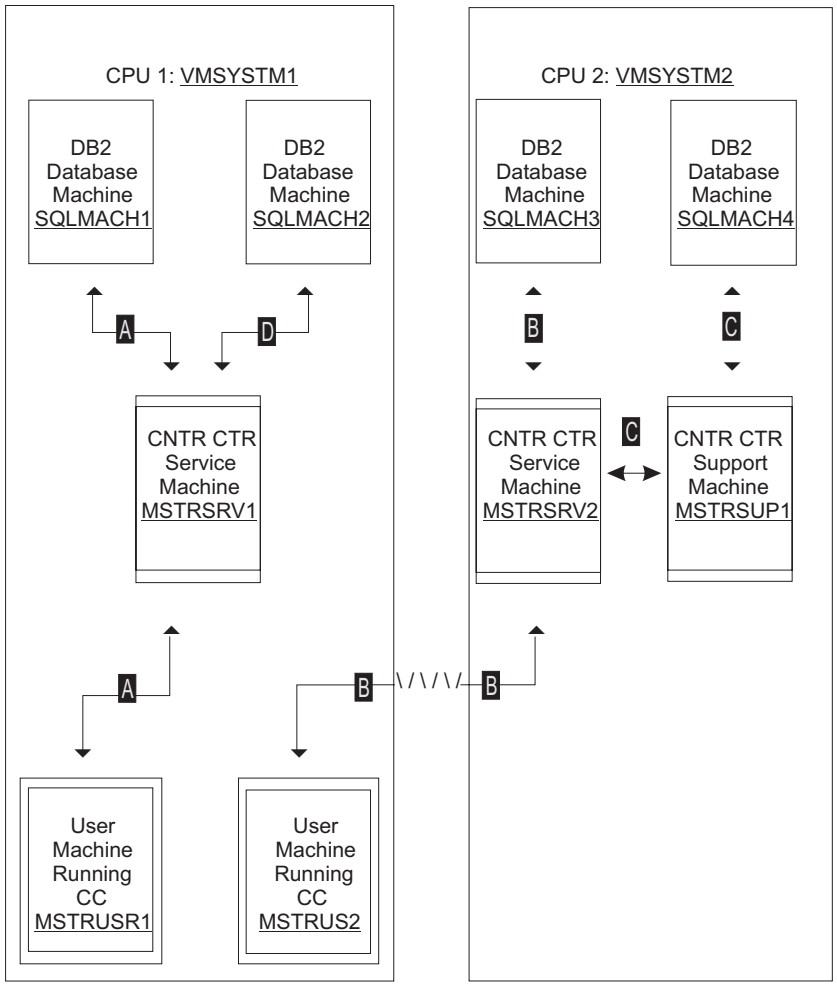


Figure 2. Control Center with Support Machine in a Multi-CPU/Multi-Database Environment

Shown in Figure 2 is a complex multi-CPU environment where two service machines and one support machine are managing the activities of four databases.

As an example, user MSTRUSR1 may be issuing operator commands to database SQLMACH1 (communication path **A**) while concurrently service machine MSTRSRV1 is running a scheduled database archive of database SQLMACH2 automatically from start to finish (communication path **D**). User MSTRUSR2 may be initiating a database recovery from tape for database SQLMACH3 on remote node VMSYSTM2 (communication path **B**) while the same managing service machine has initiated a table reorganization process to execute on support machine MSTRSUP1 (communication path **C**).

Control Center Tools Overview

There are three types of Control Center tools: System Administration, Database Administration, and Control Center Administration. The System Administration and Database Administration tools work directly with database environments, whereas Control Center Administration tools help you to utilize the product as a whole. It is important for you to fully understand these three types of tools and how they work.

System Administration Tools

System Administration tools work with the database virtual machine consoles. They manage the operational needs of the database virtual machines such as:

1. Starting and stopping of the database
2. Database archives and recoveries
3. Adding and deleting dbextents, moving dbextents
4. Adding dbspaces
5. Specification of database startup parameters
6. Interfacing the database with VM
7. Interfacing the database with tape managers
8. Management of database virtual machine consoles
9. Database operational monitoring and reporting and,
10. Any other activity that runs on the database's console

How the System Administration Tools Work

Before using any of the System Administration tools you must have a basic understanding of how these tools work. These tools work very differently from the Database Administration tools and it is important that you understand the differences in order to avoid incorrect use of the Control Center product.

The System Administration tools all share the same basic operational characteristics; each of these tools is involved either directly or indirectly with the management of activities that run on database virtual machines. These tools do not in any way work with database data or data-related objects, and therefore, they do not require any of its connect authorization.

Each of these tools manage database generated console messages. These messages can be generated on a database virtual machine console during either single user or multiple user modes of operation. Many of these messages are database prompts for information or resources that must be responded to before processing can continue. During the running of a database archive, for example, many messages and prompts for resources (tape drives, or disks) and information (when to start, or what address to archive to) are displayed on the database's console. It is the System Administration tools related to database archiving that manage the archive activity by responding to each of the database's console messages and prompts.

These tools respond to every message generated on a database virtual machine console. This includes messages generated by the database, VM (CP or CMS),

tape management products (VMTAPE, DYNAM/T , EPIC, and other), or by any message-generating program or service. All messages are routed from the database virtual machine to the Control Center service machine through the VM-provided Single Console Image Facility (SCIF) messaging service. SCIF enables the service machine to become the database machine's virtual console and keyboard, as if the service machine were logged on to the database virtual machine itself. This enables the service machine to view all messages as they appear on the database console, as well as directly feed input to the database's console.

When the service machine is routed a message through SCIF, Control Center determines what the message is related to (such as archive, or addition of a dbextent) and invokes the appropriate System Administration tool to review the message and determine if any information or resources are required by the database machine. Any required feedback is routed back to the database's console through the SCIF service.

Where the System Administration Tools Run

Except for the Direct Operator Command Interface tool, System Administration tools run on the Control Center service machine. It is the service machine that invokes tools as required to manage a database machine's console. These tools have been designed to support multiple database machines simultaneously. This means that the service machine does not become locked into and dedicated to the management of a single database machine, but rather remains available to respond to other database machines as required. Control Center, for example, can manage the processing of multiple database archives occurring on multiple database machines simultaneously.

List of System Administration Tools

Table 1 (Page 1 of 2). System Administration Tools

Tool Name	Database Connect Authority Required	Control Center Authority Required	Control Center Multiple User Mode or Single User Mode	Managed by Control Center or Runs on User's Machine
Database Archiving and Recovery Tools.				
Database Archiving	No	Yes	SUM/MUM	Control Center
Database Recovery	No	Yes	SUM/MUM	Control Center
Database Monitoring Tools.				
Database Log Check	No	Yes	MUM	Control Center
Database Up and Running Check	No	Yes	SUM/MUM	Control Center
Users Active	No	Yes	MUM	Control Center
Users Connected	No	Yes	MUM	Control Center
Users Locking	No	Yes	MUM	Control Center
DBSPACE Usage	No	Yes	MUM	Control Center
Database Counters	No	Yes	MUM	Control Center
Reset Counters	No	Yes	MUM	Control Center
POOL Usage	No	Yes	SUM	Control Center
Database Startup and Termination Tools.				

Table 1 (Page 2 of 2). System Administration Tools

Tool Name	Database Connect Authority Required	Control Center Authority Required	Control Center Multiple User Mode or Single User Mode	Managed by Control Center or Runs on User's Machine
Database Parameters	No	Yes	SUM	Control Center
Database Startup	No	Yes	SUM	Control Center
Database Termination	No	Yes	SUM	Control Center
Database Status Tool.				
Database Status	No	Yes	SUM	Control Center
Database Operation Interface Tools.				
Operator Command Interface	No	Yes	MUM	Control Center
Direct Operator Command Interface	Yes	No	MUM	User's Machine
Single User Mode Tools.				
Add DBEXTENTS	No	Yes	SUM	Control Center
Delete DBEXTENTS	No	Yes	SUM	Control Center
Copy/Expand Database Directory	No	Yes	SUM	Control Center
Copy/Move Database Log Disk	No	Yes	SUM	Control Center
Copy/Move Data Disk	No	Yes	SUM	Control Center
Add DBSPACE	No	Yes	SUM	Control Center
Database COLDLOG	No	Yes	SUM	Control Center
Reorganize Catalog Indexes	No	Yes	SUM	Control Center

Database Administration Tools

Database Administration tools are database application programs, and therefore require DBA authority. These tools do *not* work with or manage the database virtual machine. They are programs that connect to a database and perform data retrievals and updates using database queries while the database is running in multiple user mode (SYSMODE=M). Since these tools require the database to be in multiple user mode (with the exception of the Single User Mode Reorganization tools), they are sometimes referenced in this documentation as multiple user mode tools.

How the Database Administration Tools Work

The virtual machine that runs these tools must have database DBA authority. (An exception to this is the Table Reorganization and Redefinition tool, which has one option that only requires connect authority). When invoked, the tool runs a database application program that connects to the database. The program then runs database queries to manage (reorganize, backup, redefine, or migrate) data and data-related objects.

Where the Database Administration Tools Run

Unlike the System Administration tools, the Database Administration tools require a dedicated user machine console. This means that during the running of a Database Administration, the virtual machine running the tool cannot do any other processing. For example, when you run the Multiple User Mode DBSPACE Reorganization tool on your virtual machine you will not be able to perform any other work until the tool completes processing. This is contrasted with the System Administration tools, which can manage the processing of multiple database console activities simultaneously.

You should never run a Database Administration tool on the service machine, because these tools require a dedicated virtual machine console. If the Control Center service machine runs this type of tool, then it will not be available to respond to database machine console messages and prompts. Therefore, these tools should be run on virtual machines that can afford to dedicate their consoles to the processing of a single activity. The Control Center support machines should be installed and used to run these tools.

When scheduling Database Administration tools, be sure to specify a support machine (or other non-Control Center service machine) that will run the tool. **If you do not specify a support machine, the tool will run on the service machine, making it unavailable to process any other requests.**

Important Note:

A deadlock condition can occur on the service machine, if it runs a Database Administration tool that causes console messages to be generated on the database machine. A deadlock results when the service machine cannot complete tool processing until database machine console messages are responded to and; console messages cannot be responded to until the tool completes processing.

For example, if the service machine runs a reorganization tool, (a Database Administration tool) that reorganizes a large DBSPACE, the database log could fill up (due to unload/reload reorganization work). When the log fills up, the database will initiate an archive activity (logmode A or L) that the Control Center service machine must manage by responding to database console messages and prompts to start the archive, mount tapes, and so on. The deadlock occurs because the service machine is reorganizing and cannot respond to the database console prompts. The reorganization waits because the database halts all activity until log space can be freed by the taking of an archive which will not run until the console prompts are responded to.

List of Database Administration Tools

Table 2. Database Administration Tools

Tool Name	Database Connect Authority Required	Control Center Authority Required	Control CenterMultiple User Mode or Single User Mode	Managed by Control Center or Runs on User's Machine
Automated DBSPACE Maintenance Tools.				
DBSPACE Reorganization Maintenance	Yes	No	MUM	User's Machine
Statistics Maintenance	Yes	No	MUM	User's Machine
Index Reorganization	Yes	No	MUM	User's Machine
DBSPACE Reorganization Tools.				
Multiple User Mode DBSPACE Reorganization	Yes	No	MUM	User's Machine
Single User Mode DBSPACE Reload	Yes	Yes	SUM/MUM	Control Center
Single User Mode DBSPACE Reorganization	Yes	Yes	SUM	Control Center
DBSPACE Reorganization Driver	Yes	No	MUM	User's Machine
Table Reorganization and Redefinition Tool.				
Table Reorganization and Redefinition	Yes	No	MUM	User's Machine
Object Search and List Tools.				
DBSPACE Search and List	Yes	No	MUM	User's Machine
Package Search and List	Yes	No	MUM	User's Machine
Table Search and List	Yes	No	MUM	User's Machine
Table Index List	Yes	No	MUM	User's Machine
Table View List	Yes	No	MUM	User's Machine
Rebind Package Tool.				
Rebind Package	Yes	No	MUM	User's Machine

Control Center Administration Tools

Control Center Administration tools do not work directly with a database environment, but rather work in support of service and support machines. For example, the Job Scheduling tool enables the product to manage the processing of jobs. These tools have been designed to work with and enhance the usability of System Administration and Database Administration tools. You can use the Job Scheduling tool to schedule database archives (a System Administration tool), as well as dbspace reorganizations (a Database Administration tool).

In addition to supporting the Control Center environment, these tools also allow you to manage the related service and support machines. Management of these types of machines includes activities such as starting and stopping Control Center, authorizing new users, and listing and reviewing control files.

How the Control Center Administration Tools Work

These tools are invoked by the service and support machines as required. You can access them through the menu interface. Each of these tools has been designed to minimize the use of Control Center service and support machine consoles. When run, they require only brief periods of service and support machine processing, keeping the machine consoles available for the servicing of database-related activities.

Where the Control Center Administration Tools Run

Control Center Administration tools run on the service and support machines. As mentioned above, they have been designed to minimize the use of the console activity, leaving the consoles more available for use by the System Administration and Database Administration tools.

List of Control Center Administration Tools

Table 3. Control Center Administration Tools

Tool Name	Database Connect Authority Required	Control Center Authority Required	Control Center Multiple User Mode or Single User Mode	Managed by Control Center or Runs on User's Machine
Job Scheduling Tools.				
Job Scheduling	No	Yes	n/a	Control Center
Job Schedule List	No	Yes	n/a	Control Center
Master Scheduling	No	Yes	n/a	Control Center
Control Center Administration Tools.				
View Message Log	No	Yes	n/a	Control Center
Issue Database Commands	No	Yes	n/a	Control Center
List Control Center VM Database Files	No	Yes	n/a	Control Center
Query DBEXTENTS/STORPOOL Mapping	No	Yes	n/a	Control Center
Control Center Database Tape Management Tool.				
Tape Management	No	Yes	n/a	Control Center

Chapter 2. Architecture

This section describes the Control Center product architecture. It is intended primarily for those people involved in the installation, maintenance, or administration of Control Center, and users of the product.

This section is prerequisite reading for the material presented in Chapter 3, “Installation and Migration Overview” on page 29, and Chapter 11, “Managing the Environment” on page 113.

Overview

The product architecture has been designed to allow you to install a Control Center configuration that is tailored to meet your specific database support requirements. It is easily customized because of a modular design and externalized control parameters that allow you to be very specific in identifying how you want your databases managed. For example, tape management processes are often quite unique to specific VM installations, requiring that the management (for instance, mounting or cataloging) be handled in a particular manner. When installing, you can instruct the product how to manage tapes and tape-related information required by your installation. In addition to tape management, Control Center can be customized in other ways to help you handle your database environment.

Related Virtual Machines

In Control Center configuration, there are five basic types of virtual machines: service machines, database machines, support machines, user machines, and Control Center owner machines.

Figure 3 on page 14 shows a basic Control Center configuration with a service machine (MSTRSRV), two support machines (MSTRSUP1, MSTRSUP2), two database machines (SQLMACH1, SQLMACH2), two user machines (MSTRUSR1, MSTRUSR2), and a single owner machine (MSTROWN).

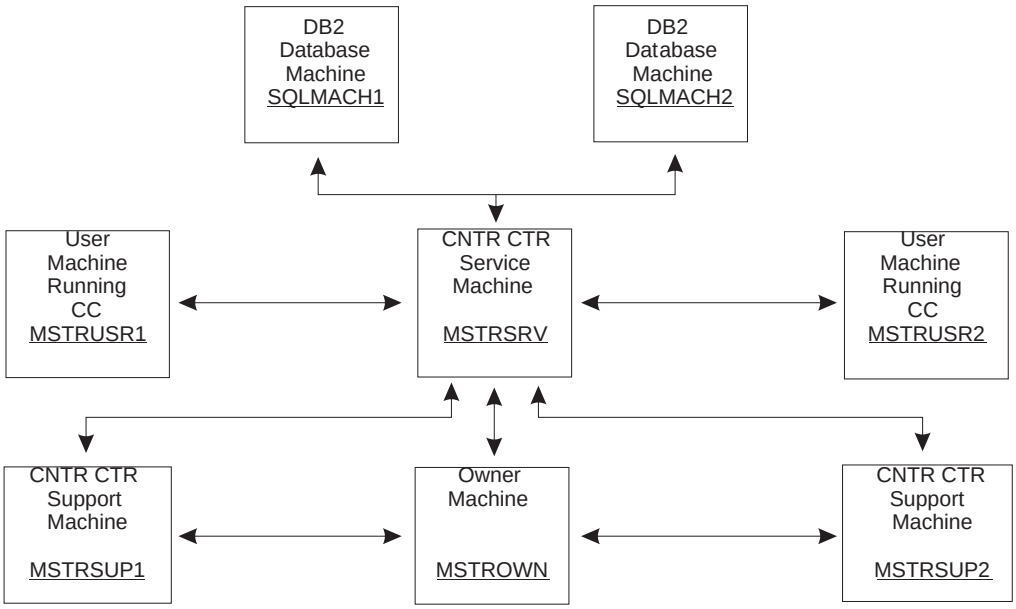


Figure 3. Control Center Machine Types

Service Machine

This is a machine that runs in disconnected mode (no physical console or keyboard attached) and is responsible for the automated operations of the database environment. This is the central machine that processes your requests as a user of Control Center, as well as manages the activities of support machines and database machines.

Support Machine

The MSTRSUP1 and MSTRSUP2 machines in the figure are support machines. These machines are controlled and managed by a service machine (MSTRSRV) while running in a disconnected mode (no physical console or keyboard attached). The service machine will send these machines long-running work activities that require dedicated processing. These long-running activities are often referred to as multiple user mode (MUM) applications, because these applications connect to a database that is running in a multiple user mode (SYSMODE=M).

Database Machine

The SQLMACH1 and SQLMACH2 machines in Figure 3 are database virtual machines. These machines are your database application servers that take requests for data or data updates from other virtual machines. These machines own the minidisks that your database data is stored on.

User Machine

The MSTRUSR1 and MSTRUSR2 machines are user machines, each with a physical console and keyboard attached. This is the type of machine that you use to interface with the Control Center and DB2 for VM Server products. These machines must be authorized to Control Center, as well as to the database itself.

As an authorized user, you can connect to a database and run MUM activities, as can a support machine. You can, however, prefer to have your longer-running MUM activities processed on a support machine so as to keep your virtual machine free for other processing activities.

Owner Machine

These machines are owners of particular service and support machines. These machines are specified during the installation of a service machine or support machine. The MSTROWN machine shown represents an owner virtual machine for the MSTRSRV service machine and the MSTRSUP1 and MSTRSUP2 support machines.

The owner machines are given Control Center Administrator level authorization. These machines will be sent Control Center machine consoles, if the Control Center machine is spooling a console, each time a service machine or support machine is cycled (stopped and restarted, IPLed, or Control Center midnight processing). They are designated as owner machines in the Control Center PROFILE file.

Control Center Tools and the Virtual Machines They Run On

As covered in “Control Center Tools Overview” on page 7, the product is comprised of three types of tools: System Administration, Database Administration, and Control Center Administration tools.

The virtual machines outlined above each use a specific type of Control Center tool. It is important, therefore, that you review and understand the “Control Center Tools Overview” on page 7 section of this manual before proceeding any further in this section.

System Administration Tools on the Service Machine

The System Administration tools are used to manage the operations of your database virtual machine consoles. In other words, these tools automate all the activities that would normally be performed by a database console operator logged directly onto the database's virtual machine.

It is the purpose of the service machine to invoke, run, and manage the System Administration tools, automating console operations for one or more databases.

How the System Administration Tools Work On the Service Machine

The service machine runs the System Administration tools in a disconnected mode, meaning that there is no physical console or keyboard attached to these machines while they are running. These machines are in a sleep (trivial CPU usage) mode until a message or file is sent to them, causing an interrupt of their sleep state. Upon interruption, the service machine "wakes up" and runs System Administration tools to review the message or file to determine what, if any, action should be taken.

Scheduled Processing

The System Administration tools are not just reactive to interrupts, but can be scheduled by you for specific invocation. For example, you can schedule (the

scheduler is a Control Center Administration tool) a database archive activity to occur at a later scheduled time. At the specified time, the archive System Administration tool will be invoked and the archive activity will be started. As the archive is processed, the database's console will then generate many messages that may require responses, and it is the job of the System Administration tools to provide the database's console with any required information.

Asynchronous Processing

A single service machine running the System Administration tools can support one or more database machines simultaneously. This means that multiple database console activities can be processed by Control Center at the same time. For example, MSTRSRV could start a full database archive for the SQLMACH1 database and at the same time start an add dbextent process on the SQLMACH2 database machine. MSTRSRV will respond to each database console message as it occurs and does not become locked into the processing of one specific database's console messages. Both activities can proceed simultaneously on each database machine.

Simultaneous Support of Multiple Database Versions

The System Administration tools running on the service machine support the simultaneous running of different versions of DB2 Server for VM therefore, you will not have to reinstall or reconfigure your service machine when you move to new versions.

Database Administration Tools on Support and User Machines

As described in “Database Administration Tools” on page 9, the Database Administration tools are very different from the System Administration tools. These tools require a dedicated virtual machine console while running, which means that multiple Database Administration tools cannot run on the same virtual machine at the same time.

These tools require a dedicated console because they are database application programs. These tools work through the database to manipulate actual database data, compared to the System Administration tools, which do not perform database connects. They are often referred to as multiple user mode (MUM) applications, because they connect to a database that is running in multiple user mode (SYSMODE=M).

Why Support Machines?

Use support machines to invoke and process Database Administration jobs that run for long periods of time (usually anything greater than a few minutes). Support machines run in disconnected mode and can be instructed to invoke and execute Database Administration tools. Install one support machine per database machine. This gives you the capability to run simultaneous Database Administration tools against one or more databases. For example, you could instruct the MSTRSRV machine to have the MSTRSUP1 machine run a reorganization of a DBSPACE in the SQLMACH1 database, and at the same time start another reorganization on the MSTRSUP2 machine for the SQLMACH2 database.

Managing Support Machines

The service machine will manage the support machines. When initiating or scheduling Database Administration tools to run on support machines, you interface with the service machine and give it instructions that drive the tools' execution on the support machine.

The service machine processing of an activity on a support machine from invocation to completion. It is the responsibility of the service machine to track the activity of each of its support machines, as well as to maintain a status of each of the activities performed.

Why Not on Service Machines?

When executing, Database Administration tools require a dedicated virtual machine. This means that the virtual machine on which these tools are running is unavailable until execution is completed. When a service machine runs a Database Administration tool, it is unavailable to meet its primary responsibility of servicing database consoles. Furthermore, a potential deadlock situation could be created (refer to the “Control Center Tools Overview” on page 7).

Control Center Administration Tools on Service and Support Machines

As described in the “Control Center Administration Tools” on page 11, these tools provide the service machines and support machines with some basic operational capabilities. For example, the Job Scheduling tool provides service machines with the ability to manage the processing of either service machine jobs (archives, recovers, add dbextents), or support machine jobs (DBSPACE reorganizations, table reorganizations).

These tools do not require their own virtual machines, but rather are invoked and run by the service machines and support machines as required. Code for these tools is located on the Control Center product code disk, that both the service machine and support machine must have read access to (refer to the code disk discussion later in this section).

Control Center Communications

The Control Center tools run on various types of virtual machines. Each of these machines contributes to the processing of the database environment, and therefore must be able to communicate with each other. This section will cover basic types of communications relating to the five types of Control Center machines previously discussed.

Three Types of Communications

There are three different types of communications relating to your configuration:

1. Single Console Image Facility (SCIF)
2. Messages and files
3. Database connects

Figure 4 on page 19 is a diagram of Control Center related virtual machines and the communication connections used between them.

SCIF Communications (Database to Service Machine)

The service machine communicates with database machines through the Single Console Image Facility (SCIF). In Figure 4 on page 19, the MSTRSRV machine is connected using SCIF with SQLMACH1 and SQLMACH2 database machines. There is no limit to the number of SCIF connections a service machine can have, but each database can have only one.

Important:

You must not identify another machine as a secondary console for a service machine. If you have database machines SCIFed to your service machine and your service machine SCIFed to a third machine, then any database console messages will, in effect, be SCIFed directly to the third machine, bypassing the service machine entirely.

SCIF is a standard component of VM that enables the service machine to act as the virtual console and virtual keyboard for each database machine it is connected to, as if the service machine were actually logged onto the database virtual machine(s). The service machine, through the SCIF connection, can issue commands directly to the database, as well as view all database console messages that are displayed on the database's console.

Breaking the SCIF Connection

The SCIF relationship will exist while the database and service machine are running in disconnected (no physical console or keyboard attached) mode. If you log onto the database machine, the SCIF connection to the service machine will be broken. Any messages produced on the database console while you are logged on will not be forwarded by SCIF to your service machine.

Logging onto the service machine will not break the SCIF connection (messages will still be forwarded and stacked for processing) but you will not be able to communicate with the service machine using the interfaces provided (SQM command mode or panel interface).

Number of Service Machines Required

It is necessary to have at least one service machine per CPU, because the SCIF facility cannot communicate across CPUs. There is no limit on the number of database virtual machines connected via SCIF to your service machine. You are not, however, limited to installing just one service machine per CPU. You can install as many service machines as are necessary to meet your specific requirements. Additional machines, for example, can be required to segregate different platforms (production, development, test) from each other, or to provide more responsive database console management.

If you have a database console environment that is very active (many console messages from archives, operator commands), and your service machine seems to be performing sluggishly, then you can install additional service machines to distribute the console processing activity across multiple machines. This can be especially true if you decide to run several database monitors (refer to Chapter 29, “Database Monitoring Tools” on page 347) with very small frequency (1 to 5 minute) intervals. This is because each monitor will cause the generation of

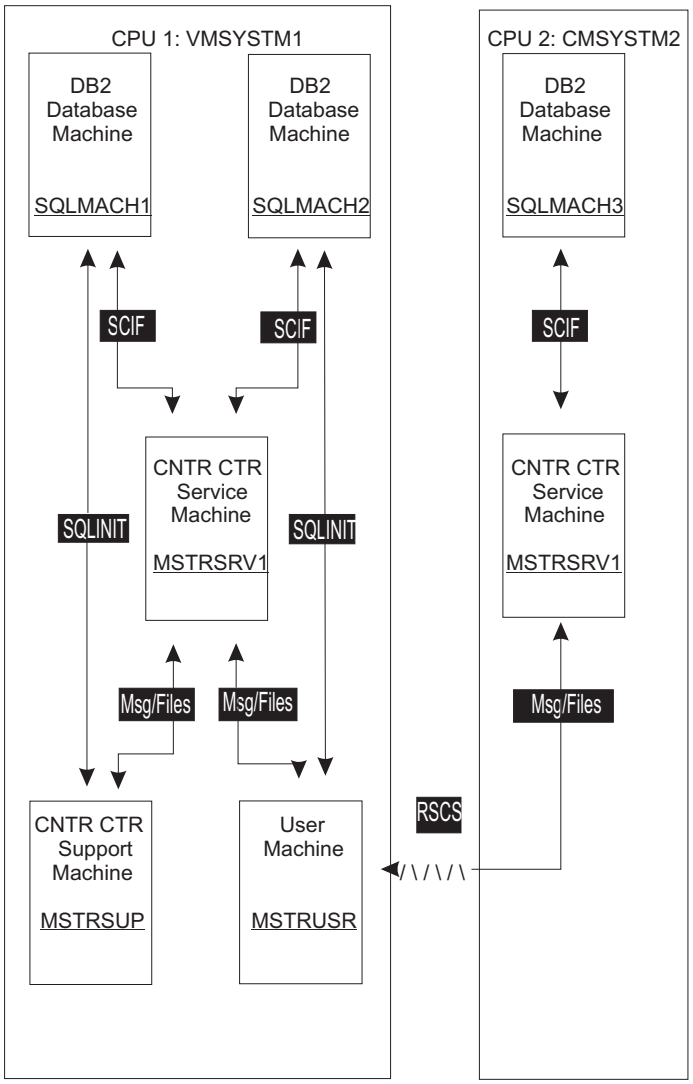


Figure 4. Control Center Communication Connections

additional database console messages that your service machine will have to analyze and react to.

Message and File Communication

As Figure 4 shows, the service machine uses messages and files to communicate back and forth with user machines and support machines. Interfaces running on the machines receive and interpret the messages and files to determine what action or what information is to be displayed.

If the MSTRUSR machine in the diagram requests information about a database (the SHOW commands), the request is transmitted to the MSTRSRV1 machine in the form of a message and the response is sent back to MSTRUSR in the form of a file. The file sent back to MSTRUSR will appear in the MSTRUSR's reader. The Control Center interface will PEEK the file, displaying it to MSTRUSR. After MSTRUSR exits the file display, the file will be purged from the reader. If MSTRUSR wants to save or print a copy of the information, a SAVE or FILE

command while in PEEK (placing the file onto MSTRUSR's 191 A-disk) can be issued.

Waiting for Requested Information

When a request for information is sent to Control Center through the SQM interface, the user will be placed in a WAIT state (using WAKEUP) until the information is returned to the user's reader. There is a predefined length of time that the SQM interface will wait for the reader file. After this amount of time, SQM will return control to the user and indicate that the expected file did not arrive. This problem will usually indicate that Control Center failed or that a network connection from the user to the Control Center machine is not available.

Operator Communication

There are four basic modes of communication with Control Center. Any of these modes can be used by any virtual machine attempting to communicate with Control Center:

- Panel
- Command
- Remote
- CMS

Panel Mode

Communicating with the Control Center facility via the panel interface is done through the SQM EXEC. SQM is invoked in panel mode by entering SQM without any parameters.

Command Mode

The SQM interface can be invoked in command mode to issue single-line commands to a service machine. This mode can be very useful when it is required to communicate with a service machine in a single-line mode format, such as issuing communications from a program (EXEC) or server machine.

CMS Mode

Files can be sent directly to service machines. If the sender is authorized, the file will be received by the service machine onto either its 191 A-disk or the code disk, depending on the type of file sent.

Remote Mode

Communications to a remote CPU service machine are done through the Remote Spooling Communications Support (RSCS) network product. If you have multiple CPUs, then communications to a service machine on another CPU will only be possible if the remote CPU has been identified to RSCS. When communicating with the remote service machine, all messages and files are routed through RSCS directly to the remote service machine; the local CPU service machine is not involved in the communications.

Database Connection

The support machine (MSTRSUP) and user machine (MSTRUSR) run the Database Administration tools. These tools will perform a database connect to a specified database (SQLMACH1 or SQLMACH2) by running the database-provided SQLINIT utility. This utility is available with all versions of DB2 Server for VM and provides applications with the ability to establish communications with a database.

To successfully connect to a database, support machines and user machines must have database DBA connect authorization. This authorization is maintained by DB2 Server for VM for each database machine. In Figure 4 on page 19 the MSTRUSR machine must be granted specific connect authorization to each database that the machine will be accessing.

Important:

The connect authorizations are maintained by the database and are completely separate and distinct from the Control Center product authorizations.

Minidisk Layout

During Control Center installation you will be prompted for the virtual device addresses of the Control Center code disk and tape management code disk. This information will be used by Control Center when establishing your virtual machine environments in preparation for activities to be performed.

Code Disk

The Control Center code disk shown in Figure 5 on page 23 contains only Control Center product code (this includes the DBINIT CONTROL and SQLMSTR DIRECTRY files). Each of the machines shown has read access to the code disk, with the service machine (MSTRSRV) having read and write access. Since MSTRSRV has write access to the code disk, this machine is referred to as the code disk owner.

Each of the virtual machines has its own read/write 191 A-disk. The service machines and support machines store various control files which are updated and stored on their A-disks. Information used by the database machines during startup and termination is kept on each database machine's A-disk. The user machine (MSTRUSR) A-disks will be updated with certain control information that will be used by the Control Center panel and command mode interfaces, as well as by any Database Administration tools that the user machine invokes.

Communications Code: The Control Center code disk contains all code needed for invoking, using, and communicating with the Control Center tools. In addition, information detailing your local and remote CPUs is stored on this disk (refer to "SQLMSTR DIRECTRY and DBINIT CONTROL Files" below).

Automatic Code Disk Reaccessing and Shared Code Disks: The service machine and support machines should use a common (shared) code disk. Since only one of the machines (must be a service machine) can own (read/write) access, then the other machines will only have shared (read) access.

In support of Control Center shared code disk access, when the code disk is updated by the owning service machine, the service machine will automatically instruct other service machines and support machines to reaccess the changed code disk. In addition, any users that are currently linked to the disk will be notified that the code disk has been changed and should be reaccessed.

The service machines and support machines will reaccess the disk automatically, provided that the owner service machine has been authorized as a Control Center Administrator to each of these machines. This feature is especially important in regard to the SQLMSTR DIRECTRY and DBINIT CONTROL files, which contain common control information that is used during Control Center communications.

SQLMSTR DIRECTRY and DBINIT CONTROL Files: These files provide basic communications control information that must be made available to all virtual machines (service machine, support machine, database machines, and user machines) that will use the Control Center tools. These files contain specific communications control information particular to your installation (Chapter 11, “Managing the Environment” on page 113). Since the information must be made available to all users of the Control Center product, these files are stored on the Control Center code disk.

When updating files that normally reside on the Control Center code disk, you can be instructed to send a copy of the files to the service machine that has write access to the Control Center code disk. If you are authorized (Control Center Administrator authority) to update the Control Center code disk, then you can update the disk by sending code files (includes SQLMSTR DIRECTRY and DBINIT CONTROL) to the service machine that owns the code disk. The service machine will automatically receive any code files to the code disk.

Database Production Code Disk

This disk contains the database product code. In order to communicate and use a database, the user and support machine must have read access to this disk. This disk must also be available to the database machine, which must have the ability to access this disk in write mode.

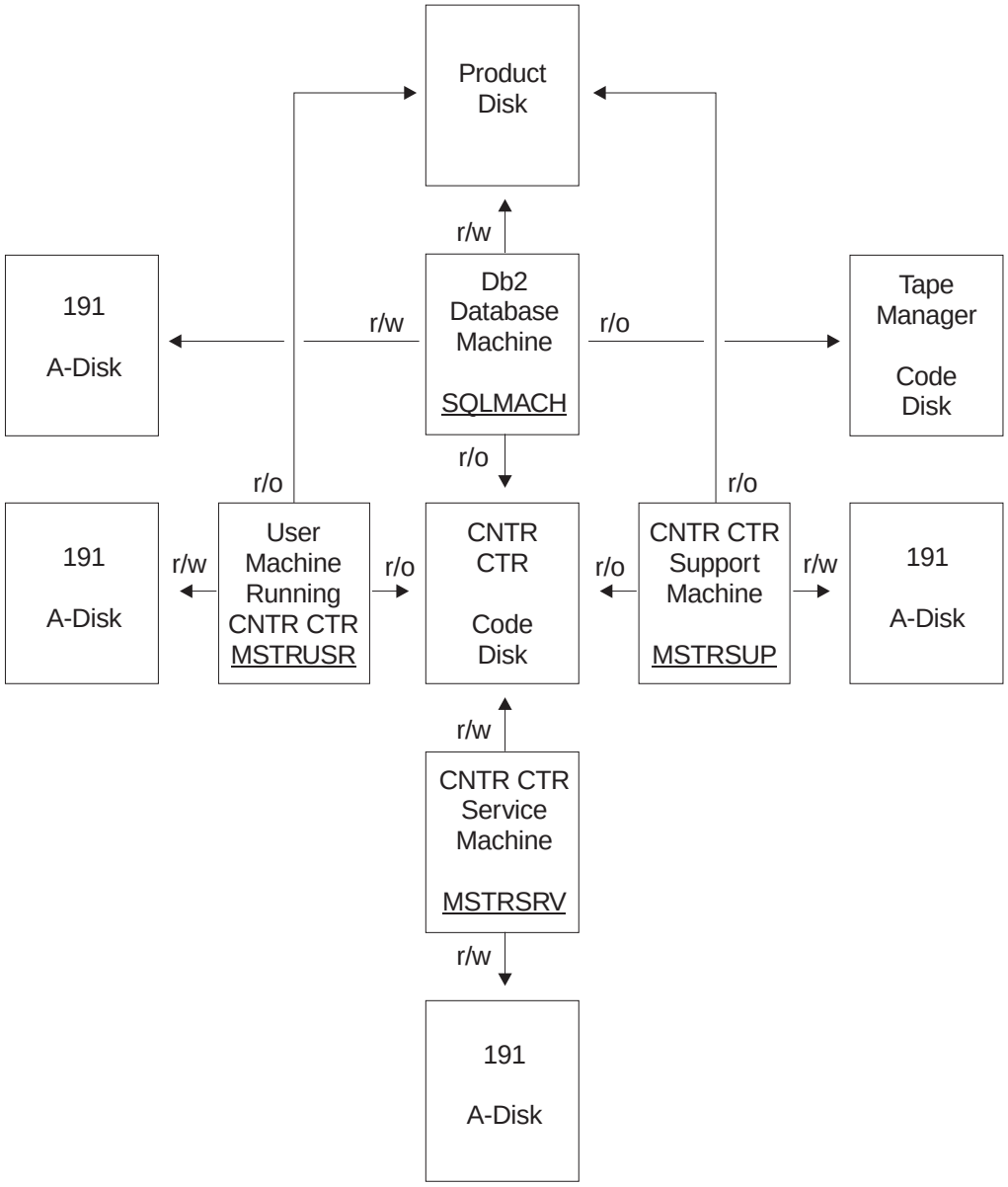


Figure 5. Control Center Code Disk

Tape Manager Code Disk

The Tape Management code disk, or any disk with code necessary for the management of tapes on your system, must be linked and accessed by your database virtual machines. Control Center will manage the linking and accessing of these disks automatically based on information supplied during installation. As shown in Figure 5, the database machine (SQLMACH) has read access to a tape management code disk.

The tape code disk virtual address and owner ID (user ID) are maintained in the SQLMSTR CONTROL file kept on the service machine's 191 A-disk. This file can be changed (taking care to maintain upper and lower case characters) and will take effect the next time Control Center is restarted.

Tape Management Interfaces

Control Center does not require a tape management product, but does support the following tape management software.

- DYNAM/T
- VMTAPE
- EPIC

In the absence of a tape manager, tape support is handled by sending tape mount request messages directly to a system operator ID. The product can also be adapted to work with other tape managers, with some customization.

Tape Management

Control Center allows the database archive and recovery functions to be automated with or without the availability of a tape management product. Physical mounting of tapes by a tape operator or automated hardware tape component (tape loader, tape hopper) is required. Ordinarily, a specific VM user ID is designated to receive tape mount requests from users on the system. The tape mount requests are usually sent as messages which include information concerning the tape to be mounted, what virtual machine requires the tape drive, what virtual address should be used for attaching the tape drive to the requesting user, whether the tape should be write-enabled, and other tape characteristics (labeled, scratch, density).

The SQMOUNT EXEC will be executed whenever a tape mount is required by any database. This exec will receive all parameters associated with the specific tape mount (which database, which virtual address, write-enabled). There are four different routines in SQMOUNT which are used to submit the appropriate tape mount request:

1. DYNAM/T
2. VMTAPE
3. EPIC
4. Generic CMS (sending a message to an operator ID)

One of the four routines will be executed, depending on information supplied during the installation process. If VMTAPE, or a routine that is not mentioned above is used, the SQMOUNT EXEC can need to be locally customized with appropriate density information. See “Post-Installation” on page 47.

Multivolume Tapes

Control Center is designed to handle multivolume tapes using the CMS FILEDEF and LABELDEF commands. When a database starts, Control Center will issue FILEDEF and LABELDEF commands for the ARIARCH ddname used by the database for archiving. When multiple labeled tapes are required for the archive, each one will be provided to CMS using the LABELDEF command.

Multi-volume tape handling is automatically processed, depending on whether the database is using the generic CMS handling of End-Of-Volume, or one of the supported EOV exits has been installed (DMSTVS or DMSTVI). When each database starts, Control Center will determine whether the database has access to

either the DMSTVS or DMSTVI module exit. When a subsequent End-Of-Volume is reached, Control Center will then perform the appropriate action based on which exit is available (if any).

The DMSTVI exit of CMS is supported. This allows tape management products to easily handle tape mounts and multivolume tape processing. DMSTVI is automatically invoked when the database performs an OS OPEN™ on a tape dataset during an archive. It is also invoked when End-Of-Volume (EOV) is reached on a tape within a multivolume tape set. Control Center will pass the specified parameter values to a tape management product through DMSTVI using the CMS FILEDEF and LABELDEF commands. The new LOGTAPE-PREMOUNT option also provides flexibility to support various implementations of the DMSTVI exit. Below are some highlights of tape processing. (DBSTART is the exec that Control Center uses to start up databases controlled by a service machine).

- Without DMSTVI module:
  - DBSTART issues ARIARCH LABELDEF specifying Volids
  - Control Center enters first ARCHIVE tape mount request
  - Control Center enters all log archive tape mount requests
  - CMS sends message to OPERATOR when EOV reached on tape
- With VMTTVS (VMTAPE):
  - DBSTART issues ARIARCH LABELDEF specifying Volids
  - DBSTART issues 'VMTTVS INSTALL' command
    - Loads nucleus extension, overlaying CMS EOV handling
  - Control Center enters first ARCHIVE tape mount request
  - Control Center enters all log archive tape mount requests
  - VMTAPE receives interrupt when EOV reached on tape
- With DMSTVI (VMTAPE):
  - DBSTART includes extra parms on FILEDEF and LABELDEF
    - SYSPARM on FILEDEF if Scratch Pool used
    - EXDTE parm on LABELDEF if RETENTION specified
    - FID parm on LABELDEF if DATASET specified
  - Tape mount sent to tape manager during OS OPEN and EOV
  - Control Center just watches process

Note: Please note that *CA-DYNAM/T* does **NOT** support multi-volume tape processing. Therefore, this version of Control Center will only support single tape handling.

Tape Mount Procedure

Control Center is designed to automate the database archiving and recovery processes. Ordinarily, a tape operator will mount a physical tape and attach the tape drive to the database machine. To reduce or eliminate manual interface, an automated tape handler or tape stacker can be used.

Control Center must rely on information (messages) within the VM/CMS operating system to provide full automation without intervention by the DBA during the archive and recovery processes. The primary area within the archive and recovery process where minimal system messages are available is during the tape mount function. Under VM/CMS, no automatic system message is provided to the requester when a tape is inserted into a tape drive and readied for use. This

makes it very difficult for Control Center to determine when the tape is available, so that the product can instruct the database to begin reading or writing to the tape.

Because of the limited number of VM/CMS tape system messages, Control Center has been designed to rely on the CP TAPE nnnn ATTACHED message which is generated when the system operator attaches the tape drive to the database virtual machine.

Note: This places a requirement on the tape operator to attach the tape drive to the database *only after* the tape has been inserted into the tape drive and readied for use. This will be required for *every* tape mounted, even multiple tape volumes defined for a single archive activity. Control Center will automatically detach the tape drive between each mount when the End-Of-Volume is reached.

Tape Management Products

Control Center currently supports:

1. DYNAM/T
2. VMTAPE
3. EPIC

For systems with VMTAPE, or other tape management products, the SQMOUNT EXEC can need to be modified to issue the appropriate MOUNT command syntax for that product. See “Post-Installation” on page 47.

Scratch Tape Pools

Scratch tape pools are supported when VMTAPE is used. This option allows separate tape pools to be defined for each database. When the scratch tape option is used, the tapes will be allocated from the scratch tape pool defined for the database.

Authorization

Users are authorized to access the product as either a Control Center Administrator, a Database Administrator, a Database Operator, or a Database User. The Control Center Administrator has the highest level of authority, which enables a user to perform any function available within a specific Control Center machine, including all DBA functions for all databases controlled by that machine. The Database Administrator can perform any Control Center function against one or more specified databases. The Database Operator can perform a subset of functions (such as archiving and database startup) for one or more specified databases. The Database User can only execute display functions for a specified database through the corresponding service machine, and perform the DATA only options of the Table Reorganization and Redefinition tool.

There is an additional level of authorization, that of database. This is for internal usage to allow communications between the database and the service machine.

The scheme below indicates the scope of authority for each level.

- **Five Authority Levels**

  5 = Control Center Administrator

4 = Database Machine
3 = Database Administrator
2 = Database Operator
1 = Database User

Authorization Level Definition

The five levels of authorization are discussed in more detail here.

1. **Administrators**

   The Control Center Administrator is responsible for the operational aspects of the service machine (there can be more than one administrator). An Administrator is allowed to perform any function that is available within Control Center, including all DBA functions for all databases under a service machine's control.

   Each Administrator user ID which is known by the service machine will receive many files and messages during Control Center operation. Many of these have to do with database activities and others with Control Center operation itself. Administrators are the main players in the setup of the Control Center system.

   At installation time one owner ID (discussed earlier in this section) will become the default administrator. Through the Control Center panel interface, this ID can also promote other users to be administrator. This user will also receive the SQLMSTR CONSOLE files that are closed and transferred from the service machine each day at midnight.

2. **Database and Support Machines**

   The database is assigned a level of authorization during the initialization process. This is needed to allow for communications between the database and the Service machine. This authorization is internal and transparent to the users of the product. It is noted here only for completeness' sake.

3. **Database Administrators**

   A Control Center Database Administrator is responsible for all operational aspects of a specific database. The DBA is allowed to perform all available operational functions within Control Center for that specific database. Every time a database is identified to the Service machine, specific information about the database must be given to it. One step in this identification process is to specify the user IDs which will be regarded as Control Center Database Administrators for that database. One user could be a Control Center DBA of many databases, but this must explicitly be told to the system in the identification step for every database which is new for a given Service machine. The Control Center Administrator will automatically acquire Control Center DBA authority over all databases controlled by a given Service machine. There can be more than one ID identified as a Control Center database administrator.

4. **Database Operators**

   Control Center Database Operators will typically have the authority to execute a subset of the commands available to the Database Administrator. For example, an operator may be able to initiate database archives and start the database, but may not be authorized to add dbextents.

5. **Database Users**

   Control Center Database Users have the lowest level of authorization. They are only allowed to issue SHOW commands against a database, but they are not

allowed to change anything in the database configuration or to perform hazardous actions against any database.

Notes:

1. The authorities within Control Center should not be confused with database authorities. These authorizations are for issuing valid Control Center commands only, and will not provide the users with any access within the database. Also, a user with database DBA authority will not automatically become a Control Center Administrator or Database Administrator. Assigning these authorization levels is part of the installation process. An Administrator or DBA must therefore be given database DBA authority to be able to successfully exploit all available Control Center functions.
2. The authorization capabilities assigned to each level can be customized by modifying the SQLMSTR PROFILE file where the various functions Control Center performs are associated with a level. See Appendix E, “Authorizations” on page 511 for a list of these functions/levels, and see Chapter 11, “Managing the Environment” on page 113, for a further discussion of this capability.

Chapter 3. Installation and Migration Overview

The next few chapters are intended for the person installing Control Center and describe:

- How to install the Control Center product.
  - Installing the Control Center Service Machine (Chapter 4)
  - Installing a Control Center Support Machine (Chapter 5)
  - How to set up a database machine to work with Control Center (Chapter 6)
- How to migrate from Control Center 5.1 or earlier releases of SQL Master. (Chapter 7)
- How to install Control Center service. (Chapter 8)

  Checklists included in each of the above Chapters, with the exception of Chapter 8, “Corrective Service” on page 91, allow you to mark off required steps as you complete them.

Information presented in this section assumes that you have read Chapter 1, “Introduction” on page 1, and Chapter 2, “Architecture” on page 13 of this manual, and are familiar with CMS commands, EXECs, and Virtual Machine (VM) operation. Additionally, you should read Chapter 3, “Installation and Migration Overview” prior to performing any of the installation processes.

Installation

The installation of the Control Center product is separated into three distinct procedures.

1. **Installation of the Control Center Service Machine** (required)

   Either,

   - *Full Installation*, which includes defining a Control Center service machine, defining and formatting a Control Center code disk, code installation from the distribution tape to the code disk, and configuration of the Control Center control files. In this configuration the Control Center service machine owns the code disk.

     OR
   - *Shared Installation*, which includes defining a Control Center service machine with a read-only link to a previously installed Control Center code disk, and configuration of the Control Center control files.
2. Refer to Chapter 4, beginning on page 37.
3. **Installation of a Control Center Support Machine** (optional, although one or more support machines is a good idea)

   Includes defining a Control Center support machine that links in READ mode to the same code disk as the managing service machine, configuring the Control Center control files, and establishing proper machine authorities.

   Refer to Chapter 5, beginning on page 51.
4. **Setting Up a DB2 Server for VM Database with Control Center** (required)

Includes redefining an existing database machine to link in READ mode to the same code disk as the managing service machine, configuring the Control Center database control files, and establishing proper machine authorities.

Refer to Chapter 6, beginning on page 61.

New installations *must*, at a minimum, complete the following:

a. Install a Control Center service machine (full installation), and

b. Set up a database to work with Control Center.

Once a Control Center service machine has been successfully installed you can at any time:

- Repeat the steps described in Chapter 6 to set up additional databases to work with the installed service machine.
- Perform the steps described in Chapter 5 to add a Control Center support machine to your installation configuration. Multiple Control Center support machines can be installed.

Migration

Migration is the process used to move from one level of Control Center code to another level.

If you are migrating from Control Center V5.1 or SQL Master (either Version 1 Release 2.0 or Version 1 Release 1.2), refer to Chapter 7 beginning on page 83. For migration from prior levels of SQL Master (SQL Master/VM Version 1 Release 1 Modification 0 or Modification 1), you must first migrate to SQL Master Version 1.2. To do this, execute the migration exec MIGRV1R2. You will be prompted through this migration. Once you have migrated to SQL Master Version 1.2, follow the recommended steps in the above referenced migration chapter to complete the migration to Control Center V6.1.

Optional Steps

Depending on the needs of your installation, there are several ways of configuring the Control Center environment. The following is a brief description of some optional steps which can be taken during the installation process.

Multiple Databases

A single Control Center service machine can manage several database servers. The only restriction is that each database machine must be on the same CPU as the Control Center service machine.

The steps for setting up a database machine to work with Control Center are discussed in Chapter 6, beginning on page 61.

Control Center Support Machine

Database Administration tools are database applications which require a dedicated virtual machine during processing. These tools should *not* be scheduled (using the Control Center Job Scheduling tool) to execute on a Control Center service machine, since all automated operation functions with all databases managed by the service machine will be disabled during tool execution.

If you plan on scheduling Database Administration tools, you should install one or more Control Center support machines where these tools can execute, allowing the Control Center service machine to continue performing other tasks. Installing multiple support machines allows you to run tools against different databases concurrently. Job scheduling is managed by the controlling Control Center service machine.

The steps for installing a Control Center support machine are discussed in Chapter 5, beginning on page 51.

Corrective Service Planning

This program release will be maintained through the use of PTF tapes. An updated Version/Release replaces the entire program code. A PTF tape can only replace the changed portion of the program code.

For more information, refer to Chapter 8, “Corrective Service” on page 91.

Processor Requirements

- Control Center is designed to operate on IBM System/370 and IBM System/390 architecture processors which support the IBM Virtual Machine operating system.
- Installation and maintenance require a magnetic tape unit and a terminal supported by VM.
- Approximately 30 cylinders of IBM 3380-equivalent direct access storage is required to store the Control Center code and control files.

Prerequisite Programs

The Control Center program requires an environment provided by:

- IBM Virtual Machine/Enterprise Systems Architecture (VM/ESA), Version 2 Release 2 (5684-030)
- DB2 Server for VM (5648-A70) Version 6 Release 1
- DB2 Server for VM (5648-158) Version 5 Release 1 (Control Center will operate with SQL/DS (5688-103) Version 3 Release 5.)

Control Center does not require a tape management product, but it provides support for the DYNAM/T , VMTAPE, and EPIC, products if they are used on the target system. Control Center also supports environments without any tape management products, sending tape mount request messages directly to a defined tape operator user ID. Control Center can also be adapted to other tape management products with minor customization.

Control Center Virtual Machine Requirements

To install the Control Center product, you need these virtual machines:

1. **The VM MAINT machine.** Used for installation of Control Center from the distribution tape. This machine or its equivalent already exists in all VM systems.
2. **A database machine.** One or more database machines (application servers) is required. Machine identifier (user ID) SQLMACH is used in this manual to refer to a database machine. You should use the identifier of a database machine in your environment.
3. **A Control Center service machine.** A virtual machine operated in disconnect mode that runs the Control Center code. The service machine can be defined as owning a Control Center code disk or sharing (links and accesses) an existing code disk on the same CPU. In this section, machine ID MSTRSRVO refers to a service machine that owns a code disk, MSTRSRVS refers to a service machine that shares a code disk, and MSTRSRV is a generic reference to both types of service machines. You can define a Control Center service machine with any valid machine identifier (user ID). A single service machine can manage one or more database servers on the same CPU. There can be more than one service machine per CPU.
4. **A Control Center support machine** (optional). A virtual machine operated in disconnect mode and managed by a Control Center service machine. It is referred to in this manual as the MSTRSUP machine, but you can define it with any valid machine identifier. Support machines are dedicated to running scheduled Database Administration tools (database applications) allowing the managing service machine and user machine to continue performing other tasks.

   Multiple Control Center support machines can be installed on a single CPU and managed by a single Control Center service machine.

   A Control Center support machine links and accesses the same code disk as its managing service machine.
5. **A Control Center owner machine.** A virtual machine designated as the owner of a Control Center service machine or support machine. (*Own* in this context describes a logical association between two virtual machines.) It is referred to in this manual as the MSTROWN machine, but you can define it with any valid machine identifier.

   Notes:

   a. The requirements for an owner machine are the same regardless of whether it is designated as the owner of a service machine or support machine. Therefore, a single machine identifier (MSTROWN) is used when referring to an owner machine.
   b. A single MSTROWN machine can be the owner of any number and combination of service machines and support machines.
   c. An owner machine can also be a Control Center user machine.
   d. References to the owner of the service machine also apply to the owner of the support machine unless specifically stated otherwise.
6. **A Control Center user machine.** Any virtual machine that has been defined to access the Control Center product. There can be many user machines.

To complete Control Center for VM installation, these machines must meet certain virtual storage and minidisk requirements. These requirements are discussed in the following sections.

VM MAINT Machine

The MAINT machine is referred to throughout this section, but you can use another machine with equivalent authority and disk access.

Virtual Storage Requirements

The Control Center product does not have any special virtual storage requirements for the MAINT machine.

Minidisk Requirements

The Control Center product does not have any special minidisk storage requirements for the MAINT machine. Control Center code is loaded from tape directly to the Control Center code disk.

Database Machine (SQLMACH)

The VM directory entries for a database machine will require updating to identify the managing Control Center service machine as the secondary (alternate) console machine on the CONSOLE statement. You will be instructed to redefine the SQLMACH machine during database setup.

Virtual Storage Requirements

The Control Center product does not have any special virtual storage requirements for the SQLMACH machine.

Minidisk Requirements

Several small files will be created on the database machine's 191 minidisk (A-disk) during normal operations under Control Center. The 191 minidisk may therefore need to be increased slightly from the requirements specified within the *DB2 Server for VM System Administration* manual.

Control Center Service Machine (MSTRSRV)

A Control Center service machine must be defined to install and run the Control Center product. It will contain a collection of data structures and operating software that allows it to manage and communicate with database servers and Control Center support machines. MSTRSRV operates in a continuous disconnected server mode, reacting to external interrupts in the form of VM messages and reader files, and acting on scheduled interrupts to execute planned activities. You will be instructed to define the MSTRSRV machine during installation.

Virtual Storage Requirements

The virtual storage required by the Control Center service machine varies depending on the number of databases being managed. The minimum machine size for a single database is 6-megabytes, and 8-megabytes for a multiple database environment.

Minidisk Requirements

Two minidisks are required for the MSTRSRV virtual machine:

- Control Center system minidisk.
- Control Center code minidisk.

Control Center System Minidisk: An A-disk (191 minidisk) which will contain control files, parameter files, output files, all database-related files, and other files customized for the local environment.

The size of the 191 minidisk will vary depending on the number of databases, the number of backup copies of modified files that MSTRSRV is instructed to keep, and the number and frequency of Database Monitoring tools that are scheduled. A typical installation will require approximately 10 cylinders of IBM 3380-equivalent DASD, which can later be adjusted as required for your environment.

Control Center Code Minidisk: The MSTRSRV virtual machine requires read access to a Control Center code disk. The code disk can either be owned by the service machine being installed or by another service machine on the same CPU. That is, a single Control Center code disk can be shared among multiple Control Center service machines (and support machines) on the same CPU.

In addition to Control Center code modules, files SQLMSTR DIRECTRY and DBINIT CONTROL are kept on the code disk and are maintained by the service machine owning the disk. These files define key characteristics of the Control Center and database environment, including what service machines manage what databases, and are required for proper Control Center operation. All machines which link to the code disk use these files and therefore, will be a part of the same environment.

A Control Center code disk (195 minidisk) will require approximately 20 cylinders of IBM 3380-equivalent DASD to contain all Control Center code.[2]

Control Center Support Machine (MSTRSUP)

A Control Center support machine is dedicated to running Database Administration tools (database applications) scheduled by a managing service machine. It is defined similar to a service machine except it always shares (links and accesses) the existing Control Center code disk used by the managing service machine. You will be instructed to define the MSTRSUP machine during installation.

Virtual Storage Requirements

The virtual storage required by the Control Center support machine is somewhat larger than that of the service machine. Depending on the type of activity and size of dbspace and table reorganizations, a 6 to 8-megabyte machine size should be used.

[2] You can use any valid virtual device number (cuu) for linking the code disk.

Minidisk Requirements

The size of the A-disk (191 minidisk) for the support machine will vary greatly depending on the type and number of activities scheduled to execute on the machine and the options chosen when tools are scheduled. See “Installation Step step 2: Define a Control Center Support Machine (MSTRSUP)” on page 53 for a further discussion of these requirements.

Control Center Owner Machine (MSTROWN)

At least one Control Center owner machine is required in the Control Center environment. The owner machine automatically has Control Center administrator authority and will receive CONSOLE files that are closed and transferred from the service machine each day at midnight. The owner of a Control Center service machine is identified in file SQLMSTR PROFILE on the service machine's 191 A-disk. The owner machine can also be a user machine. No changes to the VM directory are necessary for the MSTROWN machine.

Virtual Storage Requirements

The owner machine should have a minimum of 4 megabytes.

Minidisk Requirements

The owner machine must have an A-disk (191 minidisk) with approximately three cylinders of IBM 3380-equivalent DASD.

Control Center User Machine (MSTRUSR)

Any virtual machine that has been defined to access the Control Center product is referred to as an MSTRUSR machine. User machines must link and access the Control Center code disk to use the product. No changes to the VM directory are necessary for MSTRUSR machines.

Virtual Storage Requirements

The user machine should have a minimum of 4 megabytes. Depending on the type of activity and size of dbspace and table reorganizations, a 6-megabyte machine may be needed.

Minidisk Requirements

The user machine must have an A-disk (191 minidisk) with approximately three cylinders of IBM 3380-equivalent DASD.

Chapter 4. Installing the Service Machine

This chapter describes how to install the Control Center service machine.

- *Full Installation* includes defining a Control Center service machine, defining and formatting a Control Center code disk, code installation from the distribution tape to the code disk, and configuration of the Control Center control files.
- *Shared Installation* includes defining a Control Center service machine with a read-only link to a previously installed Control Center code disk, and configuration of the Control Center control files.

Checklist for Installation of the Control Center Service Machine

Notes:

- *Perform the steps in order.*
- *Optional steps are preceded by a circle (○).*
- *Page references appear in parentheses.*

Preliminary Installation

__ 1. Log On to the VM MAINT Machine (38)

__ 2. Define a Control Center Service Machine (38)

   a. [Full Installation Only] Define MSTRSRVO (38)

   OR

   b. [Shared Installation Only] Define MSTRSRVS (40)

__ 3. ○ Define a Control Center Owner Machine (MSTROWN) (41)

__ 4. Update the VM Directory (41)

__ 5. Link and Format the 191 Disk(s) (41)

__ 6. [Full Installation Only] Link and Format the Code Disk (42)

__ 7. [Full Installation Only] Make Code Disk Available (42)

__ 8. Add Service Machine to Autolog List (42)

Code Installation

__ 9. [Full Installation Only] Install Control Center Code (42)

__ 10. Log Off the MAINT Machine (43)

__ 11. Log On to the Service Machine (43)

__ 12. Access Code Disk (43)

__ 13. Configure Control Center Control Files (43)

__ 14. Start Control Center Service Machine (46)

Installation Verification

__ 15. ○ Log On to the MSTROWN Machine (47)

__ 16. ○ Verify Files/Messages Received (47)

__ 17. ○ Link and Access the Control Center Code Disk (47)

__ 18. ○ Verify Control Center Service Machine Operation (47)

Post-Installation

__ 19. ○ Tape Mount Interface Customization (47)

__ 20. Perform Database Setup (48)

__ 21. Perform Data Restore Setup (49)

Preliminary Installation

The first eight steps of the installation process prepare your VM system for the Control Center product by defining all required virtual machines and their minidisks, and making the Control Center code disk available to users of the product.
Figure 6 summarizes what you will accomplish when performing a *full installation*.
Figure 7 on page 39 summarizes what you will accomplish when performing a *shared installation*.

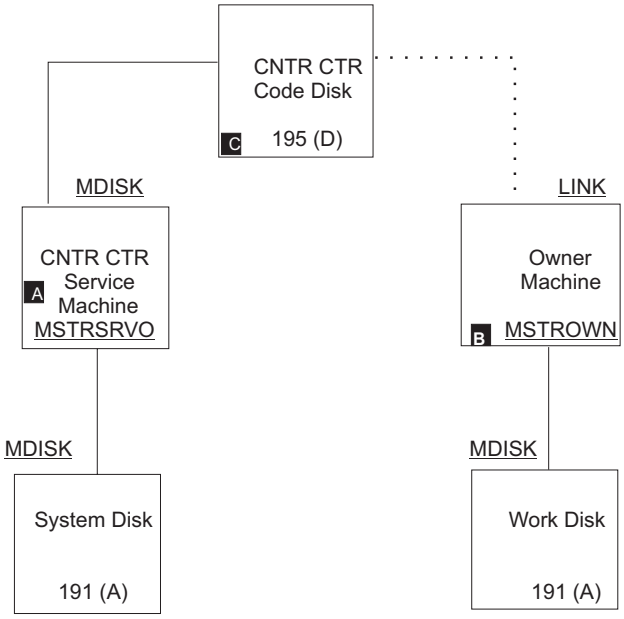


Figure 6. Virtual Machines and Minidisks in a Control Center Service Machine Full Installation. MSTRSRVO **A** is the owner of the Control Center code disk **C**. MSTROWN **B** links to the code disk in READ mode and is defined as the logical owner of MSTRSRVO.

Installation Step 1: Log On to the VM MAINT Machine

To begin the Control Center installation, log on to the VM MAINT virtual machine.

Installation Step 2: Define a Control Center Service Machine

To use the Control Center product, define a service machine with the appropriate VM directory entries.

Installation Step 2a: [Full Installation Only] Define MSTRSRVO

Update the VM directory statements as shown in Figure 8 on page 39 to define a Control Center service machine that owns a code disk.

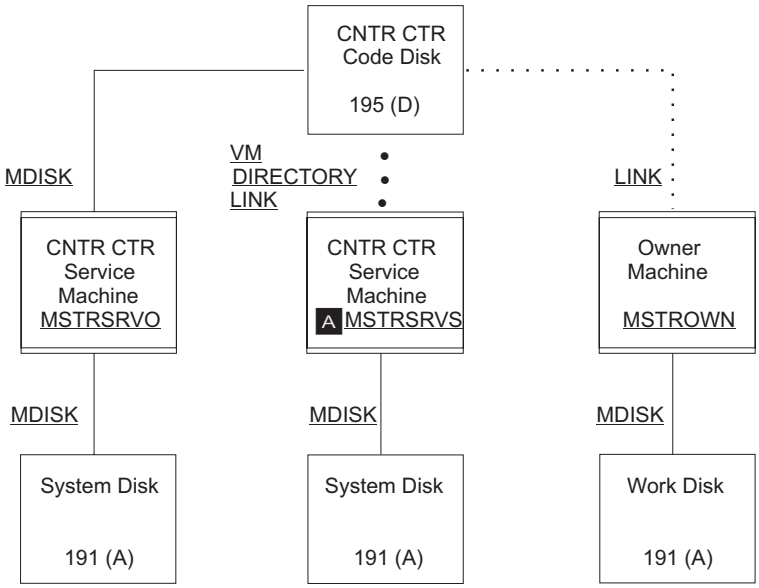


Figure 7. Virtual Machines and Minidisks in a Control Center Service Machine Shared Installation. MSTRSRVS **A** has been added to the environment shown in Figure 6 using the *shared installation* process. A single MSTROWN machine (installed during the *full installation* process) is defined as the logical owner of MSTRSRVS and MSTRSRVO. You may, however, define a separate owner machine for each service machine.

```
1→  USER MSTRSRVO XXXXXXXX 6M 6M BG
    ACCOUNT nnnnnnnn
2→  IUCV ANY MSGLIMIT 255
3→  IPL CMS
4→  CONSOLE     009    3215
    SPOOL       00C    2540          READER      *
    SPOOL       00D    2540          PUNCH       A
    SPOOL       00E    1403          A
    LINK  MAINT 190        190       RR
    LINK  MAINT 19D        19D       RR
    LINK  MAINT 19E        19E       RR
5 → MDISK 191   3380   cylr 015 volser MR
6 → MDISK 195   3380   cylr 020 volser MR
```

Figure 8. Sample VM Directory Entries for a Service Machine With a Code Disk

Statement 1: USER MSTRSRVO XXXXXXXX 6M 6M BG

Defines the MSTRSRVO machine as a 6-megabyte virtual machine with the VM privilege classes B and G. Privilege class B is optional and allows the service machine to send messages with the MSGNOH command for improved clarity.

Statement 2: IUCV ANY MSGLIMIT 255

This statement allows the service machine to communicate with any database machine through the Inter-User Communications Vehicle (IUCV) feature of VM. The **MSGLIMIT 255** option on this statement provides the service machine with a message queue capacity of 255 messages. This option is required to prevent possible loss of database messages, especially when multiple database machines are controlled by a single service machine.

Statement 3: IPL CMS

This statement should use the CMS saved segment name which is applicable for your VM environment.

Statement 4: CONSOLE 009 3215

This statement *must not* include the **T userid** option (which specifies a secondary console for the service machine).

Note:

All Control Center functions will be disabled if the CONSOLE statement includes the **T userid** option.

Statement 5: MDISK 191 3380 cylr 015 volser MR

MDISK entry for read/write Control Center system disk (A-disk) with device address 191. This is required by the service machine to contain control files and other locally modified files for Control Center operation. Between 10 and 20 cylinders of IBM 3380-equivalent DASD should be sufficient to hold all software and control files for up to six databases. This may differ by installation depending on the number of active users, the number of back copies of control files retained, and the amount of activity performed during a typical day's work (especially Database Monitoring tools).

Statement 6: MDISK 195 3380 cylr 020 volser MR

MDISK entry for read/write Control Center code disk with device address 195 (can be any address other than 191). This is required by the service machine to contain all executable Control Center code. This disk may be shared by multiple Control Center service/support machines on the same system. Approximately 20 cylinders of IBM 3380-equivalent DASD should be sufficient to hold Control Center Version 6 Release 1.0 level code.

All other VM directory entries in this example should correspond with those provided for any virtual machine on the target system.

Installation Step 2b: [Shared Installation Only] Define MSTRSRVS

To define a Control Center service machine that shares an existing Control Center code disk, update the VM directory statements as shown in Figure 9 on page 41.

Only the statements which differ from those of the full installation are explained below. For details on the other statements, refer to “Installation Step 2a: [Full Installation Only] Define MSTRSRVO” on page 38.

```
1 ──►USER MSTRSRVS XXXXXXXX 6M 6M BG
     ACCOUNT nnnnnnnn
     IUCV ANY MSGLIMIT 255
     IPL CMS
     CONSOLE      009    3215
     SPOOL        00C    2540                    READER      *
     SPOOL        00D    2540                    PUNCH       A
     SPOOL        00E    1403                    A
     LINK   MAINT      190          190 RR
     LINK   MAINT      19D          19D  RR
     LINK   MAINT      19E          19E  RR
     MDISK  191   3380   cylr  015 volser         MR
6──► LINK   MSTRSRVO   195          195  RR
```

Figure 9. Sample VM Directory Entries For a MSTRSRV Machine that Shares a Code Disk

Statement 1: USER MSTRSRVS XXXXXXXX 6M 6M BG

Defines the MSTRSRVS machine as a 6-megabyte virtual machine with the VM privilege classes B and G. Privilege class B is optional and allows the service machine to send messages with the MSGNOH command for IND$FILE PUT ARIC141 EPSBIN A (RECFM V

Statement 2: LINK MSTRSRVO 195 195 RR

This statement links in READ mode to an existing Control Center code disk (MSTRSRVO) with virtual device address 195 (can be any address other than 191).

Note: MSTRSRVS must be authorized to access the code disk in READ mode.

Installation Step 3: Define a Control Center Owner Machine (MSTROWN)

A MSTROWN machine should be defined at this time if such a machine does not already exist. MSTROWN will be designated later in the installation process as the owner of the service machine. (*Own* in this context describes a logical association between two virtual machines.) No special VM directory changes are required for this machine.

A MSTROWN machine must include an A-disk (191 minidisk) with approximately three cylinders of IBM 3380-equivalent DASD.

Installation Step 4: Update the VM Directory

After you have made the VM directory control statement changes for the Control Center service machine and the Control Center owner machine, update the VM directory using your local operating procedures.

Installation Step 5: Link and Format the 191 Disk(s)

Link to the service machine's 191 disk and format as the A-disk.

If a new MSTROWN machine was defined above, then you should also format the MSTROWN 191 disk as the A-disk.

Installation Step 6: [Full Installation Only] Link and Format the Code Disk

Important:

This step is applicable only when performing a *full installation*.

Link to the service machine's 195 code disk and format as the D-disk.

Installation Step 7: [Full Installation Only] Make Code Disk Available

Each database machine, shared service machine, support machine, user machine, and the owner machine will require read access to the Control Center code disk.

Follow your local authorization procedures to grant read access to the Control Center code disk. If your environment uses a security access product, then follow the directions of the product for ensuring access to the code disk.

Note: Universal read access to the code disk should be defined.

Installation Step 8: Add Service Machine to Autolog List

Under normal operations the service machine should be autologged each time the CPU is IPLed. The service machine should therefore be included in the list of virtual machines that are autologged each time the CPU is IPLed.

Important:

1. To correctly establish the secondary console interface (SCIF) between the service machine and each database, the service machine *must* be autologged *first* (in disconnect mode) before each database machine is logged on. If a database machine is logged on before the service machine, the SCIF interface will not be established between the two and the service machine will not be able to perform its normal function with that database.
2. If your VM system has a facility installed which logs off inactive machines, place the service machine in the list of machines to be excluded. This list is generally referred to as the log-off exclusion list.

Code Installation

The VM system is now prepared for the Control Center product. Steps 9 through 14 install the product.

Installation Step 9: [Full Installation Only] Install Control Center Code

If doing a full installation, then the Control Center code must be installed on the MSTRSRVO 195 code disk from the product distribution tape. This is normally accomplished from the MAINT user ID by linking to the MSTRSRVO 195 code disk in WRITE mode, mounting the product tape, and using VMFPLC2 to copy the code from tape to disk, as described in this procedure:

1. Link and access the MSTRSRVO 195 code disk in WRITE mode, for example as filemode K.

2. Mount the Control Center distribution tape on CUU 181 without a ring. The actual mount command will vary from site to site depending on the tape handling software installed.
3. Rewind the tape.

   VMFPLC2 REW
4. Load the files onto the accessed MSTRSRVO 195 code disk.

   VMFPLC2 LOAD * * K

   Note: The file SQLMSTR VERSION, which is included in the LOAD, will contain the version number of the Control Center software.
5. Detach the tape.

Installation Step 10: Log Off the MAINT Machine

Release and detach the service machine's 191 disk and, if doing a full installation, the code disk. Log off the MAINT Machine.

Installation Step 11: Log On to the Service Machine

Log on to the service machine to complete the installation and configuration process.

Installation Step 12: Access Code Disk

Access the Control Center 195 code disk with an available disk mode.

Installation Step 13: Configure Control Center Control Files

Use the C5648A70 EXEC supplied with Control Center to configure the Control Center software for your installation by entering,

```
C5648A70
```

at the CMS command prompt.

Note: The C5648A70 EXEC takes approximately 10 minutes to complete.

The C5648A70 EXEC will first verify that all Control Center code has been successfully installed on the code disk. It then prompts you for information about the type of installation being performed and the Control Center code disk being used:

1. The type of Control Center machine you are installing; enter `SERVICE`.
2. Whether this is a full or shared installation.
3. [Shared Installation Only] The virtual machine ID of the code disk owner.
4. The virtual device address (CUU) to be used by the service machine when linking the code disk.
   - [Full Installation Only] You must use the same CUU specified on the MDISK directory statement in Step 2a.
   - [Shared Installation Only] You must use the same CUU specified on the LINK directory statement in Step 2b.
5. The disk access mode to be used by the service machine when accessing the code disk.

Next, C5648A70 EXEC generates these Control Center control files, prompting you for information as required:

- SQLMSTR CONTROL
- SQMOUNT EXEC
- SQLMSTR PROFILE
- PROFILE EXEC
- SQLMSTR TIMES

When file SQLMSTR CONTROL is generated, you are prompted for the information described below. You should review your needs ahead of time so you can respond properly for your installation. Once Control Center is operational, you can update file SQLMSTR CONTROL through the Control Center panel interface.

1. The number of Control Center service machine log files to retain.

   Enter the number of days you wish to keep a log file before it is automatically deleted. If no entry is made, the default (5) will be used.

   Control Center service machine log files contain a history of all interrupts within the service machine and are used primarily for debugging purposes. They are generated daily (named SQLMSTR LOG*yyddd*, where *yy* is the year and *ddd* is the Julian day) and are kept on the service machine's 191 disk before being automatically deleted after the specified number of days.

2. The number of backup copies of modified files to retain.

   Enter the number of past copies of modified files that you wish Control Center to keep. If no entry is made, the default (3) will be used.

   When a file is updated on the service machine's 191 A-disk or, if it is the owner, the code disk, Control Center creates a backup copy of the original file with a date-coded filetype and places it on the same disk. Previous backup copies will be maintained by Control Center up to the number specified by this value.

3. The Spool Console option is set to Y (yes).

   The spool Console option is set to Y. This means that Control Center will spool its console upon startup. This will allow a complete console to be available if needed. This option can be modified to N if spooling is not desired by using the SQM CONSOLE OFF command from the CMS command line.

4. Whether your installation uses a tape management product.

   Specify the tape management product, if any, that your installation uses to support DB2 Server for VM.

   Enter `DYNAMT` if the CA-DYNAM/T product is used for all tape mount requests, `VMTAPE` if the VMTAPE product is used, `EPIC` if the EPIC product is used, or `NONE` if none of these products are used.

5. Whether your installation uses `DYNOPEN` as the method for handling tape mounts if `DYNAMT` is used.

6. The name of the virtual machine that should receive mount requests.

   Specify the recipient of tape mount requests.

   Control Center issues tape mount requests by sending messages to an appropriate virtual machine (user ID). The name of the tape manager virtual machine (or operator) should be entered here, whether a tape manager product

is used or not. If no entry is specified, then the previous response is used, unless it was NONE, in which case the value OPERATOR is used.

If DYNAM/T , VMTAPE, or EPIC is being used, you are prompted for information about the tape manager code disk:

a. The virtual machine ID of the tape manager code disk owner.
b. The virtual device address of the tape manager code disk.
c. The filemode the database should use to access the code disk.

These entries are only required if you want each database to automatically link and access the tape manager code disk during startup. If you provide the link and access yourself, these entries can be left blank by simply pressing only the ENTER key.

7. The Control Center disk usage warning threshold.

   Enter an integer value between 50 and 98 for the disk full percentage which, when reached, causes a warning message to be sent to Control Center Administrators. If no entry is specified, the default value (70) is used.

   Each day at midnight, Control Center determines whether the Control Center Administrators should be notified about the amount of data on the service machine's 191 A-disk. If the service machine's A-disk has reached or exceeded the percent full indicated by this value, then a message file will be sent to each Administrator notifying them of this condition.

8. The Control Center disk full warning threshold.

   Enter an integer greater than that of the previous response (and less than or equal to 99) for the disk full percentage which, when reached, causes a serious warning message to be sent to Control Center Administrators. If no entry is specified, then a value one greater than your previous response is used.

9. CMS Disk Access Options.

   Enter any valid CMS ACCESS command options to be used when accessing linked disks.

   CMS disks will be linked and accessed by the database machine under Control Center control. Control Center will use any valid options for the CMS ACCESS command that are specified here. You can use the CMS HELP ACCESS command to review valid ACCESS command options available.

10. Date/Time Format

    Control Center offers four different date/time formats to be used in all menu displays and reports generated. They are:

    - USA for mm/dd/yy and hh:mm:ss (this is the default format)
    - EUR for dd.mm.yyy and hh:mm:ss
    - ISO for yyyy-mm-dd and hh:mm:ss, and
    - JIS (same as ISO)

    Enter any of the above or blank to default to USA.

You should now receive several messages indicating that files SQLMSTR CONTROL and SQMOUNT EXEC have been generated.

When file SQLMSTR PROFILE is generated, you are prompted for the information described below. You should review your needs ahead of time so you can respond properly for your installation.

1. The logical owner of the service machine.

   Enter the machine identifier (user ID) that should be recognized as the owner of the Control Center service machine. The owner machine is referred to in this manual as MSTROWN, but it can be any existing virtual machine. If no entry is specified, the current log-on ID is used.

   This machine will automatically have Control Center Administrator authority and will additionally receive the SQLMSTR CONSOLE file that is closed and transferred from the service machine each day at midnight.

2. One or more users who will be Control Center Administrators.

   Respond in the form `userid` or `userid at nodeid`. This prompt will be displayed repeatedly until you enter a blank response (or `Q` to terminate the install process early).

   Each user ID entered will be given Control Center Administrator authority to the Control Center service machine, allowing them complete control and authority to perform any Control Center function for any database.

   Note: Control Center Administrators may be added and deleted through the Control Center panel interface once Control Center is operational.

You should now receive several messages indicating that files SQLMSTR PROFILE, PROFILE EXEC and SQLMSTR TIMES have been generated, and that Control Center parameters have been loaded into memory.

This completes C5648A70 EXEC processing.

Installation Step 14: Start Control Center Service Machine

This step will initialize and start the Control Center service machine.

Enter these commands at the CMS command prompt:

```
#CP IPL CMS
SQLMSTR
```

The service machine should immediately be disconnected from your active terminal session and the startup process will be completed in disconnect mode. You will be logged off the terminal at this point. The installation process is complete.

Installation Verification

The following steps will determine whether Control Center is functioning and can communicate with users (but not yet with databases, which is verified after a database setup under Control Center has been done).

Installation Step 15: Log On to the MSTROWN Machine

Log on to the MSTROWN machine or any machine authorized previously as a Control Center Administrator on the service machine.

Installation Step 16: Verify Files/Messages Received

If logged on to the MSTROWN machine, you should have received file SQLMSTR CONSOLE (sent by the service machine) in the virtual reader.

If, during Control Center startup processing, you are logged on to a machine with Control Center Administrator authority (including MSTROWN), you should receive a Control Center startup message from the service machine.

Note: You *must* be logged on at the time Control Center startup processing on the service machine issues the message for it to appear on your console. Continue with the verification steps even if you do not receive a message.

Installation Step 17: Link and Access the Control Center Code Disk

Link and access the Control Center code disk in READ mode with an available virtual device address (CUU) and access mode (195 D).

Installation Step 18: Verify Control Center Service Machine Operation

To establish communications with the service machine, enter this at the CMS command prompt:

```
SQM (MSTRSRV
```

Replace `MSTRSRV` with the machine ID of your service machine.

Verify the service machine is operating by entering,

```
SQM VERSION
```

Control Center will return a message identifying the service machine (MSTRSRV) and the Control Center product release and modification level. If this is successful, Control Center has been successfully installed and the verification process is complete.

For more information about SQM VERSION, refer to Chapter 10, “Version Tool” on page 107.

Post-Installation

Installation Step 19: Tape Mount Interface Customization

SQMOUNT

The Control Center tape mount interface exec (SQMOUNT EXEC) provides four different mount commands that will be issued depending on responses given during the installation process. The SQMOUNT EXEC is used when a mount request must be placed on the database or Data Restore machine's console by the Control Center service machine. The mount commands provided for EPIC should not need modification. The generic mount command provided for environments with DYNAM/T and VMTAPE, or other tape management products not mentioned

above, may require customization for local standards based upon the tape drive used.

These parameters are passed to the SQMOUNT EXEC by Control Center when a tape mount is required:

dbname The database machine that needs the mounted tape.

volid The volume identifier of the tape to be mounted.

cuu The channel and unit address (virtual device address) where the tape should be attached (typically 181, 182, 183).

density The tape density that should be used. If VMTAPE is used with a 3490C tape drive, specify a tape density of **DEN E**.

rw Either READ or WRITE to indicate whether the tape should be mounted in READ mode or WRITE mode.

dsn The dataset name specified for archive, log archive, or tracing specified in the database parameters file. The database parameters file is discussed in Chapter 6, “Database Setup” on page 61.

Any modifications made to the SQMOUNT EXEC should utilize the above parameters to execute a tape mount request with the proper characteristics.

An example of the tape mount routines provided with Control Center can be found in file SQMOUNT $EXEC on the Control Center code disk. All local changes should be made to SQMOUNT EXEC on the service machine's A-disk.

Note: All tape mount requests are issued using CP SEND CP dbname. The CP SEND CP command sends the command to the database virtual machine using the SCIF facility. The command will be executed on the database machine using the #CP command format, which allows a CP command to be issued while the database is operational. Only #CP format commands are executable on the database while the database is running, so this command construct must be kept for any modifications made to the SQMOUNT EXEC. Only valid CP commands are allowed. The supplied example utilizes the CP MSG command to issue all tape mount requests from the database machine.

SQMSTAPE

The SQMSTAPE EXEC is provided to issue mount requests during table or dbspace reorganizations, and during some Data Restore functions when the "DRTAPE_PREMOUNT" is "Y" in the database PARMS file. This exec is only used when a Data Restore machine, Support machine or user machine must issue a tape mount for its own userid. The SQMSTAPE EXEC is located on the Control Center code disk. SQMSTAPE EXEC should be modified if an unsupported tape management system is used or if local modifications are necessary. The SQMSTAPE $EXEC is provided as an example file and should not be modified.

Installation Step 20: Perform Database Setup

Perform the database setup procedure described in Chapter 6, “Database Setup” on page 61 for each database to be managed by this service machine.

Installation Step 21: Perform Data Restore Setup

If Data Restore has been installed, do the steps listed in “Data Restore Machine Setup” on page 104 to enable Control Center to manage the Data Restore Functions.

Chapter 5. Installing a Support Machine

This chapter describes how to install a Control Center support machine. A support machine is operated in disconnect mode and is managed by a Control Center service machine. It is used for running Database Administration tools (database applications) scheduled through the service machine using the Job Scheduling tool. Unlike service machines, users do not communicate directly with support machines.

The installation process is similar to that of a shared service machine. It includes defining a support machine (MSTRSUP) with read access to the same code disk accessed by the managing service machine, and an owner machine (MSTROWN) for receiving CONSOLE files. Additionally, since a support machine is used to run Database Administration tools (database applications), it is necessary to grant DBA authority in all databases it will operate against.

Notes:

1. Multiple Control Center support machines may be installed on a single CPU and managed by a single Control Center service machine.
2. Instead of defining a new SQLMOWN machine, the owner machine of the managing service machine should also be the owner of the support machine.

Prior to installing a Control Center support machine you must install or have access to an existing Control Center service machine. Refer to Chapter 4, “Installing the Service Machine” on page 37.

Checklist for Installation of the Control Center Support Machine

Notes:

- *Perform the steps in order.*
- *Optional steps are preceded by a circle (○).*
- *Page references appear in parentheses.*

Preliminary Installation

__ 1. Log On to the VM MAINT Machine (52)

__ 2. Define a Control Center Support Machine (MSTRSUP) (53)

__ 3. ○ Authorization for SFS Databases (54)

__ 4. ○ Define a Control Center Owner Machine (MSTROWN) (54)

__ 5. Update the VM Directory (54)

__ 6. Link and Format the 191 Disk(s) (54)

__ 7. Add MSTRSUP Machine to Autolog List (55)

__ 8. Log Off the MAINT Machine (55)

Code Installation

__ 9. Log On to the MSTRSUP Machine (55)

__ 10. Access Code Disk (55)

__ 11. Configure Control Center Support Machine (55)

__ 12. Start Control Center Support Machine (58)

Installation Verification

__ 13. ○ Log On to the MSTROWN Machine (58)

__ 14. ○ Verify Files/Messages Received (58)

__ 15. ○ Link and Access the Control Center Code Disk (58)

__ 16. ○ Verify Control Center Support Machine Operation (58)

Post-Installation

__ 17. Authorize Communication (59)

__ 18. Grant DBA Authority (59)

Preliminary Installation

The first seven steps of the installation process prepare your VM system by defining all required virtual machines and their minidisks. Figure 10 summarizes what you will accomplish.

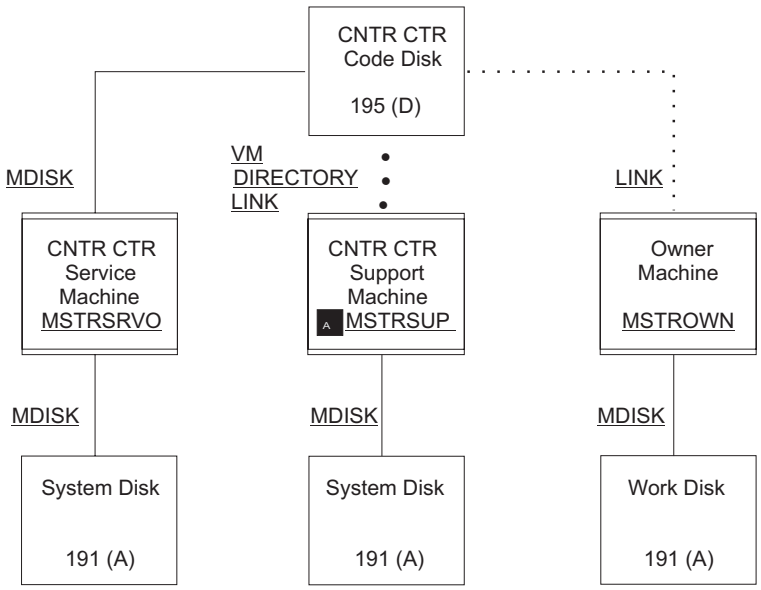


Figure 10. Virtual Machines and Minidisks in a Control Center Support Machine Environment

MSTRSUP **A** has been added to the environment shown in Figure 6 on page 38 using the support machine installation process. Service machine MSTRSRVO manages MSTRSUP and is the owner of the Control Center code disk. A shared service machine (MSTRSRVS) could have been used instead of MSTRSRVO; it would link to the same code disk as MSTRSUP. A single MSTROWN machine (installed during the service machine installation process) is defined as the logical owner of MSTRSRVO and MSTRSUP. You may, however, define a separate owner machine for the support machine.

Installation Step 1: Log On to the VM MAINT Machine

To begin the Control Center installation, log on to the MAINT virtual machine.

Installation Step step 2: Define a Control Center Support Machine (MSTRSUP)

To define an MSTRSUP machine which shares an existing Control Center code disk, update the VM directory statements as shown in Figure 11.

```
1 ──► USER MSTRSUP XXXXXXXX 6M 6M G
      ACCOUNT nnnnnnnn
2 ──► IUCV ANY
3 ──► IPL CMS
4 ──► CONSOLE     009          3215
      SPOOL       00C          2540          READER    *
      SPOOL       00D          2540          PUNCH     A
      SPOOL       OOE          1403          A
      LINK        MAINT        190    190    RR
      LINK        MAINT        19D    19D    RR
      LINK        MAINT        19E    19E    RR
5 ──► LINK        MSTRSRVO     195    547    RR
6 ──► MDISK  191  3380  cylr   100    volser MR
```

Figure 11. VM Directory Control Statements for a SQLMSUP Machine

Statement 1: USER MSTRSUP XXXXXXXX 6M 6M G

Defines the MSTRSUP machine as a 6-megabyte virtual machine with the VM privilege class G.

Statement 2: IUCV ANY

This statement allows MSTRSUP to communicate with MSTRSRV through the Inter-User Communications Vehicle (IUCV) feature of VM.

Statement 3: IPL CMS

This statement should use the CMS saved segment name which is applicable for your VM environment.

Statement 4: CONSOLE 009 3215

This statement *must not* include the **T userid** option (which specifies a secondary console for the MSTRSUP machine).

Note:

All Control Center functions will be disabled if the CONSOLE statement includes the **T userid** option.

Statement 5: LINK MSTRSRVO 195 547 RR

This statement links in READ mode to the same code disk linked to by the managing service machine. Depending on your environment, the code disk may be owned by a machine other than the managing service machine; (the managing service machine is sharing a code disk).

Note: MSTRSUP must be authorized to access the Control Center code disk in READ mode.

Statement 6: MDISK 191 3380 cylr 100 volser MR

An A-disk (191 minidisk) is required by the support machine to contain unloaded data, SQLDBSU (DBSU) control files, and DBSU listing files created by Database Administration tools. The user may request that a permanent or temporary minidisk be used by these tools. If a temporary disk is used, the DBSU listing files will be written to the A-disk to allow

the user to verify that the job ran correctly even after the temporary disk is released. If a permanent disk is used, all output files will be written to the permanent disk, which may be the A-disk or another read/write disk specified by the user.

The size of the 191 minidisk will vary greatly depending on the activities scheduled on the machine, size of data objects, and the options chosen when a tool is scheduled; (use a temporary disk).

All other VM directory entries in this example should correspond with those provided for any virtual machine on the target system.

Installation Step 3: Authorization for SFS Databases

When executing database administration tools such as SQLREORG, SQLMAINT, or SQLRINDX, the Control Center support machine must be:

- Enrolled as a USER in the SFS filepool where the production code directory is located
- Granted READ access on the production code directory and production code files
- Granted DBA on the servers where the database administration activities will be performed.

Installation Step 4: Define a Control Center Owner Machine (MSTROWN)

An MSTROWN machine should be defined at this time if such a machine does not already exist. MSTROWN will be designated later in the installation process as the owner of the MSTRSUP machine. (*Own* in this context describes a logical association between two virtual machines.)

No special VM directory changes are required for this machine. It must include an A-disk (191 minidisk) with approximately three cylinders of IBM 3380-equivalent DASD.

Note: Instead of defining a new machine, the owner machine of the managing service machine should also be the owner of the support machine.

Installation Step 5: Update the VM Directory

After you have made the VM directory control statement changes for the Control Center service machine and the Control Center owner machine, update the VM directory using your local operating procedures.

Installation Step 6: Link and Format the 191 Disk(s)

Link to the MSTRSUP 191 disk and format as the A-disk.

If a new MSTROWN machine was defined above, then you should also format the MSTROWN 191 disk as the A-disk.

Installation Step 7: Add MSTRSUP Machine to Autolog List

Under normal operations the MSTRSUP machine should be autologged each time the CPU is IPLed. The MSTRSUP machine should therefore be included in the list of virtual machines that are autologged each time the CPU is IPLed.

Important:

If your VM system has a facility installed which logs off inactive machines, place MSTRSUP in the list of machines to be excluded. This list is generally referred to as the log-off exclusion list.

Installation Step 8: Log Off the MAINT Machine

Release and detach the support machine's 191 disk, then log off the MAINT Machine.

Code Installation

Steps 9 through 12 install the Control Center code on the support machine.

Installation Step 9: Log On to the MSTRSUP Machine

Log on to the MSTRSUP virtual machine to complete the installation and configuration process.

Installation Step 10: Access Code Disk

Access the Control Center 547 code disk with an available disk mode.

Installation Step 11: Configure Control Center Support Machine

Use the C5648A70 EXEC supplied with Control Center to configure the Control Center software for your installation by entering,

```
C5648A70
```

at the CMS command prompt.

Note: The C5648A70 EXEC takes approximately 10 minutes to complete.

The C5648A70 EXEC will first verify that all Control Center code has been successfully installed on the code disk. It then prompts you for the following information about the type of installation being performed and the Control Center code disk being used:

1. The type of Control Center machine you are installing:

   Enter `SUPPORT`.

2. The virtual machine ID of the managing service machine.
3. The virtual machine ID of the code disk owner.
4. The virtual device address (CUU) to be used by the MSTRSUP machine when linking the code disk.
   - You must use the same CUU specified on the LINK VM directory statement in Installation Step 2 above.

5. The disk access mode to be used by the MSTRSUP machine when accessing the code disk.

Next, C5648A70 EXEC generates these Control Center control files, prompting you for information as required:

- SQLMSTR CONTROL
- SQLMSTR PROFILE
- PROFILE EXEC
- SQLMSTR TIMES

When file SQLMSTR CONTROL is generated, you are prompted for the information described below. You should review your needs ahead of time so you can respond properly for your installation. Once Control Center is operational on the support machine you can update file SQLMSTR CONTROL through the Control Center panel interface.

1. The number of Control Center support machine log files to retain.

   Enter the number of days you wish to keep a log file before it is automatically deleted. If no entry is made, the default (5) will be used.

   Control Center support machine log files contain a history of all interrupts within the support machine and are used primarily for debugging purposes. They are generated daily (named SQLMSTR LOG*yyddd*, where *yy* is the year and *ddd* is the Julian day) and are kept on the support machine's 191 disk before being automatically deleted after the specified number of days.

2. The number of backup copies of modified files to retain.

   Enter the number of past copies of modified files that you wish Control Center to keep. If no entry is made, the default (3) will be used.

   When a file is updated on the support machine's 191 A-disk, Control Center creates a backup copy of the original file with a date-coded filetype and places it on the same disk. Previous backup copies will be maintained by Control Center up to the number specified by this value.

3. The Control Center disk usage warning threshold.

   Enter an integer value between 50 and 98 for the disk full percentage which, when reached, causes a warning message to be sent to Control Center Administrators. If no entry is specified, the default value (70) is used.

   Each day at midnight Control Center determines whether the Control Center Administrators should be notified about the amount of data on the support machine's 191 A-disk. If the support machine's A-disk has reached or exceeded the percent full indicated by this value, then a message file will be sent to each Administrator notifying them of this condition.

4. The Control Center disk full warning threshold.

   Enter an integer greater than that of the previous response (and less than or equal to 99) for the disk full percentage which, when reached, causes a serious warning message to be sent to Control Center Administrators. If no entry is specified, then a value one greater than your previous response is used.

5. CMS Disk Access Options.

   Enter any valid CMS ACCESS command options to be used when accessing linked disks.

CMS disks will be linked and accessed by the database machine under Control Center control. Control Center will use any valid options for the CMS ACCESS command that are specified here. You can use the CMS HELP ACCESS command to review valid ACCESS command options available.

6. The job output retention period.

   Enter the number of days you wish to keep job output before it is automatically deleted. The minimum value is 1; the maximum value is 99999. If no entry is specified, the default value (30) is used.

7. Date/Time Format

   Control Center offers four different date/time formats to be used in all menu displays and reports generated. They are:

   - USA for mm/dd/yy and hh:mm:ss (this is the default format)
   - EUR for dd.mm.yyy and hh:mm:ss
   - ISO for yyyy-mm-dd and hh.mm.ss, and
   - JIS (same as ISO)

   Enter any of the above or blank to default to USA.

You should now receive a message indicating that file SQLMSTR CONTROL has been generated.

When file SQLMSTR PROFILE is generated, you are prompted for the information described below. You should review your needs ahead of time so you can respond properly for your installation.

1. The logical owner of the MSTRSUP machine.

   Enter the machine identifier (user ID) that should be recognized as the owner of the Control Center support machine. The owner machine is referred to in this manual as MSTROWN, but it can be any existing virtual machine. If no entry is specified, the current log-on ID is used.

   This machine will automatically have Control Center Administrator authority and will additionally receive the SQLMSTR CONSOLE file that is closed and transferred from the support machine each day at midnight.

2. One or more users who will be Control Center Administrators.

   Respond in the form `userid` or `userid at nodeid`. This prompt will be displayed repeatedly until you enter a blank response (or `Q` to terminate the install process early).

   Each user ID entered will be given Control Center Administrator authority to the Control Center support machine, allowing them complete control and authority to perform any Control Center function for any database.

   Note: Control Center Administrators may be added and deleted through the Control Center panel interface once Control Center is operational on the support machine.

You should now receive several messages indicating that files SQLMSTR PROFILE, PROFILE EXEC and SQLMSTR TIMES have been generated, and that Control Center parameters have been loaded into memory.

This completes C5648A70 EXEC processing.

Installation Step 12: Start Control Center Support Machine

This step will initialize and start the Control Center support machine.

Enter these commands at the CMS command prompt:

```
#CP IPL CMS
SQLMSUP
```

The support machine should immediately be disconnected from your active terminal session and the startup process will be completed in disconnect mode. You will be logged off the terminal at this point. The installation process is complete.

Installation Verification

The following steps will determine whether the Control Center support machine is functioning.

Installation Step 13: Log On to the MSTROWN Machine

Log on to the MSTROWN machine or any machine authorized previously as a Control Center Administrator on the support machine.

Installation Step 14: Verify Files/Messages Received

If logged on to the MSTROWN machine, you should have received file SQLMSTR CONSOLE (sent by the MSTRSUP machine) in the virtual reader.

If, during Control Center startup processing, you are logged on to a machine with Control Center Administrator authority (including MSTROWN), you should receive a Control Center startup message from MSTRSUP.

Note: You *must* be logged on at the time Control Center startup processing on MSTRSUP issues the message for it to appear on your console. Continue with the verification steps even if you do not receive a message.

Installation Step 15: Link and Access the Control Center Code Disk

Link and access the Control Center code disk in READ mode with an available virtual device address (CUU) and access mode (195 D).

Installation Step 16: Verify Control Center Support Machine Operation

To establish communications with MSTRSUP, enter this at the CMS command prompt:

```
SQM (MSTRSUP
```

Replace `MSTRSUP` with the machine ID of your support machine.

Verify the support machine is operating by entering,

```
SQM VERSION
```

Chapter 10, “Version Tool” on page 107 Control Center will return a message identifying the support machine (MSTRSUP) and the Control Center product release and modification level. If this is successful, Control Center has been successfully installed on the support machine and the verification process is complete.

For more information about SQM VERSION, refer to Chapter 10, “Version Tool” on page 107.

Post-Installation

Installation Step 17: Authorize Communication

The support machine must be authorized to send job completion messages to the managing service machine which controls the job schedule.

In brief, the steps to do this are:

1. Log on to a machine (user ID) that has Control Center Administrator authority on the managing Control Center service machine; for example, the owner machine of MSTRSRV.

   Note: Do *not* log on to the service machine.

2. Authority for MSTRSUP can now be added using the Control Center panel interface.

   a. Start Control Center in panel mode by entering this at the CMS command prompt:

      ```
      SQM
      ```

   b. Select Option AU (SQLMSTR Authorization) on the Control Center main menu.
   c. Select Option M (MUM Machine) on the SQLMSTR Authorization panel.[3]
   d. Enter the machine ID (user ID) and node ID of the support machine and press ENTER.

Installation Step 18: Grant DBA Authority

The support machine must be given DBA authority for each database it services. Issue the **GRANT DBA** authority by using the ISQL facility of the database.

[3] A MUM machine is the same as a Control Center support machine.

Chapter 6. Database Setup

This chapter describes how to set up a database to be managed by a Control Center service machine.

Prior to performing the database setup process detailed in this chapter you should have:

- Installed database and/or
- Generated a database, and
- Installed a Control Center service machine. Refer to Chapter 4, “Installing the Service Machine” on page 37 in this manual.

Database Setup Considerations

Prior to setting up a database to be managed by a Control Center service machine, you will need to make some decisions about the operation of the database. Control Center allows you to manage the operational functions of databases in different ways, adapting to the needs of each specific database. Decisions such as logmode; what type of archiving will be used (database or User); whether archives will use predefined tapes, scratch tapes, or disk files; and how often archives will be taken, need to be made prior to establishing a database under Control Center.

Naming Convention

Another important consideration for a database is the naming convention chosen for the database machine name and the database dbname. For ease of use and to eliminate confusion, the database machine name should be identical to the dbname. The System Administration tools rely on an IUCV communication between the Control Center service machine and the database machine. Most references within this manual and the Control Center interface for these tools will therefore use the term *database* to refer to the *database machine name*. Only Database Administration tools, which interface with the database as database applications, utilize the database dbname instead of the database machine name.

Control Center provides the capability to assign a database machine nickname in file SQLMSTR DIRECTRY. Therefore, if a database machine name differs from the dbname, it can be assigned a nickname equal to the dbname. This will allow users to enter either the database machine name or nickname (dbname) when using the Control Center panel interface. Control Center will automatically convert the nickname to the machine name prior to executing the requested tool.

Checklist for Setting up a Database to Work with Control Center

Notes:

- *Perform the steps in order.*
- *Optional steps are preceded by a circle (○).*
- *Page references appear in parentheses.*

Preliminary Setup

__ 1. Log On to the VM MAINT Machine (63)

__ 2. Redefine the Database Machine (SQLMACH) (63)

__ 3. Update the VM Directory (65)

__ 4. Add SQLMACH Machine to Autolog List (65)

__ 5. Log Off the MAINT Machine (65)

Control Files Setup

__ 6. Log On to the MSTROWN Machine (66)

__ 7. Link and Access the Control Center Code Disk (66)

__ 8. Start Control Center Session (66)

__ 9. Define Database Parameters File (67)

__ 10. Define Database TAPES File (68)

__ 11. ○ Define Database ARISPOOL File (74)

__ 12. End Control Center Session (75)

__ 13. Migrate SQLMSTR DIRECTRY and DBINIT CONTROL Files (75)

__ 14. ○ SQLMSTR DIRECTRY Customization (76)

__ 15. Log Off the MSTROWN Machine (78)

Database Machine Setup

__ 16. Log On to the Database Machine (SQLMACH) (78)

__ 17. Link and Access the Control Center Code Disk (78)

__ 18. Copy and Update DBPROF EXEC (78)

__ 19. Run DBSTART EXEC (80)

Setup Verification

__ 20. ○ Log On to the MSTROWN Machine (80)

__ 21. ○ Verify Messages Received (80)

__ 22. ○ Link and Access the Control Center Code Disk (80)

__ 23. ○ Review the Database Startup File (81)

Preliminary Setup

Steps 1 through 5 prepare your database machine for interfacing with the Control Center product. Figure 12 on page 63 summarizes what you will accomplish.

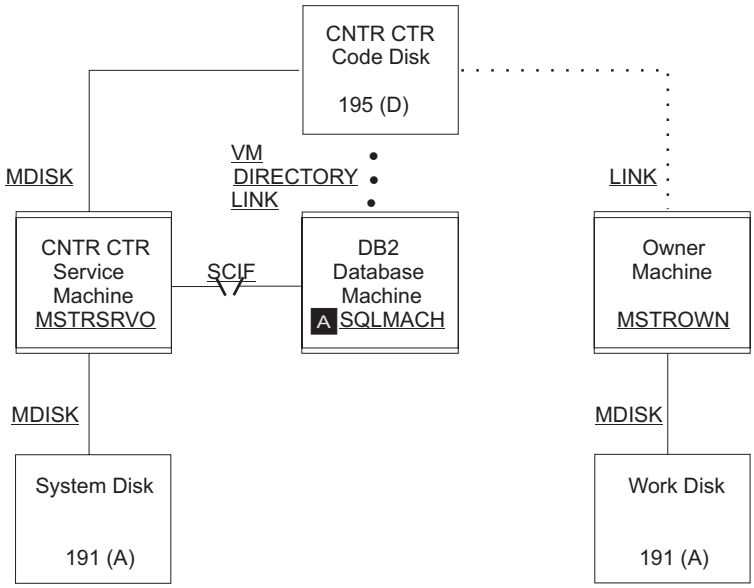


Figure 12. Database Machine in a Control Center Environment. Database machine SQLMACH **A** has been added to the environment shown in Figure 6 on page 38 using the database setup process. Service machine MSTRSRVO manages SQLMACH and is the owner of the Control Center code disk. A shared service machine (MSTRSRVS) could have been used instead of MSTRSRVO; it would link to the same code disk as SQLMACH. Communications between SQLMACH and MSTRSRVO is through the SCIF interface.

Setup Step 1: Log On to the VM MAINT Machine

To begin the database machine setup, log on to the MAINT virtual machine.

Setup Step 2: Redefine the Database Machine (SQLMACH)

Change the VM directory entries for the database machine to activate the Secondary Console Interface Facility (SCIF) and to link the Control Center code disk. Refer to the appropriate Virtual Machine Operation manuals for a complete description of these statements.

```
        USER SQLMACH XXXXXXXX 4M 6M G
          ACCOUNT nnnnnnnn
          OPTION ACCT              ECMODE MAXCON 256
          IUCV ALLOW
          IPL CMS
1 ------> CONSOLE   009      3215   T      MSTRSRVO
          SPOOL     00C      2540          READER     *
          SPOOL     00D      2540          PUNCH      A
          SPOOL     00E      1403          A
          LINK      MAINT    190    190    RR
          LINK      MAINT    19D    19D    RR
          LINK      MAINT    19E    19E    RR
          LINK      MSTRSVO  195    547    RR
2 ------> MDISK 191 3380 cylr 020 volser MR

          * DB2 Server for VM 6.1
          MDISK 193 3380 cylr 043 volser MR
          MDISK 195 3380 cylr 015 volser MR
          * DIRECTORY
          MDISK 200 3380 cylr 075 volser MR
          * LOGS
          MDISK 201 3380 cylr 075 volser MR
          MDISK 220 3380 cylr 075 volser MR
          * CATALOG'S
          MDISK 202 3380 cylr 075 volser MR
          * USER DBEXTENTS
          MDISK 203 3380 cylr 100 volser MR
          MDISK 204 3380 cylr 200 volser MR
```

Figure 13. Example VM Directory Control Statements for a Database Machine

Statement 1: CONSOLE 009 3215 T MSTRSRVO

Defines the managing Control Center service machine as being the secondary console of the database machine. This is done by including **T MSTRSRVO** in the CONSOLE statement as shown.

Note: The managing service machine can either be the owner of the code disk or a service machine that shares a code disk.

Statement 2: LINK MSTRSRVO 195 547 RR

This statement links in READ mode to the same code disk linked to by the managing service machine. Depending on your environment, the code disk can be owned by a machine other than the managing service machine; (the managing service machine is sharing a code disk).

Instead of including the above statement in the VM directory, you may, during a later step, add a LINK statement to the PROFILE EXEC of the SQLMACH machine.

Note: SQLMACH must be authorized to access the Control Center code disk in READ mode.

Important:

1. The IPL CMS statement should *not* include PARM AUTOCR. PARM AUTOCR causes an extra console interrupt to occur on the database machine when it is autologged. This results in problems with startup communications between the database machine and the Control Center service machine.
2. SYSPROF EXEC should be checked to avoid additional console interrupts from being stacked during database startup. If the SYSPROF EXEC causes a problem, the IPL statement in the database machine directory can be modified to IPL CMS PARM NOSPROF.
3. If minidisk passwords are used, please read Appendix C, “Password Support” on page 501 for more information.

Setup Step 3: Update the VM Directory

After you have made the VM directory statement changes for the database machine, update the VM directory using your local operating procedures.

Setup Step 4: Add SQLMACH Machine to Autolog List

Under normal operations the SQLMACH machine should be autologged each time the CPU is IPLed. The SQLMACH machine should therefore be included in the list of virtual machines that are autologged each time the CPU is IPLed.

For proper startup of each database when the CPU is IPLed, the database machine *must* be autologged *after* the managing Control Center service machine.

Important:

1. The Control Center service machine *must* be autologged *prior* to the database machine for SCIF communication to be properly established. Additionally, portions of the database startup process require that the Control Center service machine be fully operational.
2. If problems occur with the database machine startup when the CPU is IPLed, then a CP SLEEP 1 MIN statement should be added near the beginning of the database machine's PROFILE EXEC. This will delay the startup of the database machine for 1 minute, giving the Control Center service machine a chance to become fully operational.

Setup Step 5: Log Off the MAINT Machine

Preliminary setup is now complete. Log off the MAINT machine.

Control Files Setup

Steps 6 through 15 generate the database control files listed below. These files are kept on the managing service machine's 191 A-disk.

- Database parameters file (SQLMACH PARMS)
- Database TAPES file (SQLMACH TAPES)

- Database storage pool specification file (SQLMACH ARISPOOL) (optional)

Additionally, Control Center control files SQLMSTR DIRECTRY and DBINIT CONTROL are updated with entries for the new database. These files are kept on the service machine's 195 code disk.

Important:

If the managing service machine is using a shared code disk, additional steps are required to migrate the updated Control Center control files to the code disk.

The Control Center service machine *must* be operational to complete steps 8 through 12 and for the verification process. Refer to “Installation Step 14: Start Control Center Service Machine” on page 46.

Setup Step 6: Log On to the MSTROWN Machine

Log on to the MSTROWN machine or any machine that has Control Center Administrator authority on the service machine specified on the CONSOLE statement in Setup Step 2 (the managing Control Center service machine).

Setup Step 7: Link and Access the Control Center Code Disk

Link and access the Control Center code disk in READ mode with an available virtual device address (CUU) and access mode (195 D).

Notes:

1. Depending on your environment, the code disk may already be linked and accessed.
2. Depending on your environment, the code disk may be owned by a machine other than the managing service machine; (the managing service machine is sharing a code disk).

Setup Step 8: Start Control Center Session

To establish communications with MSTRSRV and to start Control Center in panel mode, enter these commands at the CMS command prompt:

```
SQM (MSTRSRV
SQM
```

Replace `MSTRSRV` with the machine ID of your service machine.

The Control Center Main Menu should now be displayed. Select Option **N** (New Database Setup) to display the panel shown in Figure 14. Enter the database machine name in field **A**.

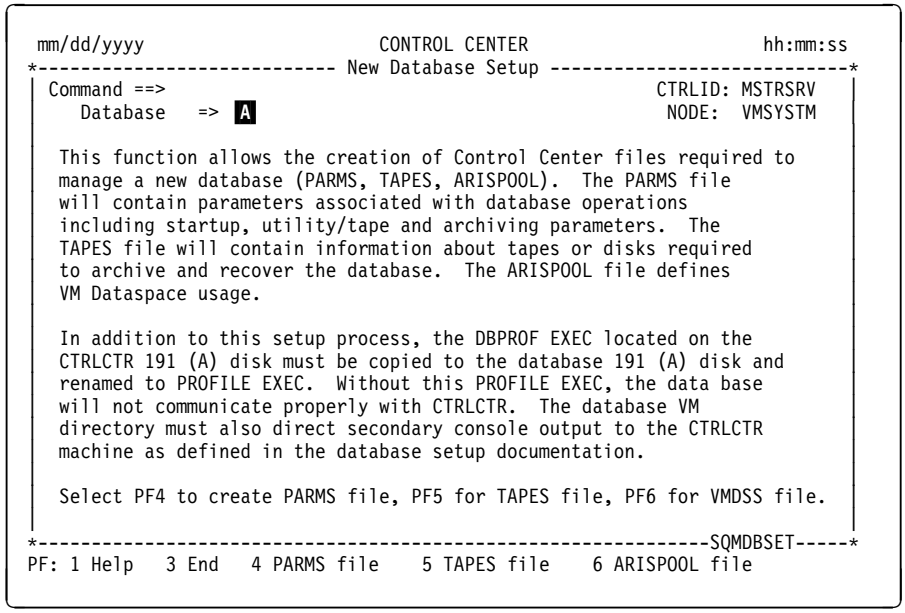


Figure 14. New Database Setup Panel

Setup Step 9: Define Database Parameters File

Press PF4 (PARMS file) to create the database parameters file. Refer to “About the Database Parameters Tool” on page 213 for a complete description of the Database Parameters tool.

About the Database Parameters File

Control Center maintains a set of database startup and control parameters for each database machine it manages. Parameters for each database managed by the service machine are saved in separate files named *database* PARMS, where *database* is the database machine ID, and are kept on the Control Center service machine's 191 A-disk. Database parameters are separated into the general groups:

1. Authorization/Notification parameters
2. Utility parameters
3. Tape parameters
4. Archiving/Tracing parameters
5. Database startup parameters
6. Data Restore parameters

Complete information about the database parameters file, including the Data Restore enablement, can be found in “Database Parameters File” on page 214.

Setup Step 10: Define Database TAPES File

Terminology Used in this Section:

1. The term *output media* refers to both *tape* (cartridge or reel-to-reel) and *disk* (CMS file) unless explicitly stated otherwise, as in "tape output medium".
2. The use of the term *tapes* when referencing the "tapes file" is not meant to exclude disk output medium.
3. The use of the term *tapes* in the Control Center panel interface is not meant, in most cases, to exclude disk output medium.

Press PF5 (TAPES file) to create the database TAPES file. Refer to Chapter 15, “Tape Management Tool” on page 185 for a complete description of the Tape Management tool.

About the Database TAPES File

A separate database TAPES file is maintained by Control Center for each database machine it manages. Tapes files are named *database* TAPES, where *database* is the database machine ID, and are kept on the Control Center service machine's 191 A-disk.

Control Center uses the database TAPES file to maintain a list of output media assigned for usage during database archive, log archive, and trace activities. It must be defined prior to starting the database under Control Center control. When the database is started, FILEDEF and LABELDEF commands will be issued for the output media identified in this file. Control Center references and updates this file as necessary during archive, log archive, and trace activities.

It is the responsibility of the database administrator to assure that the TAPES file is created correctly, with valid tape volids and with series numbers that match those identified in the database parameters file. The SERIES parameters in the parameters file indicate which group of tapes within the TAPES file will be used for the next archive. After initially setting these parameters, Control Center will maintain the values automatically when archives are performed.

A full database archive will use as many archive tapes as it needs that have the same series number. Each subsequent log archive after a full archive will use tapes within the same series as the previous full archive. Even though tapes may be available elsewhere within the TAPES file, Control Center will not use them unless they have the same series number as that specified within the database parameters file.

Important:

Output media with the same series number are considered by Control Center to belong together as a single logical group.

An important consideration when creating this file is determining how many series of tapes should be included. At least three series should be created, although Control Center will operate with fewer. It should be remembered that if an archive fails, you would want to have at least one previous successful archive to fall back on.

Each tape included in the TAPES file should be labelled and assigned to the database machine which will be using it. This must be done by using whatever tape management system is installed on your system.

Scratch Tape Acquisition

If the DYNAM/T or VMTAPE product is used at your site, the SCRATCH TAPE ACQUISITION **(S)** option of the Tape Management tool can be used to add tapes to this file. This function will request that a scratch tape be mounted, and will acquire ownership of the tape for the indicated retention period and place it in the database TAPES file.

Usage Consideration: If you have DYNAM/T installed and use DYNOPEN as your tape method, the SCRATCH TAPE ACQUISITION tool cannot be used. Only DYNMOUNT supports the use of scratch tape acquisition. Check the SQLMSTR Control file to determine if DYNOPEN has been selected as the tape method for DYNAM/T , if this is the case, scratch tape acquisition will fail.

Important:

Check your tape facility to ensure that the RETENTION period parameter is implemented correctly.

If you are running under logmode Y or A, therefore doing only full database archives, you will only need to include ARCHIVE tapes within the TAPES file. You must assure that enough tapes are listed for each series to perform a complete database archive. You should also have at least one extra tape available in each series to allow for database growth. When a database starts up, every ARCHIVE tape in the specified series will be LABELDEFed for a multivolume archive. If enough tapes are not available to complete the archive, the results will be unpredictable (dependent upon the tape management system being used).

If you are running under logmode L, performing log archiving, then you will need to include LOG tapes within the TAPES file. These tapes will be used to dump the database log file when it becomes full.

Important:

Control Center expects that the entire log file will fit on a single LOG tape. This will enable log archives to be taken while the database remains up. If the log file is larger than what will fit on a single tape, then Control Center will not operate properly. This implementation is due to a restriction in CMS and database which prevents the LABELDEF command from being reissued while the database is running.

If you are doing tracing within the database, then you will need to include TRACE tapes within the TAPES file.

TAPES File Format

Shown below is the TAPES file format.

Table 4. TAPES File Format

Value	Description
SERIES	A three-digit number indicating the output media series. Output media with the same series number are considered by Control Center to belong together as a single logical group.
TYPE	Assigns output media to a specific usage. Possible values: **ARCHIVE** Use for full database archiving activities. **ssINCBKd** Use for each Incremental Backup. "ss" is the subseries number and "d" is blank (indicating the primary) or "2" (indicating the secondary) Incremental Backup. **LOG** Use for database log archiving activities. **TRACE** Use for database tracing activities. **BACKUP** Use for primary user archives using Data Restore BACKUP. **BACKUP2** Use for secondary user archives using Data Restore with dual BACKUP. **TRANS** Use for translation of database archives into Data Restore BACKUP format. **TRANSDSK** Use for the disk location of the 3 Data Restore work files needed for translation (SYS0001, DIRWORK, & HEADER). **UNLOADID** User generated id for unloading DBSPACES using Data Restore. Used in the UNLTAPES file only.
DATE	The Julian date (*yyddd*) when output media was last used by Control Center. Should be equal to 00000 when tape is unused.
TIME	Time (*hh:mm:ss*) when output media was last used by Control Center. Should be equal to 00:00:00 when tape is unused.
STATUS	Control Center uses this field to track the status of the output media. Common status values include: **UNUSED** Output media is not being used. **FILDEF** Output media currently being used. **MOUNTD** Output media is currently mounted. **FILLED** Output media has been used.
VOLID	Tape volume serial number if output media is TAPE.
FILENAME	CMS filename if output media is DISK.
FILETYPE	CMS filetype if output media is DISK.
FM	CMS filemode (disk access mode) if output media is DISK.
CUU	*Virtual* disk address (CUU) if output media is *DISK* and *TYPE* is ARCHIVE, LOG or TRACE. *Real* disk address (CUU) if output media is *DISK* and *TYPE* is a *Data Restore* function (*TRANSLATE, BACKUP2, INCBK, BACKUP, or UNLOAD*).
CI	Century Indicator (1 = 20th Century, 2 = 21st Century) This is not displayed on the panels.

Archives to Tape Examples

The following is an example database TAPES file for archiving to tape. It specifies that Control Center will rotate three sets of archive tapes (series 100, 200, 300). It also indicates that logmode L (log archiving) will be used. Each series has three ARCHIVE tapes, allowing for a database size that will fit entirely on three tapes. Each series also has two LOG tapes. This essentially allows for only one log archive between full database archives, since the first step of a full database archive under logmode L will be to perform a Log Archive.

You must have one LOG tape more than you expect to need between full database archives. Control Center will request a scratch tape to be mounted when no tapes are available for a given series in the database TAPES file. This cannot perform properly if VMTAPE or DYNAM/T is not being used. In an effort to prevent running out of tapes, Control Center will warn the DBA(s) whenever the last tape in a series is mounted.

```
                                        VOLID/
SERIES  TYPE     DATE   TIME     STATUS  FILENAME  FILETYPE  FM   CUU   CI
 100    ARCHIVE  00000  00:00:00 UNUSED  VOL100
 100    ARCHIVE  00000  00:00:00 UNUSED  VOL101
 100    ARCHIVE  00000  00:00:00 UNUSED  VOL102
 100    LOG      00000  00:00:00 UNUSED  VOL110
 100    LOG      00000  00:00:00 UNUSED  VOL111
 200    ARCHIVE  00000  00:00:00 UNUSED  VOL200
 200    ARCHIVE  00000  00:00:00 UNUSED  VOL201
 200    ARCHIVE  00000  00:00:00 UNUSED  VOL202
 200    LOG      00000  00:00:00 UNUSED  VOL210
 200    LOG      00000  00:00:00 UNUSED  VOL211
 300    ARCHIVE  00000  00:00:00 UNUSED  VOL300
 300    ARCHIVE  00000  00:00:00 UNUSED  VOL301                    @dl
 300    ARCHIVE  00000  00:00:00 UNUSED  VOL302
 300    LOG      00000  00:00:00 UNUSED  VOL310
 300    LOG      00000  00:00:00 UNUSED  VOL311
```

Figure 15. Example TAPES File for Archiving to Tape

The matching entries in the database parameters file for the initial database setup would be:

```
Archive-media     => TAPE
Archive-blksize   => 28672
Archive-series    => 100

Logarch-media     => TAPE
Logarch-blksize   => 28672
Logarch-series    => 300
```

Figure 16. Matching Database Parameters for Archiving to Tape

The following is an example database TAPES file for archiving to tape using Incremental Backup. Series 100 consists of a set of dual backup entries for primary backup and secondary backup and three sets of dual Incremental Backup. Each tape belonging to an Incremental Backup within series 100 is grouped together by the subseries prefix. If the database is using logmode L (log archiving), then a log archive would have been performed before each BACKUP and Incremental Backup.

```
1           100  ARCHIVE  98032  16:37:21 FILLED QU1371
              100  ARCHIVE  98032  16:46:58 FILLED QU1372
              100  BACKUP2  98032  16:37:21 FILLED QU1373
              100  BACKUP2  98032  16:41:08 FILLED QU1374
2           100  01INCBK  98033  17:50:01 FILLED QU1375
              100  01INCBK2 98033  17:50:02 FILLED QU1376
3           100  02INCBK  98034  17:30:01 FILLED QU1377
              100  02INCBK  00000  00:00:00 UNUSED QU1378
              100  02INCBK2 98034  17:30:02 FILLED QU1379
              100  02INCBK2 00000  00:00:00 UNUSED QU1380
4           100  03INCBK  98035  17:50:10 FILLED QU1381
              100  03INCBK  98035  17:50:15 FILLED QU1381
              100  03INCBK  00000  00:00:00 UNUSED QU1382
              100  03INCBK2 98035  17:50:11 FILLED QU1383
              100  03INCBK2 98035  17:50:17 FILLED QU1384
              100  03INCBK2 00000  00:00:00 UNUSED QU1385
5           200  ARCHIVE  98036  16:37:21 FILLED QU1386
              200  ARCHIVE  98036  16:46:58 FILLED QU1387
              200  BACKUP2  98036  16:37:21 FILLED QU1388
              200  BACKUP2  98036  16:41:08 FILLED QU1389
6           200  01INCBK  98037  17:50:01 FILLED QU1375
              200  01INCBK2 98037  17:50:02 FILLED QU1376
              200  02INCBK  98038  17:30:01 FILLED QU1377
              200  02INCBK  00000  00:00:00 UNUSED QU1378
              200  02INCBK2 98038  17:30:02 FILLED QU1379
              200  02INCBK2 00000  00:00:00 UNUSED QU1380
              200  03INCBK  98039  17:50:10 FILLED QU1381
              200  03INCBK  98039  17:50:15 FILLED QU1381
              200  03INCBK  00000  00:00:00 UNUSED QU1382
              200  03INCBK2 98039  17:50:11 FILLED QU1383
              200  03INCBK2 98039  17:50:17 FILLED QU1384
              200  03INCBK2 00000  00:00:00 UNUSED QU1385
```

Figure 17. Example TAPES File using Incremental Backup

1 — On Sunday a dual Full Backup was performed using series 100.

2 — On Monday, an Incremental Backup was performed using the tapes in series 100 and subseries 01 (01INCBK).

3 — On Tuesday, an Incremental Backup was performed using subseries 02 (02INCBK).

4 — On Wednesday, two tapes were used for both the primary and secondary Incremental Backup using subseries 03 (03INCBK and 03INCBK2).

5 — On Thursday, a new Full Backup was executed using series 200.

6 — On Friday, a new set of Incremental Backup.s are executed in series 200 starting with subseries 01 (01INCBK).

Archives To Disk Example

The following is an example database TAPES file for archiving to disk. It specifies that Control Center will rotate three sets of archive files (series 100, 200, 300). It also indicates that logmode L (log archiving) will be used. Each series has one ARCHIVE file, since the entire database must fit on a single FILEDEFed minidisk. Each series also has two LOG files. This essentially allows for only one log archive between full database archives, since the first step of a full database archive under logmode L will be to perform a log archive.

Important:

There are several restrictions and special precautions that apply when archiving to disk rather than tapes. These are discussed in detail in “About Archiving to Disk” on page 236.

```
                                        VOLID/
 SERIES  TYPE     DATE   TIME     STATUS   FILENAME  FILETYPE  FM   CUU   CI
  100    ARCHIVE  00000  00:00:00 UNUSED   ARCHFN1   ARCHFT1   H    500
  100    LOG      00000  00:00:00 UNUSED   LOGFN1    LOGFT1    I    501
  100    LOG      00000  00:00:00 UNUSED   LOGFN2    LOGFT2    I    501
  200    ARCHIVE  00000  00:00:00 UNUSED   ARCHFN2   ARCHFT2   J    502
  200    LOG      00000  00:00:00 UNUSED   LOGFN3    LOGFT3    K    503
  200    LOG      00000  00:00:00 UNUSED   LOGFN4    LOGFT4    K    503
  300    ARCHIVE  00000  00:00:00 UNUSED   ARCHFN3   ARCHFT3   L    504
  300    LOG      00000  00:00:00 UNUSED   LOGFN5    LOGFT5    M    505
300 LOG 00000 00:00:00 UNUSED LOGFN6 LOGFT6 M 505
```

Figure 18. Example TAPES File for Archiving to Disk

The matching entries in the database parameters file for the initial database setup would be:

```
Archive-media => DISK Archive-blksize => 4096 Archive-series => 100

 Logarch-media     => DISK
 Logarch-blksize   => 4096
 Logarch-series    => 300
```

Figure 19. Matching Database Parameters for Archiving to Disk

Using Scratch Tapes Example

The following is an example database TAPES file for archiving to scratch tapes. This option is currently only available for those installations which use the VMTAPE or DYNAM/T product.

This setup is designed to allow you to stack multiple scratch tapes in a cartridge tape hopper for use by Control Center at night without operator intervention (see Appendix H, Tape Hopper Support). Since Control Center will not be requesting a specific volid for each tape mount, this option eliminates the need for intervention to mount the correct tape.

For Control Center to handle this option properly, the database parameter Scratch-tape-option (in the database parameters file) must be assigned the value Y, and the database TAPES file must be set up in a special way, with dummy placeholder entries for each tape type in each series. Control Center will request scratch tapes for each mount and will add them into the TAPES file automatically, for a possible future recovery. When Control Center rotates through the different series within the TAPES file, any previously added tapes will automatically be removed when the archive begins.

Usage Consideration: Only ARCHIVE tape *placeholders* are required for this option, even when using logmode L (log archiving). Control Center will automatically request scratch tapes for log archives when there are no pre-assigned volids available, even when this special option is not being used.

```
                                        VOLID/
SERIES  TYPE     DATE   TIME     STATUS   FILENAME  FILETYPE  FM   CUU   CI
 100    ARCHIVE  00000  00:00:00 UNUSED   SCRATCH
 200    ARCHIVE  00000  00:00:00 UNUSED   SCRATCH
 300    ARCHIVE  00000  00:00:00 UNUSED   SCRATCH
```

Figure 20. Example TAPES File for Archiving to Scratch Tapes

The matching entries in the database parameters file for the initial database setup would be:

```
Scratch-tape-option => Y

Archive-media     => TAPE
Archive-blksize   => 28672
Archive-series    => 100

Logarch-media     => TAPE
Logarch-blksize   => 28672
Logarch-series    => 300
```

Figure 21. Matching Database Parameters for Archiving to Scratch Tapes

Setup Step 11: Define Database ARISPOOL File

Press **PF6** (ARISPOOL file) to create the storage pool specifications file. Refer to “About the Database Parameters Tool” on page 213 for a complete description of the Database Parameters tool used to create the ARISPOOL file.

About the Storage Pool Specification File

The storage pool specifications file allows you to control the database VM Data Spaces feature at the individual storage pool level. Storage pool specifications for each database managed by the service machine are saved in separate files named *database* ARISPOOL, where *database* is the database machine ID, and are kept on the Control Center service machine's 191 A-disk.

Three VMDSS specifications can be specified for storage pools:

1. Whether data spaces support or the standard database DASD I/O system is to be used (data space support is the default).
2. The working storage residency priority, for those pools that use data spaces (a residency priority of 3 is the default).
3. Whether or not striping is used (striping is the default).

These settings can be changed dynamically while the database is running by issuing specific database operator commands. Dynamic changes, however, do not change the settings in the database's storage pool specifications file, the file used at database startup time.

For more detailed information on using VMDSS, refer to the *DB2 Data Spaces Support* manual.

Setup Step 12: End Control Center Session

Exit the Control Center panel interface using **PF3**.

Setup Step 13: Migrate SQLMSTR DIRECTRY and DBINIT CONTROL Files

Important:

This step is required *only* if the managing Control Center service machine does NOT own the code disk.

Control Center control files SQLMSTR DIRECTRY and DBINIT CONTROL are automatically updated with entries for the new database. These files are kept on the Control Center 195 code disk. If the managing service machine owns the code disk, the updated files are automatically migrated to the code disk. However, if the service machine is using a shared code disk, Control Center will be unable to migrate the files and will, instead, return the updated files to your virtual reader. Additional steps are then required to migrate the updated control files to the code disk. These steps are outlined below.

Important:

You should always migrate files SQLMSTR DIRECTRY and DBINIT CONTROL to the code disk after each database setup.

You will need access to a machine that has Control Center Administrator authority on the service machine that owns the code disk. (The machine having Administrator authority is referred to as MSTROWNO; the service machine owning the code disk is referred to as MSTRSRVO.) Depending on your environment, this machine may be different than the one used in previous steps.

Preliminary Steps:

1. Complete these steps only if the machine used to perform the database setup process (MSTROWN) does *not* have Control Center Administrator authority on the service machine that owns the code disk.

1. Receive files SQLMSTR DIRECTRY and DBINIT CONTROL from the virtual reader to the A-disk using the CMS RECEIVE command.
2. Send files SQLMSTR DIRECTRY and DBINIT CONTROL to MSTROWNO using the CMS SENDFILE command.
3. Erase files SQLMSTR DIRECTRY and DBINIT CONTROL from the A-disk.
4. Log off the MSTROWN machine.
5. Log on to the MSTROWNO machine.

File Migration Steps:

1. Receive files SQLMSTR DIRECTRY and DBINIT CONTROL from the virtual reader to the A-disk using the CMS RECEIVE command.

2. Send files SQLMSTR DIRECTRY and DBINIT CONTROL to MSTRSRVO using the CMS SENDFILE command.

MSTRSRVO will automatically receive the files to the code disk after first making backup copies of the existing files. You should receive these confirmation messages from MSTRSRVO after the files have been received:

```
07:48:23  * MSG FROM MSTRSRVO: SQMRDR: SQLMSTR  DIRECTRY D received by MSTRSRVO at VMSYSM1
07:48:23  * MSG FROM MSTRSRVO: SQMRDR: DBINIT   CONTROL  D received by MSTRSRVO at VMSYSM1
```

3. Erase files SQLMSTR DIRECTRY and DBINIT CONTROL from the A-disk.
4. Log off the MSTROWNO machine.
5. Log back on to the MSTROWN machine to complete the Control File Setup Process.

Setup Step 14: SQLMSTR DIRECTRY Customization

Depending on the naming convention chosen for the database machine name and database name (dbname) or whether your environment includes database machines managed by Control Center on remote nodes, it may be necessary to change the default database machine nickname in file SQLMSTR DIRECTRY. (The default nickname is equal to the database machine name.)

In brief, the steps for manually updating file SQLMSTR DIRECTRY are:

1. Copy SQLMSTR DIRECTRY from the code disk to the A-disk.
2. Update using XEDIT.
3. Migrate the updated file to the code disk by sending it to the service machine that owns the code disk. This must be done from a machine which has Control Center Administrator authority on the service machine that owns the code disk.

   Note: See the previous step for an example on how to migrate a file to the code disk.

SQLMSTR DIRECTRY File Format and Example

Table 5. SQLMSTR DIRECTRY File Format

Field	Description
1	Database Machine Name or Database Nickname
2	Database Machine Name or Virtual Machine ID
3	Control Center service machine Name
4	Control Center service machine Node ID

```
SQLDBA1  SQLMACH  MSTRSRV1 VMSYSM1
SQLDBA2  SQLMACH  MSTRSRV2 WMAVM1
CPVEND   CPVENDBA SQMSTTS1 WMAVM1
TOTDBA2  TOTDBA   SQLMSTR  MANVM2
TOTDBA7  TOTDBA   SQLMSTR  WMAVM7
SQMSTDBA CPMICDBA SQLMSTR  WMAVM1
TOTTST7  TOTTST   SQLMSTR  WMAVM1
ACCTDBA  ACCTDBA  SQLMSTR  WMAVM1
STARDBA  STARDBA  SQLMSTR  WMAVM1
```

Figure 22. Example SQLMSTR Directry File

The first two entries from the example SQLMSTR DIRECTRY file contain databases that each have the same machine name but reside on different nodes. In order to identify each database, we need to specify unique nicknames

(SQLDBA1 and SQLDBA2). The first database exists on VMSYSM1 and communicates through the MSTRSRV1 machine. The second database exists on WMAVM1 and communicates through the MSTRSRV2 machine.

You can have one or many Control Center machines on the same node. You can have the same Control Center machine name on different nodes, and one Control Center service machine can support one or more databases.

Important:

If the same database machine name exists on two different nodes, you *must* always use the nickname to direct communication to the correct database. In the previous example, if a Control Center command was issued to SQLMACH, the results would be unpredictable. If you had previously been communicating with the database on WMAVM1, then Control Center would direct your new command to the same node. If you were previously communicating with a third (different) database and directed a command to SQLMACH, the command cannot be directed to your desired database. The best policy is to always qualify the database name with the node ID or use the established unique nickname.

The DBINIT CONTROL file is used to link and access the appropriate 195 minidisk when using a Database Administration tool.

This file can be manually updated using Xedit or automatically updated using appropriate options (either **New Database Setup** or **Database Parameters**).

Each database is described in terms of:

- Database name
- Database server production disk name or SFS directory names
- Address of minidisk with database server production code, or value "SFS", if SFS is used
- User's link address (disk address at user machine) (optional)
- User's link filemode (user's access filemode) (optional)
- RXSQL Code Disk owner, or SFS directory names if SFS is used (optional)
- Address of minidisk with RXSQL production code, or 'SFS' if SFS is used (optional)
- Link address for RXSQL (disk address at user machine) (optional)
- Link filemode for RXSQL (user's access filemode) (optional)
- Optional nickname (VM MACHINE NAME or ID)

This file will be updated when a new database is identified to the product. If the machine on which the database is identified owns the code disk, the file will be replaced on the code disk. Otherwise, the updated (merged) file is sent back to the user who defined the database with instructions to send the file to the service machine which owns the code disk, so that the file can be properly replaced. This must be done by a Control Center Administrator. The file will require manual updating by the DBA to include RXSQL product disk information if desired for user availability. This file should be made available to all users who will be executing the Database Administration tools of Control Center.

```
SQLDBA1  SQLDBA1  195 5FF E RXSQL22 198 * *
SQLDBA2  SQLDBA2  195 5FF E RXSQL22 198 * *
SQLDBA3  SQLMACH3 195 *   * RXSQL31 198 * * SQLMACH3
```

Figure 23. Sample DBINIT CONTROL File

```
DB261VM  SFSPOOL1:DB261VM.SQL.PRODUCTION SFS * * * * * * *
SQLDBA   SQLMACH  195 * * * * * * SQLMACH
DB2PROD  DB2PROD  195 * * SFSPOOL1:DB261VM.SQL.RXSQL SFS * * *
```

Figure 24. Sample DBINIT CONTROL File Using SFS

With this information, the Database Administration tools of the product will be able to automatically link and access the correct database server production disks and RXSQL code disks (if available) for each database. This is especially important for certain functions, such as migrating data between databases. The user ID must also have authority to LINK and ACCESS the indicated disks in READ mode without supplying a READ password. This must be attended to according to your installation authorization procedures.

Setup Step 15: Log Off the MSTROWN Machine

Control files setup is now complete. Log off the MSTROWN machine.

Database Machine Setup

The next set of steps set up the database machine to work with a Control Center service machine.

Setup Step 16: Log On to the Database Machine (SQLMACH)

Log on to the database machine to complete the installation of the new database under Control Center control.

Setup Step 17: Link and Access the Control Center Code Disk

Access the Control Center code disk used by the managing service machine with an available disk access mode (D). If you did not include a LINK statement in the VM directory (Setup Step 2: Redefine the Database Machine (SQLMACH)) you will first need to link to the code disk with an available CUU (547).

Note: SQLMACH must be authorized to access the Control Center code disk in READ mode.

Setup Step 18: Copy and Update DBPROF EXEC

Use these commands to make a backup copy of the existing database machine PROFILE EXEC and to replace it with the DBPROF EXEC from the code disk (547 D).

```
COPYFILE PROFILE EXEC A PROFILE OEXEC A (REP
COPYFILE DBPROF EXEC D PROFILE EXEC A (REP
```

The PROFILE EXEC file may need to be modified for any LINK and ACCESS statements required at your installation.

Important:

If you did not include a LINK statement in the VM directory (“Setup Step 2: Redefine the Database Machine (SQLMACH)” on page 63) for the Control Center code disk *or* use a device address (CUU) other than 547, you will need to modify the PROFILE EXEC.

A partial listing of the PROFILE EXEC is shown in Figure 25.

- The ACCESS statement for the Control Center code disk is indicated by an **A** .
- If required, a LINK to the Control Center code disk should be inserted immediately before the ACCESS statement.

```
/* DBPROF  EXEC  */
/*   5648-A70 (C) COPYRIGHT IBM CORP. 1990, 1998   */
/*   Licensed material - Program Property of IBM   */
trace 'o'
address COMMAND

'SET AUTOREAD OFF'

 /*** try default access of SQLMSTR 195 disk for directory links ***/
 accopts = Read_globalv('SQLMSTR' 'Disk_access_options')
 if accopts > ' ' then accopts = '(' || accopts

 'SET CMSTYPE HT'
A address CMS 'ACCESS 547 D' accopts
 'SET CMSTYPE RT'

/*******************************************/
/*  Check to see if a terminal is connected */
/*******************************************/
termaddr = '
 'EXECIO * CP (STRING QUERY USERS' userid()
                  .
                  .
                  .
```

Figure 25. Partial Listing of the Database Machine PROFILE EXEC (Renamed from DBPROF EXEC)

Two database disks will be linked by the database machine during normal operation: the database production disk (usually referred to as the 195 disk) and the database service disk (usually referred to as the 193 disk). During most database operations, both of these disks are linked in READ mode. During certain database single user mode operations (coldlogs, adding dbextents, adding dbspaces), the 195 disk will be linked in WRITE mode. These disks are linked by the DBLINK EXEC which is supplied with Control Center and are linked as the 195 (Q filemode) and 193 (V filemode) unless otherwise specified in the database parameters file. It is the responsibility of the installer to handle any DIRMAINT or access requirements of the installation environment which will allow the DBLINK exec to switch between read and write links to these disks. This may require a modification to the DBLINK EXEC file.

Other Disk Links

If the database machine needs access to other local disks, the PROFILE EXEC should be modified to add the required LINK and ACCESS statements. When the DYNAM/T , EPIC or VMTAPE product is used, a link may be required to provide the database with access to code required for tape operation. For example, the DMSTVI module must be accessed when VMTAPE or EPIC is used. The DMSTVI module is required to support multivolume archive tapes.

Setup Step 19: Run DBSTART EXEC

If all of the previous steps were completed successfully, then the database can be started using the database startup parameters defined earlier by entering this command at the CMS command prompt:

```
DBSTART
```

The DBSTART process will disconnect the database machine from your console and will start the database initiation process under control of Control Center.

Setup Verification

The following steps will determine whether the database machine and the managing service machine can communicate.

Setup Step 20: Log On to the MSTROWN Machine

Log on to the MSTROWN machine or any machine that has Control Center Administrator authority on the service machine specified on the CONSOLE statement in Setup Step 2; (the managing Control Center service machine).

Setup Step 21: Verify Messages Received

If, during database startup processing, you are logged on to a machine with Control Center Administrator authority, you should receive these messages:

```
11:41:43  * MSG FROM MSTRSRV: SQMA0025I SQLMACH=> Database .... testing SCIF interface
11:41:50  * MSG FROM MSTRSRV: SQMA0025I SQLMACH=> Database .... is starting
11:42:36  * MSG FROM MSTRSRV: SQMA0045I SQLMACH=> database has been started, logmode
```

Note: You *must* be logged on at the time database startup processing on SQLMACH issues the messages for it to appear on your console. Continue with the verification steps even if you do not receive a message.

Setup Step 22: Link and Access the Control Center Code Disk

Link and access the Control Center code disk in READ mode with an available virtual device address (CUU) and access mode (547 D).

Notes:

1. Depending on your environment, the code disk may already be linked and accessed.
2. Depending on your environment, the code disk may be owned by a machine other than the managing service machine; (the managing service machine is sharing a code disk).

Setup Step 23: Review the Database Startup File

Enter these commands at the CMS command prompt to request a copy of the database SQLSTART file:

```
SQM (MSTRSRV
SQM CMS SENDFILE SQLMACH SQLSTART TO MSTROWN
```

Replace `MSTRSRV`, `SQLMACH` and `MSTROWN` with appropriate machine IDs.

MSTRSRV will send the SQLSTART file (SQLMACH SQLSTART) to your virtual reader where it can be viewed using the CMS PEEK command. Review this file to verify successful database startup.

Shown in Figure 26 on page 82 is an example database SQLSTART file.

```
ARI0025I Database testing SCIF interface...
ARI0025I TVITEST: DMSTVS MODULE available
"ACCESS 547 D (NOSAVE" performed, RC=0
ARI0025I DB2
VERSION 6 RELEASE 1 MOD 0
File SQLMACH DBEXTMAP A1 sent to MSTRSRV at VMSYSM1 on 05/25/97 14:24:06
14:24:08  * MSG FROM MSTRSRV : SQMRDR: SQLMACH DBEXTMAP A received
by MSTRSRV at VMSYSM1
DMSLBD441R Enter VOLID information:
DMSLBD441R Enter VOLID information:
DMSLBD441R Enter VOLID information:
DBM0001I OVERRIDE Startup Parameters Request
ARI0717I Start SQLSTART EXEC: 05/25/97 14:24:29 EDT.
ARI0663I FILEDEFS in effect are:
Z        DISK     DMSNAM   LOADLIB  *
ARIARCH  TAP1  SL  00001  VOLID QU1412
ARISQLLD DISK     ARISQLLD LOADLIB  Q1
BDISK    DISK     300
LOGDSK1  DISK     201
LOGDSK2  DISK     202
DDSK1    DISK     200
DDSK2    DISK     204
DDSK3    DISK     220
DDSK4    DISK     221
DDSK5    DISK     222
DDSK7    DISK     224
DDSK9    DISK     240
DDSK12   DISK     243
DDSK13   DISK     244
DDSK14   DISK     245
DDSK15   DISK     225
DDSK10   DISK     241
ARIUSRDD DISK     USERLIB  LOADLIB  *
ARITRAC  TAP2  SL  00001
ARILARC  TAP3  SL  00001
ARI0025I The program ARISQLDS is loaded at 6F8000.
ARI0025I The program ARIXRDS is loaded at 59F000.
ARI0025I The program ARIXSXR is loaded at 932000.
ARI0025I The program ARICMOD is loaded at 988000.
ARI0012I The database manager ignores ARCHPCT = 79
         when LOGMODE = Y or N.
ARI0015I ACCOUNT parameter value is N.
ARI0015I DUMPTYPE parameter value is P.
ARI0015I LOGMODE parameter value is Y.
ARI0015I STARTUP parameter value is W.
ARI0015I SYSMODE parameter value is M.
ARI0015I EXTEND parameter value is N.
ARI0015I CHARNAME parameter value is ENGLISH.
ARI0015I DBNAME parameter value is SQLMACH.
ARI0015I PARMID parameter value is SQLMACH.
ARI0015I TRACDSC parameter value is 00.
ARI0015I TRACRDS parameter value is 0000000.
ARI0015I TRACDBSS parameter value is 00000000000.
ARI0016I CHKINTVL parameter value is 10.
ARI0016I NCSCANS parameter value is 30.
ARI0016I NCUSERS parameter value is 6.
ARI0016I NDIRBUF parameter value is 1000.
ARI0016I NLRBS parameter value is 3022.
ARI0016I NLRBU parameter value is 1000.
ARI0016I NPAGBUF parameter value is 400.
ARI0016I SLOGCUSH parameter value is 90.
ARI0016I SOSLEVEL parameter value is 10.
ARI0016I DISPBIAS parameter value is 7.
ARI0283I Log analysis is complete.
ARI0282I LUW UNDO is completed.
ARI0281I LUW REDO is completed.
ARI0060I DB2 initialization complete.
ARI0045I Ready for operator communications.
```

Figure 26. Example Database SQLSTART File

Chapter 7. Migration

This chapter describes how to migrate an existing Control Center V5.1 or SQL Master environment (either Version 1 Release 2.0 or Version 1 Release 1.2) to Control Center Version 6 Release 1.0.

To migrate from earlier levels of SQL Master (Version 1 Release 1.0 or Version 1 Release 1.1), you must first migrate to SQL Master Version 1.2. Follow the steps as you are prompted by migration exec MIGRV1R2. You will be instructed to provide the correct values for your environment. After you have successfully migrated to Version 1.2, it will be necessary for you to then execute MIGRV6R1 to complete the migration to Control Center Version 6 Release 1.

Checklist for Migrating from SQL Master

Notes:

- *Perform the steps in order.*
- *Optional steps are preceded by a circle (○).*
- *Page references appear in parentheses.*

Code Migration

__ 1. Log On to the Service/Support Machine (84)

__ 2. Link and Access Code Disk (84)

__ 3. Make Backup Files (84)

__ 4. Install Control Center Code (84)

__ 5. Migrate to Control Center (85)

Post Code Migration

__ 6. Apply Local Changes (86)

__ 7. Start Control Center Service/Support Machine (86)

Service/Support Machine Migration Verification

__ 8. ○ Log On to the MSTROWN Machine (86)

__ 9. ○ Link and Access the Control Center Code Disk (86)

__ 10. ○ Verify Service/Support Machine Operation (86)

Important:

This completes the migration process for a Control Center support machine. If you are not migrating a Control Center service machine, omit all of the steps that follow.

[Service Machine Only] Database Machine Migration

__ 11. Log On to the Database Machine (SQLMACH) (87)

__ 12. Link and Access the Control Center Code Disk (87)

__ 13. Copy and Update DBPROF EXEC (87)

__ 14. Run DBSTART EXEC (88)

[Service Machine Only] Database Migration Verification

__ 15. ○ Log On to the MSTROWN Machine (88)

__ 16. ○ Verify Messages Received (88)

__ 17. ○ Link and Access the Control Center Code Disk (88)

__ 18. ○ Review the Database Startup File (89)

YEAR 2000 Changes for SQLMAINT Users

__ 19. ○ Convert SQLMAINT.SQLMAINT_TABLE (89)

Data Restore Machine Migration

__ 20. Rename the XTS9X001 Exec (90)

Code Migration

The following steps of the migration process load the Control Center Version 6 Release 1.0 code and preserve any modified Control Center control files.

Migration Step 1: Log On to the Service/Support Machine

To begin the Control Center migration, log on to the service machine.

Migration Step 2: Link and Access Code Disk

Link and access the Control Center code disk with an available virtual device address (cuu) and access mode.

Note: Depending on the machine being migrated and/or your installation configuration, the code disk may already be linked or linked and accessed.

Migration Step 3: Make Backup Files

This step applies to the Control Center service machine. Use the CMS COPYFILE command to back up and rename these files from the code disk to the A-disk:

- DATABASE MESSAGES
- SQLMSTR DIRECTRY
- DBINIT CONTROL

For example, enter the following at the CMS command prompt to back up the DATABASE MESSAGES file:

```
COPYFILE DATABASE MESSAGES D DATABASE OMESSAGE A
```

Depending on your installation configuration, files SQLMSTR DIRECTRY and DBINIT CONTROL might not exist on the code disk. If this is the case, you do not need to back up these files. Note that these files will NOT be overlaid by the new code - backing them up is only a cautionary measure in the event they are needed in case of any unforeseen problem with the install.

Important:

Control Center modules on the code disk will be overwritten. You should back up all files that have been locally customized to the A-disk.

Migration Step 4: Install Control Center Code

This step applies to the Control Center service machine. The Control Center code must be installed on the MSTRSRVO 195 code disk from the product distribution tape. This can be done from the MSTRSRVO machine by accessing the code disk in WRITE mode, mounting the product tape, and using VMFPLC2 to copy the code from tape to disk, as described in the procedure:

1. Link and access the MSTRSRVO 195 code disk in WRITE mode; for example, as filemode K.

2. Mount the Control Center distribution tape on cuu 181 without a ring, that is, in a write-protected manner. The actual mount command will vary from site to site depending on the tape handling software installed.
3. Rewind the tape.

   VMFPLC2 REW
4. Load the files onto the accessed MSTRSRVO 195 code disk.

   VMFPLC2 LOAD * * K

   Note: The file SQLMSTR VERSION, which is included in the LOAD, will contain the version number of the Control Center software.
5. Detach the tape.

Migration Step 5: Migrate to Control Center

Use the MIGRV6R1 exec supplied with Control Center to modify the SQLMSTR CONTROL file to add the new Date-Time-Format (if not there); and to update the database parms files with new database and Data Restore parameters. Enter at the CMS command prompt line.

```
MIGRV6R1
```

The exec will prompt you for:

1. The current level of code. If you are at SQL Master Version 1.1.2, you will be prompted to supply the Date-Time-Format desired (USA, EUR, ISO, JIS). If you are at SQL Master level 1.2.0 **AND** do not have PTF UV59853 installed for Year 2000 readiness, you will be prompted for the same information. The current SQLMSTR CONTROL file will be backed up on your A-disk with the name SQLMSTR OCONTROL and be updated with the new Date-Time field parameter.
2. The exec will then backup all the existing database parms files on your A-disk, if you are a Service Machine (Support Machines do not have such files) with the new file type of OPARMS.
3. After this is completed, you will receive a **Migration process has completed** message. If errors are encountered, you will be informed, and you must re-run the exec.
4. **IMPORTANT:** Database ARCHHIST files from older versions of Control Center are NOT compatible with Control Center 6.1.0 After MIGRV6R1 has executed, if you move control of other databases from older Control Center machines to a Control Center 6.1.0 machine, you must migrate the ARCHHIST file of that database by executing: **SQMMIGHS dbmach** on the CMS command line of the Control Center machine. Dbmach must be the userid of the database and the dbmach ARCHHIST and BACKHIST file must be on the Control Center service machine A-disk. Database TAPES files from older versions of Control Center will not contain Data Restore tapes. Executing: **SQMMIGTP dbmach** on the CMS command line of the Control Center machine will merge the DRTAPES file entries into the database TAPES file. Do **not** execute the MIGRV76R1 exec again to reformat individual ARCHHIST and TAPES files.

Post-Service/Support Machine Migration

Migration Step 6: Apply Local Changes

The SQLMSTR $PROFILE file contains entries which are not on your current SQLMSTR PROFILE file. This file should be compared to your SQLMSTR PROFILE file and any updates should be made.

Migration Step 7: Start Control Center Service/Support Machine

This step will initialize and start the Control Center service machine.

Enter these commands at the CMS command prompt:

```
#CP IPL CMS
startexec
```

Replace `startexec` with `SQLMSTR` if migrating a service machine. Replace it with `SQLMSUP` if migrating a support machine.

The service/support machine immediately disconnects from your active terminal session and the startup process completes in disconnect mode. You will be logged off the terminal at this point. The migration process is complete.

Service/Support Machine Migration Verification

These steps will determine whether Control Center is functioning and can communicate with users:

Migration Step 8: Log On to the MSTROWN Machine

Log on to the MSTROWN machine or any machine that has Control Center Administrator authority on the managing service machine.

Migration Step 9: Link and Access the Control Center Code Disk

Link and access the Control Center code disk in READ mode with an available virtual device address (cuu) and access mode (195 D).

Migration Step 10: Verify Service/Support Machine Operation

To establish communications with the service/support machine, enter this command at the CMS command prompt:

```
SQM (machine
```

Replace `machine` with the machine ID of your service/support machine.

Verify the service/support machine is operating by entering,

```
SQM VERSION
```

Control Center will return a message identifying the machine and the Control Center product release and modification level.

You *must* be logged on to the MSTROWN or another machine with Control Center administrator authority for these messages to appear on your console.

If this is successful, Control Center has been successfully installed and the verification process is complete.

[Service Machine Only] Database Machine Migration

The next set of steps migrates a database machine to work with the service machine. They should be completed for each database machine managed by the service machine.

Migration Step 11: Log On to the Database Machine (SQLMACH)

Log on to the database machine.

Migration Step 12: Link and Access the Control Center Code Disk

Link and access the Control Center code disk used by the managing service machine with an available virtual device address (cuu) and disk access mode (547 D). Depending on your environment, the code disk may already be linked.

Note: SQLMACH must be authorized to access the code disk in READ mode.

Migration Step 13: Copy and Update DBPROF EXEC

Use these commands to make a backup copy of the existing database machine PROFILE EXEC and to replace it with the DBPROF EXEC from the code disk (547 D):

```
COPYFILE PROFILE EXEC A PROFILE EXECO A (REP
COPYFILE DBPROF EXEC D PROFILE EXEC A (REP
```

The PROFILE EXEC file may need to be modified for any LINK and ACCESS statements required at your installation.

Important:

If the database machine does not link to the Control Center code disk with an entry in the VM directory *or* you used a device address (cuu) other than 547, you will need to modify the PROFILE EXEC.

A partial listing of the PROFILE EXEC is shown in Figure 25 on page 79.

- The ACCESS statement for the Control Center code disk is indicated by an **A**.
- If required, a LINK to the Control Center code disk should be inserted immediately before the ACCESS statement.

Two database disks will be linked by the database machine during normal operation: the database production disk (usually referred to as the 195 disk) and the database service disk (usually referred to as the 193 disk). During most database operations, both of these disks are linked in READ mode. During certain database single user mode operations (coldlogs, adding dbextents, adding dbspaces), the 195 disk will be linked in Multiple-write mode. These disks are linked by the DBLINK EXEC which is supplied with Control Center and are linked as the 195 (filemode Q) and 193 (filemode V) unless otherwise specified in the database parameters file. It is the responsibility of the installer to handle any DIRMAINT or access requirements of the installation environment which will allow

the DBLINK exec to switch between read and write links to these disks. This may require a modification to the DBLINK EXEC file.

Other Disk Links

If the database machine needs access to other local disks, the PROFILE EXEC should be modified to add the required LINK and ACCESS statements. When the DYNAM/T , EPIC or VMTAPE product is used, a link may be required to provide the database with access to the code required for tape operation. For example, the DMSTVI module must be accessed when VMTAPE or EPIC is used. The DMSTVI module is required to support multivolume archive tapes.

Migration Step 14: Run DBSTART EXEC

If all of the previous steps were completed successfully, then the database can be started by entering this command at the CMS command prompt:

```
DBSTART
```

The DBSTART process will disconnect the database machine from your console and will start the database initiation process under control of Control Center.

[Service Machine Only] Database Migration Verification

The following steps will determine whether the database machine and the managing service machine can communicate. They should be completed for each database machine managed by the service machine.

Migration Step 15: Log On to the MSTROWN Machine

Log on to the MSTROWN machine or any machine that has Control Center Administrator authority on the managing service machine.

Migration Step 16: Verify Messages Received

If, during database startup processing, you are logged on to a machine with Control Center Administrator authority, you should receive these messages:

```
11:41:43  * MSG FROM MSTRSRV: SQMA0025I SQLMACH=> Database .... testing SCIF interface
11:41:50  * MSG FROM MSTRSRV: SQMA0025I SQLMACH=> Database .... is starting
11:42:36  * MSG FROM MSTRSRV: SQMA0045I SQLMACH=> database has been started, logmode
```

Note: You **must** be logged on to an userid with Control Center Administrator authority *during* database start up processing in order for these messages to appear on your console. Continue with the verification steps even if you do not receive a message.

If, during database startup processing, you are logged on to a machine with Control Center Administrator authority, you should receive these messages:

Migration Step 17: Link and Access the Control Center Code Disk

Link and access the Control Center code disk in READ mode with an available virtual device address (cuu) and access mode (547 D).

Notes:

1. Depending on your environment, the code disk may already be linked and accessed.
2. Depending on your environment, the code disk may be owned by a machine other than the managing service machine; (the managing service machine is sharing a code disk).

Migration Step 18: Review the Database Startup File

Enter these commands at the CMS command prompt to request a copy of the database SQLSTART file:

```
SQM (MSTRSRV
SQM CMS SENDFILE SQLMACH SQLSTART TO MSTROWN
```

Replace `MSTRSRV`, `SQLMACH` and `MSTROWN` with appropriate machine IDs.

MSTRSRV will send the SQLSTART file (SQLMACH SQLSTART) to your virtual reader where it can be viewed using the CMS PEEK command. Review this file to verify successful database startup.

Shown in Figure 26 on page 82 is an example database SQLSTART file.

Migration Step 19: Convert SQLMAINT.SQLMAINT_TABLE

Note: If installing the SQLMAINT function for the first time, or if you are migrating Control Center Version 5.1, the instructions in this step are not necessary.

If you are migrating from SQL Master 1.1.2 or 1.2 and use SQLMAINT, and PTF UV59853 has **NOT** been installed, the steps below need to be taken to convert the SQLMAINT.SQLMAINT_TABLE to Year 2000 format. A conversion exec (YR2000MC) has been supplied to aid in the conversion of the SQLMAINT.SQLMAINT_TABLE. The SQLMAINT_TABLE has 2 date fields which are in Julian format. These fields will be converted to an "extended Julian" format, a 4-digit year followed by the 3-digit day number). The exec provided unloads the data, drops and recreates the table, and reloads the table with the new extended date fields.

There are 2 DBSU input files used by the exec which are also supplied (YR2000 UNLDBSU and YR2000 LOADBSU).

1. Access the Control Center code disk.
2. From a DBA ID, run the SQLTABLE tool to capture the DDL for the SQLMAINT_TABLE.
   a. Use Option **3** with PAUSE=YES and REBIND PACKAGES=NO option.
   b. Unload to the A-disk or some other permanent disk.
   c. Save the unloaded A4 file (SQLMAIN$ SQLTABLE A4) as a backup.
   d. Save the SQLMAIN$ EXEC file if a recovery is needed.
   e. Discard the SQLMAIN$ LOADDBSU file.
   f. Save the DDLDBSU file.
   g. Move the 2 new supplied DBSU files (YR2000 UNLDBSU and YR2000 LOADBSU) from the Control Center code disk to the DBA's A-disk (or the permanent disk used for unloading the table).
3. Modify the YR2000 LOADBSU file, supply the DBSPACE name for the CREATE Table IN statement.

4. Run the YR2000MC exec.
5. If you have packages which were invalidated, do the following:
   a. Execute SQMUTIL from the CMS ready screen. This will place you in the Object Search List Utility. Press **PF10** for the Table Search panel. Fill in the fields for the SQLMAINT_TABLE and Press Enter.
   b. You will be presented with the Table List panel.
   c. Enter Option "**P**" for the SQLMAINT_TABLE to get a list of packages on the table.
   d. From the Package List screen, use the "**R**" option to rebind or the "**U**" and "**L**" options to unload and reload the packages.

Migration Step 20: Rename the XTS9X001 Exec

If you are using Control Center to manage your Data Restore functions then you must rename the XTS9X001 EXEC provided by Data Restore. This step must be performed for each Data Restore machine after you have installed/migrated your Data Restore code and should be checked again each time Data Restore service is applied.

The XTS9X001 EXEC is a template provided by Data Restore to allow users to change log Labeldefs during Reload Recovery. Control Center provides this EXEC for you on its code disk. You must rename the XTS9X001 EXEC on the Data Restore code disk so that the Control Center code will execute.

Locate the XTS9X001 EXEC on the Data Restore code disk and execute this CMS command:

```
RENAME XTS9X001 EXEC fm = EXECDRF =
```

where "fm" is the filemode of the Data Restore disk.

Chapter 8. Corrective Service

IBM provides corrective service for the Control Center product. Corrective service is provided in PTFs.

All service for Control Center is provided as directly replaceable files, with no additional compilation or link-edit steps required. However, any locally customized files will need to be considered when new versions of those files are received on the Control Center service tape.

Installing Corrective Service

To apply service to Control Center from a PTF tape, follow the instructions provided with the tape to copy the contents of the tape to a temporary disk, rename the file types, and then copy the files to the Control Center code disk. After applying service, the database should be restarted so that the new code will be re-accessed.

After the Control Center code disk has been updated, it may be necessary to update certain files on each service machine's A-disk (191 minidisk) by manually using an editor (such as XEDIT). These additional updates will only be required if the applied Control Center service includes any of the special files that are customized for local use and copied to the service machine's 191 disk. These files will have a special filetype that begins with the $ character. Files provided with Control Center which may be customized include:

- DBINIT $CONTROL
- SQLMSTR $CONTROL
- SQLMSTR $DIRECTR
- SQLMSTR $HOLIDAY
- SQLMSTR $PROFILE
- SQLMSTR $TIMES
- SQMOUNT $EXEC
- SQMQMF $EXEC
- SQMSTAPE $EXEC
- UARCCAN $EXEC
- UARCSUB $EXEC
- SPBANLZ $EXEC
- SPBSTART $EXEC
- SPBEND $EXEC

Additional information for some of the files listed above can be found in Chapter 11, “Managing the Environment” on page 113.

It may also be necessary to modify the PROFILE EXEC on the service machine, support machines, and database machines.

Service Machine

The SQMPROF $EXEC provided with Control Center is renamed to PROFILE EXEC and placed on the service machine's A-disk during Control Center service machine installation. If a new SQMPROF $EXEC is provided on the PTF tape, it should be used to replace the existing PROFILE EXEC. Make changes for local use as required.

Support Machines

The SUPPROF $EXEC provided with Control Center is renamed to PROFILE EXEC and placed on the support machine's A-disk during Control Center support machine installation. If a new SUPPROF $EXEC is provided on the PTF tape, it should be used to replace the existing PROFILE EXEC. Make changes for local use as required.

Database Machines

The DBPROF $EXEC provided with Control Center is used as the PROFILE EXEC for each database machine. If a new DBPROF $EXEC is provided on the PTF tape, it should be used to replace the existing PROFILE EXEC on each database virtual machine. Make changes for local use as required.

Chapter 9. Data Restore Setup with Control Center

This chapter describes how to setup a Data Restore machine to be managed by Control Center.

Prior to performing the Data Restore setup process detailed in this chapter you should have:

- Generated a DB2 Server for VM database,
- Installed a Control Center service machine,
- Set up a DB2 Server for VM database to work with Control Center (refer to Chapter 4, “Installing the Service Machine” on page 37 of this manual), and
- Installed Data Restore.

Data Restore Considerations

Control Center helps you manage the operational functions of databases in different ways, adapting to the needs of each specific database. You should determine your recovery strategy and decide how Control Center, together with Data Restore, can best meet your needs. Decisions should be made such as:

- if Data Restore BACKUP, or a TRANSLATED archive (SQLEND ARCHIVE or SQLEND UARCHIVE) will be used;
- whether BACKUP, TRANSLATE, and UNLOAD will use predefined tapes, scratch tapes or disk files; and
- how often these activities will be executed.

Checklist for Setting up the Data Restore Interface

Notes:

- *Perform the steps in order.*
- *Optional steps are preceded by a circle (○).*
- *Page references appear in parentheses.*

Preliminary Setup

__ 1. Log On to the VM MAINT Machine (95)

__ 2. Redefine the Data Restore Machine (DRMACH) (95)

__ 3. Update the VM Directory (96)

__ 4. Add Data Restore Machine to Autolog List (96)

Control Files Setup

__ 5. Log On to the MSTROWN Machine (97)

__ 6. Link and Access the Control Center Code Disk (97)

__ 7. Define Data Restore Parameters File (97)

__ 8. Create the SDRESTOR DIRECTRY (98)

__ 9. Migrate SDRESTOR DIRECTRY File (99)

__ 10. Start Control Center Session (99)

__ 11. Update Database Parameters File (103)

__ 12. Define & Update Database TAPES Files (103)

__ 13. Update the Control Center SQLMSTR Profile (103)

__ 14. Create the LINKPWDS File (103)

__ 15. Restart Control Center (103)

Data Restore Machine Setup

__ 16. Log On to the Data Restore Machine (DRMACH) (104)

__ 17. Link and Access the Control Center Code Disk (104)

__ 18. Create the Data Restore SQLMSTR Profile (104)

__ 19. Copy and Update DRFPROF EXEC (104)

__ 20. Rename XTS9X001 EXEC (105)

__ 21. Run DRMSTR EXEC (105)

Setup Verification

__ 22. ○ Send the Data Restore Machine (DRMACH) a Message (105)

Preliminary Setup

Steps 1 through 4 prepare Control Center to interface with your Data Restore machine. Figure 27 summarizes what you will accomplish.

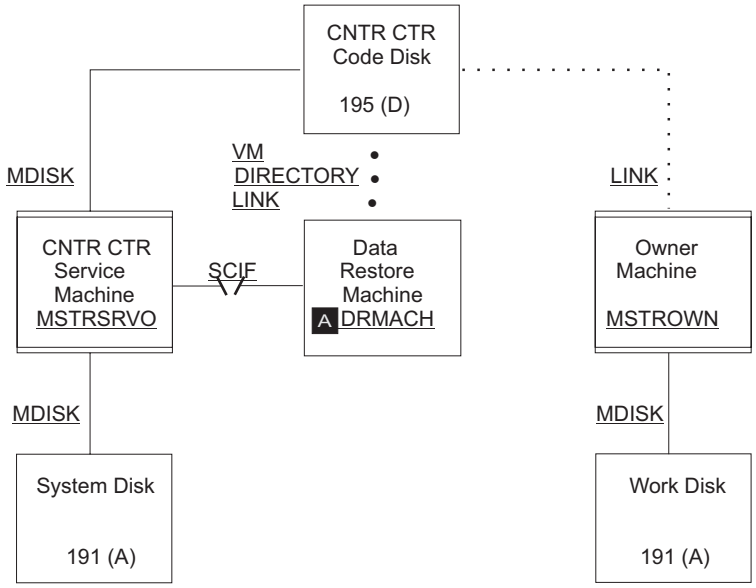


Figure 27. Data Restore machine in a Control Center Environment. Data Restore machine DRMACH **A** has been added to the environment shown in Figure 27 using the Data Restore machine setup process. Service machine MSTRSRVO manages DRMACH and is the owner of the Control Center code disk. Communications between DRMACH and MSTRSRVO is through the SCIF interface.

Setup Step 1: Log On to the VM MAINT Machine

To begin the Data Restore machine setup, log on to the MAINT virtual machine.

Setup Step 2: Redefine the Data Restore Machine (DRMACH)

Change the VM directory entries for the Data Restore machine to activate the Single Console Interface Facility (SCIF) and to link the Control Center code disk. Refer to the appropriate VM/ESA Operation manuals for a complete description of these statements.

```
         USER DRMACH XXXXXXXX 4M 6M G
           ACCOUNT nnnnnnnn
           OPTION ACCT
           IUCV ALLOW
           IPL CMS
1 ───►     CONSOLE    009         3215    T       MSTRSRV0
           SPOOL      00C         2540            READER    *
           SPOOL      00D         2540            PUNCH     A
           SPOOL      00E         1403            A
           LINK       MAINT       190     190     RR
           LINK       MAINT       19D     19D     RR
           LINK       MAINT       19E     19E     RR
2 ───►     LINK       MSTRSV0     195     495     RR
         MDISK 191 3380 cylr 080 volser MR
         MDISK 193 3380 cylr 043 volser MR
         MDISK 195 3380 cylr 015 volser MR
         MDISK D01 3380 cylr 075 volser MR
         *          TRANSDSK  DISKs
         MDISK D02 3380 cylr 075 volser MR
         MDISK D03 3380 cylr 075 volser MR
         MDISK D04 3380 cylr 075 volser MR
         *          UNLOAD DISKs
         MDISK D05 3380 cylr 100 volser MR
         MDISK D06 3380 cylr 200 volser MR
         MDISK D07 3380 cylr 100 volser MR
```

Figure 28. Example VM Directory Control Statements for a Data Restore machine

Note: The number and size of minidisks needed by the DRMACH will depend on the size of the tables, whether tapes or disks are used, the number of databases being managed, and the frequency of Data Restore operations. The DRMACH's 191 **must** be large enough to hold the output from a RELOAD with log recovery.

Statement 1: CONSOLE 009 3215 T MSTRSRVO

Defines the managing Control Center service machine as being the secondary console of the Data Restore machine. This is done by including **T MSTRSRVO** in the CONSOLE statement as shown.

Note: The managing service machine can either be the owner of the code disk or a service machine that shares a code disk.

Statement 2: LINK MSTRSRVO 195 495 RR

This statement links in READ mode to the same code disk linked to by the managing service machine. If you have a shared environment, the

code disk can be owned by a machine other than the managing service machine.

Instead of including the above statement in the VM directory, you can, during a later step, add a LINK statement to the PROFILE EXEC of the DRMACH machine.

DRMACH must be authorized to access the Control Center code disk in READ mode.

Important:

1. The IPL CMS statement should *not* include PARM AUTOCR. PARM AUTOCR causes an extra console interrupt to occur on the Data Restore machine when it is autologged. This results in problems with startup communications between the Data Restore machine and the Control Center service machine.
2. SYSPROF EXEC should be checked to avoid additional console interrupts from being stacked during Data Restore machine startup. If the SYSPROF EXEC causes a problem, the IPL statement in the database machine directory can be modified to IPL CMS PARM NOSPROF.
3. If minidisk passwords are used, please read Appendix C, “Password Support” on page 501 for more information.

Setup Step 3: Update the VM Directory

After you have made the VM directory statement changes for the Data Restore machine, update the VM directory using your local operating procedures.

Setup Step 4: Add Data Restore Machine to Autolog List

Under normal operations the DRMACH machine should be autologged each time the CPU is IPLed. The DRMACH machine should therefore be included in the list of virtual machines that are autologged each time the CPU is IPLed.

For proper startup of each Data Restore machine when the CPU is IPLed, the Data Restore machine *must* be autologged *after* the managing Control Center service machine.

Important:

1. The Control Center service machine *must* be autologged *prior* to the Data Restore machine for SCIF communication to be properly established. Additionally, portions of the Data Restore machine startup process require that the Control Center service machine be fully operational.
2. If problems occur with the Data Restore machine startup when the CPU is IPLed, then a CP SLEEP 1 MIN statement should be added near the beginning of the Data Restore machine's PROFILE EXEC. This will delay the startup of the Data Restore machine for 1 minute, giving the Control Center service machine a chance to become fully operational.

Preliminary Data Restore setup is now complete. Log off the MAINT machine.

Setup Step 5: Log On to the MSTROWN Machine

Log on to the MSTROWN machine or any machine that has Administrator authority on the Control Center service machine.

Setup Step 6: Link and Access the Control Center Code Disk

Link and access the Control Center code disk in READ mode with an available virtual device address (CUU) and access mode (195 D).

Notes:

1. Depending on your environment, the code disk may already be linked and accessed.
2. If you have a shared environment, the code disk can be owned by another machine.

Setup Step 7: Define Data Restore Parameters File

To define the Data Restore DRPARMS file, perform these tasks:

- Link to Control Center's code disk and copy the SDRESTOR $DRPARMS file to your 191 disk.
- Update using an editor (such as XEDIT).
- Rename the SDRESTOR $DRPARMS to *DRMACH* DRPARMS, where DRMACH is the name of your Data Restore machine.
- Migrate the *DRMACH* DRPARMS file to Control Center's 191 disk by using the CMS *sendfile* command.

```
*********************************************************************
*                          DREST1 DRPARMS                           *
*********************************************************************
:Nick.DREST61
 :Drmstr_Machine.Y
 :Version.
 :Administrators.DUKEWE
 :Operators.SQMDEMO3
 :Notify_error.DUKEWE
 :Notify_severe.DUKEWE
 :Notify_dbstatus.DUKEWE
 :Notify_operator.
 :Drtape_premount_timeout.60
```

Figure 29. Example DREST1 DRPARMS File

Parameter Field Name	Description
Nick	Machine ID of the Data Restore machine.
Drmstr_Machine	(Y or N): "Y" (YES) indicates that this machine is enabled. If "N" is specified, messages for this machine will be ignored by Control Center.
Version	Current version, release and modification level of Data Restore.

Administrators	Users who are authorized administrators on the Data Restore machine.
Operators	Users who are allowed to execute operator commands against the Data Restore machine.
Notify-error	Users who are to be notified when an error is received from the Data Restore machine.
Notify-severe	Users who are to be notified when a severe error is received from Data Restore.
Notify-dbstatus	Users who are to be notified whenever an alert or status message is sent about the Data Restore machine.
Notify-operator	VM operator who should be notified when serious problems occur on the Data Restore machine.
Drtape_premount_timeout	Indicates how many minutes Control Center will wait for a tape to be mounted after issuing a mount request before cancelling the mount request and Data Restore function. The default value is 60 minutes. When Drtape-premount = Y in the database PARMS file, Control Center will issue a tape mount request for the first tape of a single or multi-tape series, and wait until the tape is mounted before executing the Data Restore function. Control Center will also premount tapes when using native CMS tape handling or when the DMSTVI tape exit is not present. When the tape mount request exceeds the number of minutes indicated by Drtape_premount_timeout, the tape mount and function will be cancelled.

Setup Step 8: Create the SDRESTOR DIRECTRY

The SDRESTOR DIRECTRY file is used to show which database and Control Center service machine work with which Data Restore machine.

In brief, the steps for manually updating the SDRESTOR DIRECTRY are:

1. Link to the Control Center service machine's 195 disk and copy the SDRESTOR $DIRECTR to your '191' or A-disk.
2. Update using an editor (such as XEDIT).
3. Rename the file to SDRESTOR DIRECTRY.

```
*DBNICK   DBMACH    DBNAME    SQMMACH   NODE       DRMACH
DB2VM510  DB2VM510  DB2VM510  MSTRSRVO  MSTRSRV1   DREST51
DB2VM51A  DB2VM51A  DB2VM51A  MSTRSRVO  MSTRSRV1   DREST51
```

Figure 30. Example SDRESTOR Directory File

Table 6. SDRESTOR DIRECTRY File Format

Word	Description
1	Database machine name or nickname
2	Database machine name (virtual machine name)
3	Database name
4	Control Center service machine name
5	Control Center service machine node
6	Data Restore machine name

Figure 30 on page 98 contains entries for two Data Restore machines on the same node, both managed by Control Center service machine, *MSTRSRVO*.

Important:

If the same database name exists on two different nodes, you *must* always use the nickname to direct communication to the correct Data Restore machine.

Setup Step 9: Migrate SDRESTOR DIRECTRY File

You must have Control Center Administrator authority to migrate a file to the 195 disk owned by Control Center. Use the CMS *sendfile* command to transfer the SDRESTOR DIRECTRY to the code disk.

Setup Step 10: Start Control Center Session

To establish communications with MSTRSRVO, and to start Control Center in panel mode, enter these commands at the CMS prompt:

```
SQM (MSTRSRVO
SQM
```

Replace `MSTRSRVO` with the machine ID of your service machine. The Control Center Main Menu should now be displayed.

Database TAPES files

Terminology Used in the various tape file Sections:

1. The term *output media* refers to both *tape* (cartridge or reel-to-reel) and *disk* (CMS file) unless explicitly stated otherwise, as in "tape output medium".
2. The use of the term *tapes* when referencing the "tapes file" is not meant to exclude disk output medium.
3. The use of the term *tapes* in the Control Center panel interface is not meant, in most cases, to exclude disk output medium.

You must now update or define the various database TAPES files for Data Restore enablement. Refer to Chapter 15, “Tape Management Tool” on page 185 for a complete description of the Tape Management tool.

About the TAPES Files Required for Data Restore

Two separate database tapes files are maintained by Control Center for each database that will be enabled for Data Restore support. These files are kept on the Control Center service machine's 191 disk and are listed below.

TAPE NAME (fn ft)	USED BY
database TAPES	DATABASE ARCHIVE, LOG ARCHIVE, TRACE, and, Data Restore TRANSLATE, BACKUP, and INCREMENTAL BACKUP
database UNLTAPES	DATA RESTORE UNLOAD

Control Center uses the two *database* tapes files to maintain a list of output media assigned for usage during a database archive, log archive, trace activities and Data Restore functions.

If you have set up a database using Control Center, then the database TAPES file will already exist. The BACKUP, TRANS and TRANSDSK entries will be added to this file for every series. If Dual Backups will be performed, then BACKUP2 entries will be added. If Incremental Backup will be performed, then INCBK entries will be added. In “Setup Step 12: Define & Update Database TAPES Files” on page 103 which follows, new UNLTAPES files will be created.

Data Restore functions are executed on the Data Restore machine. All output media used for Data Restore functions (BACKUP, TRANSLATE, and UNLOAD) will be owned by the Data Restore machine. If output is to DISK, then the CUU indicated for the Data Restore functions must be the real address of the disk. The filemode value is used as a placeholder and will be indicated with an asterisk (*).

When a Data Restore function is started, Control Center will automatically link to the specified real address using the next free virtual address starting at CUU 120, and access the disk using the next free filemode available. When the function either completes or fails, Control Center will automatically detach any disks or tapes used during that operation. While in contrast, for database activities such as ARCHIVES, LOG ARCHIVES or TRACES, all disks used by the database must be linked and accessed beforehand.

It is the responsibility of the database administrator to ensure that the TAPES files are created correctly with valid tape volids and with series numbers that match those identified in the database parameters file. The *Series* field in the parameters file indicates which group of tapes within the TAPES file will be used for the next operation. Figure 31 on page 101 illustrates a sample *database* TAPES file. The Table 7 on page 102 explains the TAPES file format.

```
100  ARCHIVE  00000  00:00:00 UNUSED VOL001
100  BACKUP   00000  00:00:00 UNUSED VOL002
100  BACKUP2  00000  00:00:00 UNUSED VOL003
100  01INCBK  00000  00:00:00 UNUSED VOL004
100  01INCBK2 00000  00:00:00 UNUSED VOL005
100  02INCBK  00000  00:00:00 UNUSED VOL006
100  02INCBK2 00000  00:00:00 UNUSED VOL007
100  LOG      00000  00:00:00 UNUSED VOL008
100  LOG      00000  00:00:00 UNUSED VOL009
100  LOG      00000  00:00:00 UNUSED VOL010
100  TRANS    00000  00:00:00 UNUSED VOL011
100  TRANSDSK 00000  00:00:00 UNUSED SQLDB100 TRANSDSK * 196
200  ARCHIVE  00000  00:00:00 UNUSED VOL012
200  BACKUP   00000  00:00:00 UNUSED VOL013
200  BACKUP2  00000  00:00:00 UNUSED VOL014
200  01INCBK  00000  00:00:00 UNUSED VOL015
200  01INCBK2 00000  00:00:00 UNUSED VOL016
200  02INCBK  00000  00:00:00 UNUSED VOL017
200  02INCBK2 00000  00:00:00 UNUSED VOL018
200  LOG      00000  00:00:00 UNUSED VOL019
200  LOG      00000  00:00:00 UNUSED VOL020
200  LOG      00000  00:00:00 UNUSED VOL021
200  TRANS    00000  00:00:00 UNUSED VOL022
200  TRANSDSK 00000  00:00:00 UNUSED SQLDB200 TRANSDSK * 197
300  ARCHIVE  00000  00:00:00 UNUSED VOL023
300  BACKUP   00000  00:00:00 UNUSED VOL024
300  BACKUP2  00000  00:00:00 UNUSED VOL025
300  01INCBK  00000  00:00:00 UNUSED VOL026
300  01INCBK2 00000  00:00:00 UNUSED VOL027
300  02INCBK  00000  00:00:00 UNUSED VOL028
300  02INCBK2 00000  00:00:00 UNUSED VOL029
300  LOG      00000  00:00:00 UNUSED VOL030
300  LOG      00000  00:00:00 UNUSED VOL031
300  LOG      00000  00:00:00 UNUSED VOL032
300  TRANS    00000  00:00:00 UNUSED VOL033
300  TRANSDSK 00000  00:00:00 UNUSED SQLDB300 TRANSDSK * 198
```

Figure 31. Sample database TAPES File With Data Restore Entries

Table 7. TAPES File Format

Value	Description
SERIES	A three-digit number indicating the output media series. Output media with the same series number are considered by Control Center to belong together as a single logical group.
TYPE	Assigns output media to a specific usage. Possible values: **ARCHIVE** Use for full database archiving activities. **ssINCBKd** Use for each Incremental Backup. "ss" is the subseries number and "d" is blank (indicating the primary) or "2" (indicating the secondary) Incremental Backup. **LOG** Use for database log archiving activities. **TRACE** Use for database tracing activities. **BACKUP** Use for primary user archives using Data Restore BACKUP. **BACKUP2** Use for secondary user archives using Data Restore with dual BACKUP. **TRANS** Use for translation of database archives into Data Restore BACKUP format. **TRANSDSK** Use for the disk location of the 3 Data Restore work files needed for translation (SYS0001, DIRWORK, & HEADER). **UNLOADID** User generated id for unloading DBSPACES using Data Restore. Used in the UNLTAPES file only.
DATE	The Julian date (*yyddd*) when output media was last used by Control Center. Should be equal to 00000 when tape is unused.
TIME	Time (*hh:mm:ss*) when output media was last used by Control Center. Should be equal to 00:00:00 when tape is unused.
STATUS	Control Center uses this field to track the status of the output media. Common status values include: **UNUSED** Output media is not being used. **FILDEF** Output media currently being used. **MOUNTD** Output media is currently mounted. **FILLED** Output media has been used.
VOLID	Tape volume serial number if output media is TAPE.
FILENAME	CMS filename if output media is DISK.
FILETYPE	CMS filetype if output media is DISK.
FM	CMS filemode (disk access mode) if output media is DISK.
CUU	*Virtual* disk address (CUU) if output media is *DISK* and *TYPE* is ARCHIVE, LOG or TRACE. *Real* disk address (CUU) if output media is *DISK* and *TYPE* is a *Data Restore* function (*TRANSLATE, BACKUP2, INCBK, BACKUP, or UNLOAD*).
CI	Century Indicator (1 = 20th Century, 2 = 21st Century) This is not displayed on the panels.

Important:

There are several restrictions and special precautions that apply when output is to disk rather than tapes. These are discussed in detail in “About Archiving to Disk” on page 236.

Setup Step 11: Update Database Parameters File

Update Option P from Control Center's main menu. Then choose the update option found on the Database Parameter's menu (see Figure 90 on page 213) to update the database parameters file. Information about the Data Restore values in the database parameters file can be found in Figure 98 on page 224. For a complete description of the Database Parameters Tool, see “About the Database Parameters Tool” on page 213.

Control Center maintains a set of database startup and control parameters for each database it manages. The PARMS file must be updated for each database that will execute Data Restore functions.

Setup Step 12: Define & Update Database TAPES Files

Use tape maintenance functions from the Data Restore Menu, Figure 115 on page 263, to define the Data Restore BACKUP (*TAPES*) and UNLOAD (*UNLTAPES*) files. These files will be used for the Data Restore BACKUP and UNLOAD operations. Select Option **BM** to add BACKUP and TRANSLATE tapes to the *database* TAPES file and option **UM** to define the UNLOAD, *database* UNLTAPES file. For more details on tape usage, refer to the Data Restore chapters on BACKUP, UNLOAD, and TRANSLATE.

Setup Step 13: Update the Control Center SQLMSTR Profile

To update the Control Center service machine SQLMSTR profile:

- Link to the Control Center's 191 disk and copy the SQLMSTR PROFILE to your 191 disk.
- Using an editor (such as XEDIT), you must add this statement to the User authority section of the SQLMSTR Profile to allow the Data Restore machine to communicate with Control Center:
  - *USER DRMACH AT nodename 4*
- Use the CMS *sendfile* to transfer the SQLMSTR PROFILE to the service machine's 191 disk.

Setup Step 14: Create the LINKPWDS File

To use Data Restore functions, a LINKPWDS file is required for each Data Restore machine. See “Password File Setup for the Data Restore machine” on page 504 for specific instructions on creating this file. After you have created the *DRMACH* LINKPWDS, use the CMS *sendfile* to transfer it to the Control Center service machine's 191 disk.

Setup Step 15: Restart Control Center

To restart Control Center, or to initialize the new SQLMSTR profile, execute the Control Center NEWPROF command. This can be invoked either through CMS by entering *SQM NEWPROF*, or from the Control Center menu interface by first selecting Option **G** on the Control Center main menu, followed by Option **N** from the General Utilities Functions menu.

Once these steps have been completed, log off the MSTROWN machine.

Data Restore Machine Setup

Steps 16 through 21 complete the Data Restore interface to Control Center.

Setup Step 16: Log On to the Data Restore Machine (DRMACH)

Log on to the Data Restore machine. At the CMS prompt type: *query secuser*. If the response does not indicate the Control Center service machine, review the VM directory entries to determine if the SCIF interface has been established. See “Setup Step 2: Redefine the Data Restore Machine (DRMACH)” on page 95.

Setup Step 17: Link and Access the Control Center Code Disk

Access the Control Center code disk with an available disk access mode. If you did not include a LINK statement in the VM directory (Step 2), you will first need to link to the code disk with an available CUU that is **not** the same as any address that the database uses.

Note: DRMACH must be authorized to access the Control Center code disk in READ mode.

Setup Step 18: Create the Data Restore SQLMSTR Profile

To create the Data Restore machine's SQLMSTR profile, link to the Control Center service machine's 191 disk and copy the SQLMSTR profile to your 191 disk.

- Using an editor (such as XEDIT), change the USER statement for the Data Restore machine from a Level 4 to Level 5 authority.
  - *USER DRMACH AT nodename 5*
- Add appropriate Data Restore Administrators IDs
  - *USER MSTROWN AT nodename 5*

Setup Step 19: Copy and Update DRFPROF EXEC

1. Use the following commands to make a backup copy of the existing Data Restore machine's PROFILE EXEC and to replace it with the DRFPROF EXEC from the Control Center service machine's code disk.

   ```
   COPYFILE PROFILE EXEC A PROFILE OEXEC A
   COPYFILE DRFPROF EXEC * PROFILE EXEC A (REP
   ```

2. The PROFILE EXEC file may need to be modified for any LINK and ACCESS statements required at your installation.

Note: You must not link to any disk using a virtual address that is the same as the real address of any database directory, log, or data disk, as Data Restore will detach those disks.

- If you did not include a LINK statement in the VM directory (Step 2) for the Control Center code disk, add the necessary link and access statements to the PROFILE EXEC.

- If the Data Restore code is not on the 191 disk, then include the appropriate line and access statements for the Data Restore code disk.
- If Data Restore needs access to other local disks, the PROFILE EXEC should be modified to add the required LINK and ACCESS statements. When the DYNAM/T , EPIC or VMTAPE product is used, a link may be required to provide the database with access to code required for tape operation. For example, the DMSTVI module must be accessed when VMTAPE or EPIC is used. The DMSTVI module is required to support multivolume archive tapes.

Important:

You do **NOT** need to link or access output disks for backups, translates, or unloads. These actions will be done by Control Center, based on information in the TAPES files, prior to executing the Data Restore Function.

Setup Step 20: Rename XTS9X001 EXEC

The XTS9X001 EXEC is a template provided by Data Restore to allow users to change log Labeldefs during Reload Recovery. Control Center provides this EXEC for you on its code disk. You must rename the XTS9X001 EXEC on the Data Restore code disk so that the Control Center code will execute.

Locate the XTS9X001 EXEC on the Data Restore code disk and execute this CMS command:

```
RENAME XTS9X001 EXEC fm = EXECDRF =
```

where "fm" is the filemode of the Data Restore disk.

Setup Step 21: Run DRMSTR EXEC

If all of the previous steps were completed successfully, then the Data Restore machine can be started by entering this command at the CMS command prompt:

```
DRMSTR
```

The DRMSTR process will disconnect the Data Restore machine from your console and will start the Data Restore interface with Control Center.

Setup Step 22: Send the Data Restore Machine (DRMACH) a Message

To determine if Data Restore is properly setup, from an authorized Control Center ID, *MSTROWN*, where *DRMACH* is the Data Restore machine name, enter the CMS command:

```
SQM DRMACH CMS TELL MSTROWN HELLO
```

This tells the Data Restore machine to send you a "HELLO" message, thereby verifying that the SCIF processing paths are complete. You should see these messages on the screen from which you issued the command above:

```
'MSG FROM' MSTSRV ':TELL' MSTOWN 'HELLO ISSUED TO' DRMACH
'MSG FROM' DRMACH 'HELLO'
```

Chapter 10. Version Tool

Overview

The Version Tool was developed to provide you and the Control Center support staff with an easy way to gather comprehensive information about code, files, PTFs, and the overall product environment.

Features

The Version Tool:

1. Lists all Control Center code files and current PTFs on all minidisks accessed by the user or product.
2. Searches for specific PTFs or ranges of PTFS on all minidisks accessed by the user or product.
3. Reports duplicate code files.
4. Reports missing code files.
5. Lists non-Control Center files on a code or other selected disk.
6. Reports selected CP and CMS system information.
7. Runs against local or remote Control Center machines.

Invocation

The Version tool is executed as a command from CMS. Be sure you are addressing the correct service or support machine by entering:

```
SQM ( SQM-ID
```

where SQM-ID is the userid of the machine.

Command Format

The Version command format is:

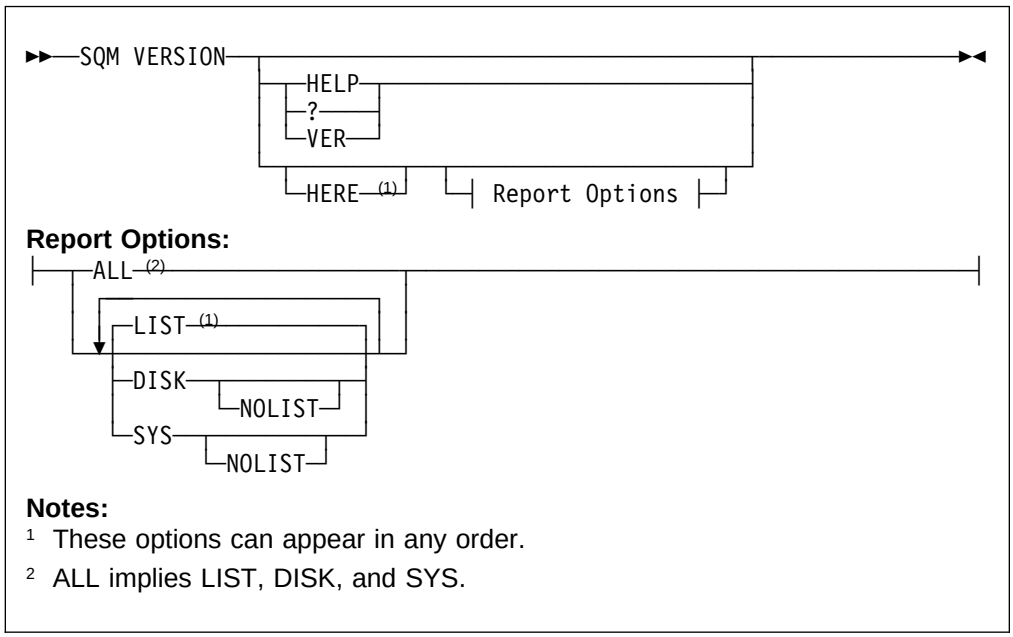


Figure 32. Version command format

The order of options is not important. Reports are created and saved on the Control Center A-disk and sent to the user's reader. Old reports are erased before new reports are created. Execution of the Version command can be restricted to the service machine's administrators by changing the VERSION entry in the PROFILE file to level 5.

General Options

The general options of the Version command are:

Option	Description
HELP/?	Displays documentation for the Version command. All other options are ignored. Can also be invoked by typing: HELP SQMVERSN
VER	Displays the current version only. No report is sent and no "LAST PTF" information is given. All other options are ignored.
ALL	Creates and sends LIST, DISK and SYS reports.
HERE	Gathers the selected PTF, System, or disk information based on minidisks the user is currently accessing. Runs Version on the user's virtual machine, rather than a service machine as is normally the case.

PTF List Options

When you choose the PTF list options, the product looks for occurrences of every file found in the SQLMSTR LIST file. Its PTF information can be found at the bottom of executable files that have a file type of EXEC, XEDIT or $EXEC. $EXEC files are template files that are customized for local use either by the product or by the user. It searches backwards through these files, extracting PTF numbers and information. Depending on the option chosen, a file will be searched until the current PTF record is found or the search condition can no longer be met.

The valid PTF List options are:

Option	Description
NOLIST	Avoids creating and sending a PTF list to the user's reader. This is useful if only the DISK or SYS report is desired.
LIST	Examines Control Center code file on all accessed disks and creates a listing and analysis of the current PTF level of each file. By default, a PTF list and PTF analysis are generated each time the command is executed unless the NOLIST option is used. A PTFLIST report is sent to the user's reader. The filename is the userid of the service machine. The report includes PTF information about files on all accessed disks, including: • Each code file on each disk and its latest PTF • Duplicate code files • Missing code files • The last PTF applied

Note:

The NOLIST, VER, and HELP options suppress creation of the PTF report

Non-Control Center Files

The options to list non-Control Center files are:

Option	Description
DISK	Reports files on the code disk that are not in the SQLMSTR LIST file. The product only allows files in the SQLMSTR LIST file to be received to its code disk during normal disconnected operation. This option helps identify local execs or other files that have been placed there. An XTRACODE report is sent to the user's reader. The filename is the userid of the service machine. If all the files on the disk are Control Center files, the report is not sent to the user.
DISK = fm	You can change the disk to be examined by using this parameter, where "fm" is the file mode of a disk that the product has accessed. You can determine the disks that it is accessing by using the SYS option. By default, the location of the code disk is determined by the location (fm) of the first SQLMSTR LIST file found. This option always uses the first SQLMSTR LIST file it finds as the basis for its examination regardless of what "fm" is chosen.

System Report Option

The System Report option provides information about the environment in which the product is operating. When problems occur, this information can be used to verify that everything is set up correctly.

Option	Description
SYS	Reports current CMS and CP system information about the service machine including: • CP level • Virtual memory and unallocated memory • System privileges • Disks Control Center is accessing • NUCXMAP information • Segments in use • User IDs linked to the product code disk An SQMSYS report is sent to the user's reader. The filename is the userid of the service machine.

PTF Report

The figure below is an example of a PTF report.

```
                        MSTRSVRO CONTROL CENTER PTF REPORT
                Control Center for VM Version 6 Release 1 Modification 0
                                01 Jan 1998

Type: SQM VERSION HELP   for explanation of this report

Files listed in SQLMSTR LIST D ...  350       LAST PTF:       UV58736
SQLMSTR code files on disk D .....  349       Missing files:  1
Code files that have duplicates ..    8       FINDPTF:        CURRENT

            Control Center Code  File     File     PTF         FILE/PTF
PTF Number FileName FileType FM   Date     Time     Date   DUP  DESCRIPTION
---------- -------- -------- -- -------- -------- -------- ---- ---------
           CPVENDBA PARMS    A1 10/04/96 10:13:30               Database
           CPVENDBA TAPES    A1 10/04/96 10:12:52               Database
           SQLDBA   $PARMS   C1  5/05/96 12:32:57               Database
           DATABASE MESSAGES D1  5/05/96 11:02:41               Data Base
           DATABASE MSGLIST  D1  5/20/96 16:11:00               Message
Base Code  DBADEXT  EXEC     D1  6/16/96  6:52:52 06/14/96      Database
**DANGER** SQLMAINT EXEC     *     ERROR     FILE  MISSING      Control Center
           SQLMSTR  DIRECTRY D1 10/03/96 11:38:28          DUP2 Database
           SQLMSTR  DIRECTRY M1  9/15/96  9:30:48          DUP2 Database
Base Code  DBADSPAC EXEC     D1  6/16/96  6:52:54 06/14/96      Database
UV58520    DBEXTMAP EXEC     D1  7/22/96 11:26:21 07/22/96      *FIX PROB
Base Code  DBFORCE  EXEC     D1  6/16/96  6:52:50 06/14/96      FORCE use
           DBINIT   $CONTROL D1  5/05/96 12:32:54               Database
           DBINIT   CONTROL  D1  7/19/96 17:12:13          DUP2 Database
           DBINIT   CONTROL  M1  7/16/96  9:04:41          DUP2 Database
UV58119    DBINIT   EXEC     D1  2/09/96 14:47:28 02/09/96 DUP2 *REMOVE PASS=
```

Figure 33. Version PTF Report

The PTF Report field descriptions are as follows:

Field	Description
PTF Number	Contains the most current PTF number found or "Base Code" if no PTFs have been applied to the file since the base code for the current release was installed. This field is blank if the file type is NOT either EXEC, XEDIT or $XEDIT. IF the file is not on any accessed disk, the field contains "**DANGER**" followed by a message.
FileName FileType FM	The name, type and mode of the file that is being reported on.
File Date File Time	The current date and time stamp.
PTF Date	The date the PTF was issued.
DUP	If duplicates are found, the field contains "DUP" followed by the number of times the file was found. The report is sorted by file name, so duplicate files should be grouped together in the report.
File/PTF Description	If no PTF is found, this field contains a description of the file. If a PTF is found, this field contains an "*" followed by a description of the PTF.

Chapter 11. Managing the Environment

This chapter provides detailed information concerning the operation of Control Center, including the Control Center-to-database server communication interface necessary to effectively manage the Control Center environment and resolve problems that can occur. Although it is intended primarily for those involved in the management and administration of Control Center itself, all users can benefit from the material presented.

Chapter 12, “Getting Started” on page 135 introduces you to the Control Center panel interface and provides enough information to get you started using Control Center in your environment.

Information presented in this section assumes that you have read Chapter 2, “Architecture” on page 13.

Control Center supports command mode operation and panel mode operation. This section deals primarily with panel mode operation. Additional information for command mode operation, including command line syntax, is provided in Appendix G.

Finally, throughout this section we've included specific database information also found in the DB2 Server for VM manual set. It is provided for convenience and should not be used as a reference. The proper reference for database information is the database manual set for the level of DB2 Server for VM that you are using.

The Control Center Administrator is responsible for the configuration and operation of Control Center's virtual machines. There can be more than one administrator defined to a given machine ID. This authority is given to a user by inserting a USER statement in the SQLMSTR PROFILE file with a privilege level of 5 (during installation or later through the panel interface). This privilege level will enable that user to perform any function that is available within Control Center, including all DBA function for all databases under a service machine's control. A Control Center Administrator must therefore be given DB2 Server for VM DBA authority to be able to successfully exploit all available Control Center functions.

At installation time at least one user ID will become Administrator. With the Control Center panel interface, the Administrator can also promote other users to be administrator. The Control Center Administrator will automatically acquire Control Center DBA authority over all databases controlled by a given service machine. There is one special user who is called the owner of the service machine. This user will automatically have Administrator authority and will receive the SQLMSTR CONSOLE file that is closed and transferred from the service machine each day at midnight.

Administrator authority is required to install a new database under a service machine. This function is normally performed by using the New Database Setup **(N)** option from the Control Center Main Menu.

Each Administrator user ID identified to Control Center will receive many notes and messages during operation. Many of these will pertain to database activities and others will pertain to the Control Center operation itself. Each time Control Center is started, a report on current disk usage will be sent to the Administrators. If

Control Center's 191 A-disk approaches 90% full, then the Administrator should take some action to remove files that are not needed or should increase the size of the disk.

Communications Methodology

Messages are the central form of communication between the Control Center service machines and users. The service machine will perform specific actions upon the receipt of predefined command messages from authorized users. These predefined messages are generated by the Control Center interface program, either in command mode or panel mode.

Messages are also received from the database machines through the *Single Console Interface Facility* (SCIF). These messages are analyzed by the service machine to determine what action should be performed and whether any Control Center user(s) should be alerted. These messages are usually issued by the database manager itself as part of archiving, recovery, operator commands, or some other database activity. The Control Center code contains the logic necessary to respond to these messages with an appropriate action, thereby completing any administration function without DBA intervention.

Understanding Control Center Interrupt Processing

Control Center is designed to recognize two distinct types of interrupts, scheduled and unscheduled.

Scheduled interrupts are controlled by the SQLMSTR TIMES file on the service machines's 191 A-disk. This file is updated whenever authorized users schedule new events or remove old ones. Three entries are included in the SQLMSTR TIMES file for normal Control Center operation and must not be removed.

Unscheduled interrupts occur in two basic forms: *messages* and *reader files*. All interrupts are controlled by the SQLMSTR PROFILE file, which defines all actions that Control Center will perform when an unscheduled interrupt occurs. Specific actions are defined for known types of messages and reader files. General actions are defined for any unexpected messages or files.

Starting and Stopping a Service Machine

Control Center is ready to be started after all control files and execs have been customized. To make sure that the correct disks are accessed, the PROFILE EXEC should be executed first. After setting up the environment, the PROFILE EXEC will leave the service machine in native CMS.

The startup process will read the values from several control files from the 191 A-disk and place them in memory for the current session. This enables Control Center to repeatedly access these values without performing any additional disk I/O. The control files that are read into memory during startup are:

- SQLMSTR CONTROL
- SQLMSTR PROFILE

- DATABASE MESSAGES
- Each database PARMS file

Remember that when any of the previously mentioned files are changed on the service machine's 191 A-disk, the new values will not be recognized by the machine until they are again read into memory. This is done when the service machine reinitializes itself each day (by stopping and then restarting) at midnight, or when explicitly specified by the SQM NEWPROF command.

Manual Startup

While logged onto the service machine, the SQLMSTR EXEC can be executed to start the Control Center facility. The first step of the SQLMSTR EXEC is to disconnect the virtual machine. This will allow the service machine to run in disconnected mode. It will take Control Center a few minutes from the disconnection to become fully operational.

If for some reason there is a need to watch the startup procedure, the SQLMSTR EXEC file can be modified to suspend the disconnection by commenting out the CP DISC line. To then manually disconnect the service machine after startup, enter #CP DISC while the service machine is in RUNNING mode.

Automatic Startup

The PROFILE EXEC is designed to detect whether a terminal is present at startup. If the service machine has been AUTOLOGed and there is no terminal, the PROFILE EXEC will automatically call the SQLMSTR EXEC, which will start up the Control Center facility. This is the preferred way to start the service machine.

Once Control Center has started, communications with each database is established through the SCIF. Communications with Control Center users is also possible through the panel interface.

Part of the normal service machine startup routine will perform maintenance of its environment. The previous SQLMSTR CONSOLE file will be closed and sent to the OWNER identified within the SQLMSTR PROFILE. The DATABASE MESSAGES file, SQLMSTR CONTROL file, and each database PARMS file will be read and their parameter values will be stored in memory for access until the next startup. Old copies of the SQLMSTR LOGyyddd files will be purged, based on the parameter value of Log_copies in the SQLMSTR CONTROL file. Old archives within the database ARCHHIST file of each database will also be purged.

Stopping Control Center

The service machine is designed to run continuously (although it stops and restarts itself daily) if for some reason it must be stopped, the administrator should check that no archives, coldlogs, recoveries, or other special functions are active for any database. If the service machine is stopped during one of these activities, it is highly probable that the activity will fail.

The service machine can be stopped from another user ID by using the SQM STOP command. The user ID must have Administrator authority.

The service machine will then be left at the CMS ready level. Note that once the stop command has been issued, no further communications with it is possible. To restart the service machine, log onto the machine and follow the instructions discussed in the preceding paragraphs.

The service machine can also be stopped by logging onto it. After logging on, the ENTER key must be depressed twice to provide a console interrupt that will stop the Control Center code and leave the user in CMS. This method of stopping a service machine may leave it in an abnormal state. To restore the normal CMS environment, it may be necessary to IPL CMS. If a periodic RUNNING condition prevents typing on the keyboard, use the SET BLIP OFF command. (It can be difficult to type the command in between blips.)

Starting and Stopping a Support Machine

The Control Center support machines function primarily as vehicles for running Database Administration jobs such as reorganizations. They do not control or communicate directly with any databases with the SCIF connection. The primary consideration to stopping and starting a support machine is that the machine must be available (logged on and in disconnected state) when the service machine sends a job to it to run. If the support machine is not available, the job will fail.

Operational Note: Place the Control Center support machine ID in the AUTOLOG list. See Chapter 5, “Installing a Support Machine” on page 51. This will prevent the machine from not being available when required. Alternatively, the Control Center Administrator will have to ensure that the support machine is logged on and put into a disconnected state prior to running any jobs.

The methods of starting and stopping a service machine apply to a support machine as well, since the code executed is the same, with the exception of a tailored PROFILE EXEC which does not do as much as the service machine's.

Starting and Stopping the Database Machine

DB2 Server for VM database machines should be stopped and started solely through the managing Control Center.

If a database machine has to be stopped, it should be done through Control Center. If you log onto the database machine and issue a SQLEND command, Control Center would no longer be synchronized with the database, since it would not receive any database manager messages.

Note:

Do not log onto a database machine since this cuts the SCIF communication with the Control Center service machine. All database functions (operator commands) and any CP commands can be issued through the Control Center panel interface.

If it is necessary for whatever reason to restart a database machine when logged onto it, execute the PROFILE EXEC, then respond to the prompt to enter

DBSTART. This will ensure that the database is resynchronized with the service machine.

Adding Users to Control Center

New users may need to be added as users, DBAs, or administrators during the life of Control Center. Others may need to be deleted or have their authorities changed. This should be done in the following manner:

Administrator additions, changes, or deletions should be handled through the SQLMSTR Authorization **(AU)** option on the SQM panel interface. This option also allows you to get a list of all current authorizations on a service machine. This list is helpful in determining current levels of accessibility to the service machine. The SQLMSTR PROFILE is changed by this function.

Use the Database Parameters tool to add, delete, or change DBA administrators, operators, and general users to a specific database. The database PARMS file is changed by this function.

Note:

Updates to the affected files will not become effective until the service machine is stopped and restarted. A convenient way to effect the changes is to use the SQM NEWPROF command, which causes the service machine to stop and restart.

Maintenance

There are several types of maintenance involved in running Control Center. Most are automatic. Some control file maintenance or modification may be required to tailor an installation's needs when running Control Center.

Routine Maintenance

Control Center is mostly self-maintaining. It routinely purges back copies of files from the 191 A-disk and will maintain only the most recent information within log files. The SQLMSTR CONSOLE file of the service and support machine's console output is closed and sent to the Administrator machine owner each time the machine is brought down.

Disk Maintenance

If the service or support machine's 191 A-disk begins to become full, you may need to reduce the number of versions that are retained of the log files (Log_copies parameter of the SQLMSTR CONTROL file) and the number of backup copies that are kept of files that are changed (Backup_copies parameter of the SQLMSTR CONTROL file). Another file that may need to be purged periodically is the machine NETLOG file. This file has a filename the same as the service or support machine name. It is maintained by CMS to record when files are sent and received by the machine.

Control File Maintenance

When installing the Control Center system, the installation process will build and modify several control files. In this section a general description of the most important ones are given to provide a better understanding of the things that play an important role during the service machine operation.

These control files reside on the service machine's 191 A-disk (except the SQLMSTR DIRECTRY and the DBINIT CONTROL files, which are kept on the 195 code disk).

Maintaining the SQLMSTR PROFILE File

The SQLMSTR PROFILE file is a fundamental part of the disconnected virtual machine architecture. It controls the interface to users and database machines by defining valid commands and designating the corresponding code segment for each command. A secondary purpose of this file is to identify the special Administrator privileged users. These USER entries are normally the only portion of this file changed for the local installation. Advanced users can also modify the MESSAGE command privilege levels to change the default command privileges for the DBA, OPERATOR, and USER authority levels.

This file must be modified to add or remove authorized Control Center administrators. This function is automated within Control Center panel interface with the SQLMSTR Authorization **(AU)** option.

The SQLMSTR PROFILE file can also be modified if an installation wants to change the commands that are authorized for the various types of users (Administrators, Database Operators, Database Users). Each command is indicated within the SQLMSTR PROFILE by one of the MESSAGE entries. Each MESSAGE entry contains a single-digit number between 1 and 5 which indicates the authority level required to execute the associated command.

With proper precautions (including a backup copy of the original version), the authority levels of these MESSAGE entries can be modified for local preferences. See Appendix E, “Authorizations” on page 511.

- A level 5 MESSAGE privilege corresponds to the Control Center Administrator authority.
- A level 4 MESSAGE privilege is reserved for DATABASES and Control Center Support machines.
- A level 3 MESSAGE privilege corresponds to the Control Center Database Administrator authority.
- A level 2 MESSAGE privilege corresponds to the Control Center Database Operator authority.
- A level 1 MESSAGE privilege corresponds to the Control Center Database User authority.

Note that authority levels are hierarchical, such that someone with Database Administrator authority to a database would be able to execute all MESSAGE commands with a privilege level of 1, 2, or 3. A Database Operator would be able to execute commands with a privilege level of 1 or 2.

Figure 34 on page 120 is an example of the SQLMSTR PROFILE. All uppercase entries are keywords that are required. All lowercase entries must be replaced with installation-specific values.

```
OWNER sqmowner
LOG HOLD 7
USER ALL 3
USER cntrlid  AT node1    5
USER userb    AT nodeb    5
USER userc    AT nodec    5
MESSAGE CMD      5 ( .MS
MESSAGE CMS      5 ( EXEC SQMCMD   .US .NO .MS
MESSAGE SQMPROFU 5 ( EXEC SQMPROFU .US .NO .MS
MESSAGE SQMQPROF 5 ( EXEC SQMQPROF .US .NO .MS
MESSAGE RDRLIST  5 ( EXEC SQMRDRL  .US .NO
MESSAGE SQMCONS  5 ( EXEC SQMCONS  .US .NO .MS
MESSAGE SDRESTRT 3 ( EXEC SDRUSTRT .MS
MESSAGE SQMDBU   5 ( EXEC SQMDBUR  .US .NO .MS
MESSAGE SDRELOAD 3 ( EXEC SDRELOAD .US .NO .MS
MESSAGE SDLISTLG 3 ( EXEC SDLISTLG .US .NO .MS
MESSAGE SDAPLYLG 3 ( EXEC SDAPLYLG .US .NO .MS
MESSAGE SDUNLOAD 3 ( EXEC SDUNLOAD .US .NO .MS
MESSAGE SDRTRANS 3 ( EXEC SDRTRANS .US .NO .MS
MESSAGE SDSHOWDB 3 ( EXEC SDSHOWDB .US .NO .MS
MESSAGE SHOWDBS  3 ( EXEC SDSHOWDB .US .NO .MS
MESSAGE SQMACCES 3 ( EXEC SQMACCES .US .NO .MS
MESSAGE SQMFLIST 3 ( EXEC SQMFLIST .US .NO .MS
MESSAGE TRACE    3 ( EXEC SQMTRACE .US .NO .MS
MESSAGE SQMDBCMS 3 ( EXEC SQMDBCMS .US .NO .MS
MESSAGE SQMDCHKQ 3 ( EXEC SQMDCHKQ .US .NO .MS
MESSAGE SQMADBSP 3 ( EXEC SQMADBSP .US .NO .MS
MESSAGE SQMADBEX 3 ( EXEC SQMADBEX .US .NO .MS
MESSAGE SQMCDBEX 3 ( EXEC SQMCDBEX .US .NO .MS
MESSAGE SQMDDBEX 3 ( EXEC SQMDDBEX .US .NO .MS
MESSAGE SQMCIREO 3 ( EXEC SQMCIREO .US .NO .MS
MESSAGE SQMCOLDL 3 ( EXEC SQMCOLDL .US .NO .MS
MESSAGE SQMEVENT 3 ( EXEC SQMEVENT .US .NO .MS
MESSAGE SQMEVDEL 3 ( EXEC SQMEVDEL .US .NO .MS
MESSAGE SQMJDONE 3 ( EXEC SQMJDONE .US .NO .PR .MS
MESSAGE SQMJEXEC 3 ( EXEC SQMJEXEC .US .NO .MS
MESSAGE SQMJSTRT 3 ( EXEC SQMJSTRT .US .NO .MS
MESSAGE SQMFLDEL 3 ( EXEC SQMFLDEL .US .NO .MS
MESSAGE SQMREORF 3 ( EXEC SQMREORF .US .NO .MS
MESSAGE SQMREORG 3 ( EXEC SQMREORG .US .NO .MS
MESSAGE SQMREOTI 3 ( EXEC SQMREOTI .US .NO .MS
MESSAGE SQMMNRPR 3 ( EXEC SQMMNRPR .US .NO .MS
MESSAGE SQMMNSN  3 ( EXEC SQMMNSN  .US .NO .MS
MESSAGE SQMMNSL  3 ( EXEC SQMMNSL  .US .NO .MS
MESSAGE SQMMNSR  3 ( EXEC SQMMNSR  .US .NO .MS
MESSAGE SQMTAPEU 3 ( EXEC SQMUTAPE .US .NO .MS
MESSAGE START    2 ( EXEC SQMDBREQ .US .NO .MS
MESSAGE STOP     2 ( EXEC SQMDBREQ .US .NO .MS
MESSAGE SET      2 ( EXEC SQMDBREQ .US .NO .MS
MESSAGE RESET    2 ( EXEC SQMDBREQ .US .NO .MS
MESSAGE FORCE    2 ( EXEC SQMDBREQ .US .NO .MS
MESSAGE CANCEL   2 ( EXEC SQMCANCL .US .NO .MS
MESSAGE SQMDBEND 2 ( EXEC SQMDBEND .US .NO .MS
MESSAGE SQMDBEGN 2 ( EXEC SQMDBEGN .US .NO .MS
MESSAGE SQMMODEU 2 ( EXEC SQMMODEU .US .NO .MS
MESSAGE SQMSTATU 2 ( EXEC SQMUSTAT .US .NO .MS
MESSAGE SQMARCH  2 ( EXEC SQMARCH  .US .NO .MS
MESSAGE SQMRECOV 2 ( EXEC SQMRECOV .US .NO .MS
MESSAGE SQMRECQT 2 ( EXEC SQMRECQT .US .NO .MS
MESSAGE SQMRECST 2 ( EXEC SQMRECST .US .NO .MS
MESSAGE SQMRECTQ 2 ( EXEC SQMRECTQ .US .NO .MS
MESSAGE SQMDBINI 2 ( EXEC SQMDBINI .US .NO .MS
MESSAGE SQMMNSVU 2 ( EXEC SQMMNSVU .US .NO .MS
MESSAGE SQMTAPEA 2 ( EXEC SQMTAPEA .US .NO .MS
MESSAGE SQMCUARC 2 ( EXEC SQMCUARC .US .NO .MS
MESSAGE SQMQUARC 2 ( EXEC SQMQUARC .US .NO .MS
MESSAGE SHOW     1 ( EXEC SQMUSREQ .US .NO .MS
MESSAGE COUNTER  1 ( EXEC SQMUSREQ .US .NO .MS
MESSAGE SQMQSTAT 1 ( EXEC SQMQSTAT .US .NO .MS
MESSAGE SQMDBLST 1 ( EXEC SQMDBLST .US .NO .MS
MESSAGE SQMEVDIS 1 ( EXEC SQMEVDIS .US .NO .MS
MESSAGE SQMSFILE 1 ( EXEC SQMSFILE .US .NO .MS
MESSAGE VERSION  1 ( EXEC SQMVERSN .US .NO .MS
MESSAGE SQMGCTRL 1 ( EXEC SQMGCTRL .US .NO .MS
MESSAGE ?          ( EXEC SQMMSG   .US .NO .PR .CO .MM
MESSAGE ON
READER * *       1 ( EXEC SQMRDR .US .NO .PR .SP .FN .FT
READER ?           ( HOLD
CLASS *
```

Figure 34. Example SQLMSTR PROFILE File

Maintaining the SQLMSTR CONTROL File

The SQLMSTR CONTROL file is used for housekeeping purposes. It is built during Control Center installation process when the C5648A70 exec is invoked.

```
 mm/dd/yyyy                    Control Center                    hh:mm:ss
*--------------------- SQLMSTR CONTROL File Parameters --------------------*
| Command ==>                                            CTRLID: MSTRSRV1 |
|                                                        NODE:   VMSYSTM1 |
| Sqlmstr-userid        ==> SQLDBA                                        |
| Log-copies            ==> 5        (Number of Days to keep LOG files)   |
| Spool-console         ==> Y        (Y or N, spool Control Center console|
| Backup-copies         ==> 3        (Old copies to keep of modified files|
| Tape-manager          ==> VMTAPE   (e.g. DYNAMT, VMTAPE)                |
| Dynamt-method         ==> ________ (specify DYNOPEN, if used w/DYNAMT)  |
| Tape-manager-userid   ==> VMTAPE   (Disconnected machine id)            |
| Tape-code-userid      ==> $MAINT   (Tape manager code disk id)          |
| Tape-code-address     ==> 19E      (Tape code disk virtual address)     |
| Tape-code-filemode    ==> Y        (Tape code disk link filemode)       |
| Disk-warn             ==> 80       (Control Center disk full warn percnt)|
| Disk-full             ==> 81       (Seriously full warn percent)        |
| Disk-access-options   ==> ________________________________________      |
| Jobout-retention-days ==> 30       (Number of days to keep Job output   |
| Date-time-format      ==> USA      (USA, EUR, ISO, JIS)                 |
|                                                                         |
| Press Enter to process changes                                          |
|                                                                         |
*-------------------------------------------------------------------SQMCTR-*
 PF:   1 Help    3 End
```

Figure 35. Sample SQLMSTR CONTROL File

The service machine will receive many messages and will produce many listings and logs. To prevent the 191 A-disk from being filled, only a limited number of copies will be held, specified by the Log_copies and Backup_copies parameters. Keep these numbers to the minimum needed. A full discussion of all fields is in Chapter 4, “Installing the Service Machine” on page 37.

This file is normally configured during installation and will not need to be changed. If modifications need to be made after installation, they can be done two ways.

One method would be to stop the service/support machine, log onto it, modify the file, then start the machine by running the SQLMSTR EXEC. Another method would be to use the FILELIST **(F)** option under the General Control Center commands **(G)** option on the main menu. This will allow you to update the file from your user ID and send it to the service/support machine.

Maintaining the SQLMSTR DIRECTRY File

The SQLMSTR DIRECTRY file provides the interface with a directory of all database and service machines at a given installation. It is used to make a unique identification for each database in the system (local and remote nodes), and to make switching easier within the panel and command mode interfaces. When this file is properly built and available to Control Center users, only the database name needs to be supplied in commands. The user interface will reference the SQLMSTR DIRECTRY file to automatically direct the communication to the correct service machine.

In this file each database is described in terms of:

- Database Machine Name or Database Nickname
- Database Machine Name (Virtual Machine Name)
- Service Machine User ID
- Service Machine Node ID

The service machine's node ID must be the same as the database node ID. If a database resides on a remote node ID, a service machine user ID at that node ID is required (see example). With this information, every database, even if it resides on a remote node, will be unique for the system.

This file will be updated when a new database is identified to Control Center. If the machine on which the database is identified owns the code disk, the file will be replaced on the code disk. Otherwise, the updated (merged) file is sent back to the user who defined the database with instructions to send the file to the service machine which owns the code disk, so that the file can be properly replaced. This must be done by a Control Center Administrator. Note that this must also be done for any SQLMSTR DIRECTRY files on remote nodes.

A helpful tip for installations with database virtual machine names that are different from the dbname of the database is to use the dbname as the nickname within the SQLMSTR DIRECTRY file. This will allow users to specify either the machine name or the dbname when using the Control Center interface (the service machine will understand either name). If the same DBNAME is used within multiple database machines, then an alternate naming scheme must be chosen to keep the nicknames within the SQLMSTR DIRECTRY file unique. The following example illustrates this point:

```
SQLDBA1  SQLMACH1 MSTRSRV1 VMSYSTM1
SQLDBA2  SQLMACH2 MSTRSRV1 VMSYSTM1
SQLDBA3  SQLMACH3 MSTRSRV1 VMSYSTM1
SQLDBA4  SQLMACH4 MSTRSRV1 VMSYSTM1
SQLDBA5  SQLMACH5 MSTRSRV2 VMSYSTM2
```

Figure 36. Sample SQLMSTR DIRECTRY File

If multiple service machines are being used at the installation, then manual updates to each SQLMSTR DIRECTRY file will be required to include entries within each file for databases that are not defined to the corresponding service machine. For example, if you have two service machines, one on node X and one on node Y, manually add the databases on node Y to the SQLMSTR DIRECTRY file on node X. Also manually add the databases on node X to the SQLMSTR DIRECTRY file on node Y. A single common SQLMSTR DIRECTRY file should contain entries for all databases accessible within the enterprise and should be made available to users through a common disk. Another example is where several service machines share the code disk of another service machine. In this case, the code disk of the owning service machine should be updated to reflect all databases of all service machines. The disk should then be relinked by each service machine to pick up the new copy of the file.

The following example shows a SQLMSTR DIRECTRY file that contains entries for two databases. Both databases have the same machine name but reside on different nodes. In order to identify each database, we needed to give them a nickname (SQLDBA1 and SQLDBA2). The first database exists on VMSYSTM1 and communicates through the MSTRSRV1 machine on that node. The second

database exists on VMSYSTM2 and communicates through the MSTRSRV2 machine, also located on VMSYSTM2. A database SHOW command could be issued for the first database in command mode as:

```
SQM SQLDBA1 SHOW USERS
```

```
SQLDBA1  SQLMACH1 MSTRSRV1  VMSYSTM1
SQLDBA2  SQLMACH1 MSTRSRV2  VMSYSTM2
```

Figure 37. Nickname Example with SQLMSTR DIRECTRY File

Operational Note: If the same database machine name exists on two different nodes, you must *always* use the nickname to direct communication to the correct database. In the previous example, if a command was issued to SQLMACH1, the results would be unpredictable. If you had previously been communicating with the database on VMSYSTM2, then Control Center would direct your new command to the same node. If you were previously communicating with a third (different) database and directed a command to SQLMACH1, the command cannot be directed to your desired database. The best policy is to always qualify the database name with the node ID or use the established unique nickname.

Maintaining the DBINIT CONTROL File

The DBINIT CONTROL file is used to link and access the appropriate 195 minidisk when using a Database Administration tool.

This file can be manually updated using Xedit or automatically updated using appropriate options (either **New Database Setup** or **Database Parameters**).

Each database is described in terms of:

- Database name
- Database server production disk name or SFS directory names
- Address of minidisk with database server production code, or value "SFS", if SFS is used
- User's link address (disk address at user machine) (optional)
- User's link filemode (user's access filemode) (optional)
- RXSQL Code Disk owner, or SFS directory names if SFS is used (optional)
- Address of minidisk with RXSQL production code, or "SFS" if SFS is used (optional)
- Link address for RXSQL (disk address at user machine) (optional)
- Link filemode for RXSQL (user's access filemode) (optional)
- Optional nickname (VM MACHINE NAME or ID)

This file will be updated when a new database is identified to Control Center. If the machine on which the database is identified owns the code disk, the file will be replaced on the code disk. Otherwise, the updated (merged) file is sent back to the user who defined the database with instructions to send the file to the service machine which owns the code disk, so that the file can be properly replaced. This must be done by a Control Center Administrator. The file will require manual updating by the DBA to include RXSQL Control Center disk information if desired for user availability. This file should be made available to all users who will be executing the Database Administration tools of Control Center.

```
SQLDBA1  SQLDBA1  195 5FF E RXSQL22 198 * *
SQLDBA2  SQLDBA2  195 5FF E RXSQL22 198 * *
SQLDBA3  SQLMACH3 195 *   * RXSQL31 198 * * SQLMACH3
```

Figure 38. Sample DBINIT CONTROL File

```
DB261VM  SFSPOOL1:DB261VM.SQL.PRODUCTION SFS * * * * * * *
SQLDBA   SQLMACH  195 * * * * * * SQLMACH
DB2PROD  DB2PROD  195 * * SFSPOOL1:DB261VM.SQL.RXSQL SFS * * *
```

Figure 39. Sample DBINIT CONTROL File Using SFS

With this information, the Database Administration tools of Control Center will be able to automatically link and access the correct database server production disks and RXSQL code disks (if available) for each database. This is especially important for certain functions, such as migrating data between databases. The user ID must also have authority to LINK and ACCESS the indicated disks in READ mode without supplying a READ password. This must be attended to according to your installation authorization procedures.

Shared File System (SFS) Support

On a VM/ESA system, DB2 Server for VM can be installed in a CMS Shared File System (SFS) environment. With SFS, the database service and production files are located in directories rather than on minidisks. To access code within an SFS file structure, the appropriate directory names together with a file mode must be used. In the following example, the *filepoolid*, SFSPOOL1, contains the production and service code for SQLMACH.

```
SFSPOOL1:SQLMACH.SQL.SERVICE
SFSPOOL1:SQLMACH.SQL.PRODUCTION
```

There are two ways to set up a database with SFS:

- From the Database Setup function, select option **N** on the Main Menu, followed by option **PF4**, or
- Through the Database Parameters function, select option **P** followed by option **U**, and scroll forward to the Utility Parameters menu.

When the Database Name is Not the Virtual Machine Name

If your database name is **not** the same as the virtual machine name, be sure to follow the format shown in Table 8 on page 125 for the DBINIT Control file.

You must also specify the database name in the *dbname* field of the Database Startup Parameters file, see “Database Startup Parameters” on page 222. In addition, make sure that the *dbmachid* (virtual machine) name is designated as the filename for both the database PARMS and TAPES files.

Table 8. DBINIT Control File Format

Field	Description
1	*DBNAME* or *dbmachid*. If the *dbmachid* or virtual machine ID is different from the *dbname*, put the *dbname* in this field, then be sure to enter the *dbmachid* in *Field 10*, or *Field 6*, if RXSQL is not used with the database.
2	*DBMACHID* that owns the database production code. If the production code is on a shared disk, the *dbmachid* VM directory should contain a LINK statement to that disk (*LINK PRODBA 195 195 RR*). Then use the *dbmachid* as the disk owner. If SFS is used, specify the appropriate directory name.
3	Disk address where the production code resides (195), or "SFS", if SFS is used.
4	Link address used for the production code (optional field). If none, enter an asterisk (*).
5	Filemode used to access production code (optional field). If none, enter an asterisk (*).
6	Virtual machine ID that owns the RXSQL code disk or SFS directory name if SFS is used (optional field). If none, enter an asterisk (*). This field can also be used to specify the *dbmachid*, if RXSQL is not used with the database, and the *dbname* is different from the *dbmachid*.
7	Disk address where RXSQL code resides, or the value "SFS", if SFS is used (optional field). If none, enter an asterisk (*).
8	Link address used for the RXSQL code (optional field). If none, enter an asterisk (*).
9	Filemode used to access RXSQL code (optional field). If none, enter an asterisk (*).
10	*DBMACHID* or virtual machine ID (nickname) when *dbname* is different from the *dbmachid* (optional field).

Please note from Table 8 above, to allow DBAs to use either VM machine names (IDs) or database server names with various Control Center tools, the DBINIT CONTROL file MUST contain the correct entries for those databases which have different names from the VM machine id. The "optional nickname" field MUST be the database VM machine id; (SQLMACH3). If there are no RXSQL parameters, the "optional nickname" parameter can be the 6th parameter.

Maintaining the DATABASE MESSAGES File

The DATABASE MESSAGES file contains message identifiers that will cause the service machine to perform a predefined function when those messages are received from a database machine through the SCIF communication interface. Each message that needs to be trapped for the normal database functions (such as archiving, recovery and adding dbspaces) is included with the Control Center installation package.

The DATABASE MESSAGES file is the single point of control for all database communications and automated operations functions. It provides a list of message IDs (ARI0043I) that the service machine is programmed to expect from a database machine. Each message ID within the DATABASE MESSAGES file has a corresponding name of a module that will be executed by the service machine when the message is received from any database machine.

Figure 40 on page 126 is an abbreviated example of the DATABASE MESSAGES file. The file supplied with the Control Center code provides all entries needed for supported database functions.

Important:

An experienced user can insert additional entries for messages not currently included, but existing entries *must not* be changed.

```
ARI0016I     SQMA015I I     Database is starting up (parameters)
ARI0025I     SQMA025I I     Database is starting up
ARI0029I     SQMA029I I     Still N active communication links
ARI0043I     SQMA043I I     The database came down
ARI0045I     SQMA045I I     The database just started
DMSTLM428I   SQMD428I I     EOV tape mark encountered/Tape filled
HCPCFC003E   SQMDUMMY I     Unknown CP command option
HCPQCS150A   SQMH150A I     Database machine issued a VM or CP read
HCPSEC068E   SQMUDONE I     End of Message from database OP command (ERROR)
LOGOFF       SQMDBDWN I     Database machine was logged off
```

Figure 40. Sample DATABASE MESSAGES File

An installation can want to add new messages and action routines to the DATABASE MESSAGES file for some function that is not included with Control Center. The first step would be to create an exec (or module) that will be invoked when the message is received by the service machine. The database machine name and message will be passed to the exec as arguments when the exec is invoked by the service machine. This exec should be placed on the service machine's 195 code disk. The second step is to add a new line to the DATABASE MESSAGES file on the service machine. Do this by stopping the service machine, log onto it, update the file, then start the machine back up. This can also be done using the Control Center command mode interface with the MESSAGE option. See “SQM: Control Center” on page 536.

The format of each line of the DATABASE MESSAGES file is:

```
POSITION   VALUE    DESCRIPTION
--------   -----    ----------------------------------------------------
1-12       Msgid    A message identifier that will be the first token
                    (blank-delimited word) of the message received

14-21      Exec     An executable module name (may be an exec or module)

23         Msgtype  A one-character indication of the type of message,
                    I for Information, D for DB status, E for Error,
                    S for Severe Error

29-255     Descr    Any text description of this message and action
```

Figure 41. Database Message Format

Maintaining the SQLMSTR HOLIDAYS File

The SQLMSTR HOLIDAYS file is an optional component of the Job Scheduling tool. It allows an installation to define special days that should not be considered by Control Center for scheduled job execution. For example, if the local installation is closed for an entire week in July, Control Center can be instructed to avoid scheduling jobs on any of these days. For each day that Control Center should avoid, an alternate day must be provided, which the Job Scheduler will use instead.

See “SQLMSTR HOLIDAYS File” on page 183 for a detailed explanation of the SQLMSTR HOLIDAYS file setup and processing. A sample file is shown in Figure 42 on page 127.

```
19970101 19970102 * New Year's Day
19970404 19970406 * Good Friday
19970407 19970406 * Easter
19970526 19970527 * Memorial Day
19970704 19970705 * Independence Day
19970901 19970909 * Labor Day
19971127 19971126 * Thanksgiving
19971128 19971126 * Thanksgiving day 2
19971225 19971218 * Christmas
19980101 19980102 * New Year's Day
```

Figure 42. Sample SQLMSTR HOLIDAYS File

Maintaining the SQLMSTR TIMES File

The SQLMSTR TIMES file is initialized with three entries for a service machine and one entry for a support machine. These entries must not be changed. Other entries are dynamically added when jobs are scheduled or monitors are added, as well as when other internal service machine events need to be scheduled. In general, you should not need to modify this file. Most changes can be made through the job scheduling and monitor panels. If needed, you can alter entries or delete them if you have a good knowledge of the effects of doing so. For example, you might change an event scheduled for a certain day to another day, or to another time on that same day.

Database TAPES File

Each database must have a TAPES file to allow Control Center to initiate and manage database archive activity. The filename is equal to the database virtual machine name; the filetype is TAPES.

The database TAPES file consists of a number of entries dealing with the archive process. This file is used to set up archiving for each particular database. It does not imply that tapes will be the medium for all archiving. The file is updated using the Database Tapes Menu option on the Control Center main menu. See Chapter 15, “Tape Management Tool” on page 185 for a detailed description of tape file setup and management.

The most significant parameter in this file is the SERIES parameter. This parameter is used by the service machine to set up different sets of archive copies. If it is useful to have three different sets of restorable copies of the database, the database TAPES file will reflect that by having three different values for the series parameter. Each series describes a set of tapes or files, which as a total, restores the database to a given point in time.

```
100  ARCHIVE  00000  00:00:00 FILDEF VOL100
100  LOG      00000  00:00:00 UNUSED SQL31DB 03269101 F 501
100  LOG      00000  00:00:00 UNUSED SQL31DB 03269102 F 501
100  LOG      00000  00:00:00 UNUSED SQL31DB 03279102 F 501
100  TRACE    00000  00:00:00 FILDEF VOLT01
200  ARCHIVE  97086  11:09:58 FILLED VOL200
200  LOG      97086  11:34:56 FILLED SQL31DB 03279103 G 502
200  LOG      00000  00:00:00 UNUSED SQL31DB 03269105 G 502
200  LOG      00000  00:00:00 UNUSED SQL31DB 03279101 G 502
200  TRACE    00000  00:00:00 UNUSED VOLT02
300  ARCHIVE  97086  11:49:02 FILLED VOL300
300  LOG      00000  00:00:00 UNUSED SQL31DB 03269106 H 503
300  LOG      00000  00:00:00 UNUSED SQL31DB 25039107 H 503
300  LOG      00000  00:00:00 UNUSED SQL31DB 25039108 H 503
300  TRACE    00000  00:00:00 UNUSED VOLT03
```

Figure 43. Sample Database TAPES File

Figure 43 is an example of a database TAPES file with three different series of archives. In this example, database archives are directed to tape, log archives to disk, and trace data to tape. The database has log archiving enabled. Archive series 200 is completed. The database archive for series 300 is also completed, but no log archive has been taken yet. That means the next log archive will be in series 300, because in that series the latest archive has been taken, to which the new log archive logically belongs for recoverability.

The FILEDEF for this log archive is not mentioned in this file because it can be done dynamically when the log archive takes place (see database PARMS file). If another database archive is taken, it will be directed to series 100, which was FILEDEFed when the database was started. The system will update this file (and the database PARMS file) every time an archive is taken.

Database PARMS File

Each database has a database PARMS file. The filename is equal to the database virtual machine name; the filetype is PARMS. This file describes many characteristics of the particular database. The general types of parameters within this file include:

1. Authorization/Notification parameters
2. Utility parameters
3. Tape parameters
4. Archiving parameters
5. Database Startup parameters
6. Data Restore parameters
7. VMDSS parameters

See “About the Database Parameters Tool” on page 213 for a complete description of these parameters.

The first section, Authorization/Notification, provides a list of users authorized to issue database manager commands for the database and additional users who will be notified of various database errors and status changes.

The second section defines some operational parameters and preferences which are used during database operation. For example: Do you want an automatic

archive to be performed after adding a DBEXTENT? If your preference is YES, the Archive_addextent parameter will equal Y.

The third section provides information about tape mounting and tape handling for the database, such as tape density, retention period, and whether scratch tapes or predefined tapes should be used.

The fourth section provides the link with the database TAPES file (described above). This section is also initially filled during the installation of Control Center, but will be maintained by Control Center when an archive is taken.

In Figure 44 on page 130, the archive_series parameter is set to 100, which matches the FILEDEF for the database archive in the database TAPES file. The Logarch_series is set to 300 and that matches the latest full database archive.

Important:

In the initial setup it is very important that the Archive_series parameter has a different value than the Logarch_series parameter.

The fifth section of the PARMS file consists of the database startup parameters, which can be changed by the Administrators and DBAs when the database is up. They become effective the next time the database is started.

The sixth section of the PARMS file consists of parameters that must be specified for Data Restore functions such as BACKUP, TRANSLATE, RESTORE, UNLOAD, and RELOAD. These new parameters are described in further detail in “Data Restore Parameters in the Database PARMS File” on page 223.

```
*********************************************************************
*                     SQLDBA PARMS (Sample)                         *
*********************************************************************
:Nick.SQLDBA
   :Version.6.1.0
   :Administrators.DBA1 DBA2 AT NODE3
   :Operators.OPER1 AT NODE2 OPER2
   :Users.USER1 AT NODE2 USER2
   :Notify_error.DBA1 DBA2
   :Notify_severe.DBA1
   :Notify_dbstatus.DBA1
   :Notify_operator.

*********************************************************************
***   Utility Parameters                                          ***
   :S_Disk_Userid_Directory.SFSPOOL1:SQLDBA.SQL.SERVICE
   :S_Disk_Address_or_SFS.SFS
   :S_Disk_Userid_Directory_filemode.V
   :P_Disk_Userid_Directory.SFSPOOL1:SQLDBA.SQL.PRODUCTION
   :P_Disk_Address_or_SFS.SFS
   :P_Disk_Userid_Directory_filemode.Q
   :Archive_addspace.N
   :Archive_addextent.N
   :Uarchive_enabled.N
   :Submit_routine.
   :Cancel_routine.
   :Spool_console_option.N
   :Sqlend_quick.N

*********************************************************************
***   Tape parameters                                             ***
   :Tape_density.38K
   :Scratch_tape_option.N
   :Tape_retention.365
   :Tape_archive_dsn.IU.SQLDBA.ARCHIVE
   :Tape_log_dsn.IU.SQLDBA.LARCH
   :Tape_trace_dsn.
   :Scratch_pool.SPSQLDBA
   :Logtape_premount.N

*********************************************************************
***   Archiving/Tracing parameters                                ***
   :Archive_media.TAPE
   :Archive_blksize.28672
   :Archive_series.100
   :Logarch_media.TAPE
   :Logarch_blksize.28672
   :Logarch_series.400
   :Trace_media.DISK
   :Trace_blksize.4096
   :Trace_series.400

*********************************************************************
***   STARTUP parameters                                          ***
   :Dbname.SQLDBA
   :Dcssid.
   :Sumdcssid.
   :Sysmode.M
   :Dbmode.
   :Logmode.L
   :Startup.W
   :Ncusers.8
   :Npagbuf.500
   :Ndirbuf.500
   :Nlrbu.
   :Nlrbs.
```

Figure 44 (Part 1 of 2). Sample Database PARMS File

```
   :Dispbias.
   :Ncscans.
   :Chkintvl.
   :Slogcush.
   :Archpct.75
   :Soslevel.
   :Charname.
   :Ltimeout.
   :Account.D
   :Dumptype.
   :Extend.
   :Syncpnt.
   :Tracebuf.10
   :Tracdbss.00010000100
   :Tracrds.020000
   :Tracdsc.00
   :Tracconv.0
   :Tracwum.0
   :Tracdrrm.0000
   :Tracstg.1
   :Progname.
   :Tracing.OFF
   :Amode.
   :Protocol.
   :Npackage.
   :Npackpct.
   :Tcpport.
   :Tcpportr.
   :Sectype.
   :Secalver.
   :Ptimeout.
   :Procmxab.

*********************************************************************
***   VMDSS parameters                                            ***
   :Mapping.
   :Saveintv.
   :Sepintdb.
   :Targetws.

*********************************************************************
***   Data Restore parameters                                     ***
   :Drmstr_enabled.N
   :Data_restore_machine.
   :Drtape_premount.N
   :Data_restore_lang.S001
   :Dual_backup.N
   :Backup_wrksize.2048
   :Backup2_media.
   :Backup2_scratch_tape.
   :Incbackup_enabled.
   :Incbackup_series.
   :Auto_full.
   :Current_increference.
   :Translate_media.
   :Translate_scratch_tape.
   :Tape_translate_dsn.
   :Tape_unload_dsn.
   :Tape_backup_dsn.
   :Tape_backup2_dsn.
```

Figure 44 (Part 2 of 2). Sample Database PARMS File

This file is updated by Control Center or by an administrator. When a database archive occurs, the series parameters for archives and log archives will be updated by the service machine. Startup parameters will normally be changed by authorized users (Administrator, DBA).

Changing the DATE/TIME Format

All screen displays and Control Center reports now use a full 4 digit year. In addition, you can choose from four different DATE/TIME display formats - ISO, JIS, USA, and EUR.

A format is chosen at Control Center installation, or migration. If none is chosen, USA is the default.

To change the DATE/TIME format:

1. Select Option **G** (General Control Center Commands).
2. Select Option **S** (View/Update SQLMSTR CONTROL FILE). The new field (Date-Time-format) will be presented on the menu.
3. Change this field to any valid option and press ENTER to process the change.
4. Then choose Option **N** (NEWPROF) to restart Control Center. You will receive a Control Center message as a result of issuing the NEWPROF command.
5. Next choose Option **DT** to reset your own date-time format. The new format will then be displayed.

Problem Resolution

Due to the asynchronous operation of Control Center, it is sometimes difficult to determine the cause of problems when they occur. When it is necessary to debug a problem, there are several sources of information available.

Each service machine keeps a running log on the 191 A-disk of every interrupt that occurs. This log can be viewed by linking the service machine's 191 A-disk in READ mode and using XEDIT or BROWSE to examine the current SQLMSTR LOG*yyddd* file (where *yyddd* is the Julian format date; *yy* is the year and *ddd* is the day of the year).

This file will show the sequence of interrupts that have occurred on the service machine for the day indicated by the Julian date. Control Center will keep logs of prior days' activities up to the number specified by the Log_copies parameter of the SQLMSTR CONTROL file.

A more detailed source of information is available in the SQLMSTR CONSOLE file of the service machine. This spool file records every screen I/O that would normally be displayed on a terminal (console) if the machine wasn't running in disconnected mode. By viewing the SQLMSTR CONSOLE file, you will be able to see everything that you would have seen if you were logged onto the service machine during the entire time period covered by the console file.

Similar to the SQLMSTR LOG*yyddd* file, the SQLMSTR CONSOLE file is normally kept on a daily basis. Each night at midnight, the console will be closed and the file transferred to the user ID specified on the OWNER statement in the SQLMSTR PROFILE file. The OWNER will find a new SQLMSTR CONSOLE file in their virtual reader each day, where it can be examined for possible errors and purged as desired.

At any given time, a user with Control Center's Administrator authority can have the current SQLMSTR CONSOLE closed and transferred to the requestor for

debugging purposes. This is done in Control Center's command mode by using the SQM CONSOLE command, or in panel mode by selecting the CONSOLE CLOSE (CO) option under the General SQLMASTR commands (G) option on the Control Center main menu.

For problems related to a specific database, the machine's 191 A-disk will contain several files which can help in debugging. Each file will have a filename of the database machine name. The filetype will differ depending on the information within it. There is an SQLSTART file which is the CONSOLE output from the database machine. This file will always be one SQLSTART behind the current session, being closed and sent to SQLMSTR when the database is terminated.

Each single user mode activity will also be recorded in a separate file on the machine's 191 A-disk.:

- The last database archive will be recorded in a file with a filetype of ARCHLOG.
- The last database recovery will be contained in a file with a filetype of RECOVLOG.
- The previous execution of the ADD DBEXTENT function will be recorded in a file with the filetype ADBEXLOG.
- The DELETE DBEXTENT information will have a filetype of DBEXLOG.
- The last ADD DBSPACE execution will be logged in a file with filetype ADDSPLOG.
- The last COLDLOG will have a filetype of COLDLOG.
- The previous execution of a Catalog Index reorganization will be recorded in a file with a filetype of SQLCIREO.
- A Single User Mode Reorganization will be recorded in a filetype of SQMREORG.

The most common problems encountered with Control Center are related to the communication interface between the Control Center virtual machine and the database virtual machine. When the SCIF interface is not set up properly, then the Control Center virtual machine does not receive the database messages that are necessary for correct service machine operation. This will further result in the database not receiving the commands from Control Center to perform the expected operations.

The SCIF interface is tested during database startup and an error message will be sent to Control Center users if the interface is not operational. When this type of problem occurs, VM and SCIF expertise will be required to debug the cause and provide a solution.

Chapter 12. Getting Started

This chapter provides you with enough information to get you started in using Control Center in your environment. You will be introduced to the Control Center panel interface, the Control Center Main Menu, *list* panels and *response* panels, and we will show you how to tell Control Center which DB2 Server for VM database you want to work with.

After completing this chapter you'll be all set to start learning and using the Control Center tool set described in the remaining chapters of this book.

Using Control Center

To use Control Center you *must* first link and access the Control Center code disk in READ mode with an available virtual device address (cuu) and access mode.

Starting a Session

Control Center can be invoked in panel mode by typing "SQM" from CMS and pressing ENTER. To start the service machine in command mode, type "SQM" along with one or more command arguments. If you are a first-time user, use panel mode, which provides you with an interface that guides and assists you throughout all of the Control Center tools and functions.

Even if you are not a first-time user, you are likely to find that panel mode is the easiest and fastest way for you to do the work you need to do. Command mode is available and in some situations may be the better approach. Nevertheless, we have decided to leave the details concerning command mode operation to Appendix G, “Command Mode Interface” on page 523.

Control Center Main Menu

After you have started Control Center, you will see the Control Center Main Menu, shown in Figure 45.

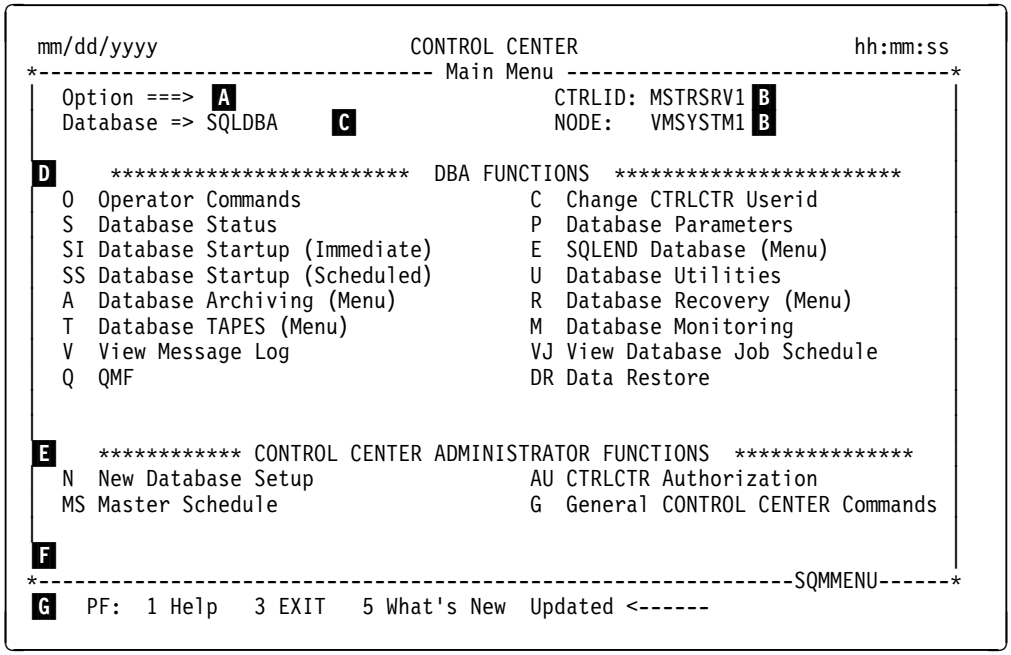


Figure 45. Control Center Main Menu

Some things you should know about the main menu are:

A The Option Line— You invoke options by entering the option abbreviation on this line and pressing the ENTER key. The option abbreviation is the one to two-character identifier highlighted to the left of the option description. They can be entered in upper or lower case. Some options include one or more parameters which you may need to provide prior to invoking (none are shown on the Main Menu). The option description will identify any available parameters. Valid choices for a parameter are separated by a vertical bar (|). Parameters shown in uppercase are to be entered as is. Those shown in lowercase are for you to supply.

In addition to the Control Center options, you can also invoke any valid CMS command from the option line by preceding the command with CMS.

B CTRLID & NODE—These fields identify the VM system ID and node, respectively, of a service machine in your environment.[4]

A communication path is established with this service machine enabling you to work with any of the databases that it manages, assuming you have the proper authority.

Note: Prior to invoking any of the Control Center tools you must first establish a communication path to a service machine in your environment. Refer to “Control Center Communication Path Setting” on page 145 for additional information on establishing and changing your communication path.

[4] Your local Control Center administrator can provide you with this information.

C Database Entry Field—Enter the name or the defined nickname of the database machine you want to work with in this field.[5]

Refer to “Control Center Database Setting” on page 145 for additional information.

D DBA Functions—This area of the main menu lists the Control Center functions available to a database administrator. Most functions invoke Control Center tools which require you to have the proper authority against the selected database.

E Administrator Functions—This area of the main menu lists the Control Center functions available to an administrator. Administrator functions pertain to the operation of Control Center itself, and require the highest level of Control Center authorization to execute.

F The Message Line—Control Center displays information and error messages on this line.

G Program Function (PF) Keys—PFkeys are always shown at the bottom of the panels. What each key does depends on the panel being displayed. However, you can always count on PF1 to provide you with help for the displayed panel and PF3 to back you out to the previous panel.

Other Type of Panels

While using the Control Center you'll encounter numerous *menu* panels like the main menu. Except for the available options, PF keys, and input fields, they all operate in a similar manner. Other types of panels you are likely to encounter include *list* panels and *response* panels. These are discussed in the sections “List Panels” on page 145 and “Response Panels” on page 146.

Invoking QMF

When Q is entered from the main menu, Control Center will initialize to the database named in the Database option field. A check is made to see if the Q.Object_Data table exists, to determine if QMF is installed on the database. If the table is present, Control Center will will invoke the SQMQMF EXEC. If not, the main menu and an error message are displayed.

These steps must be completed before invocation can take place:

1. Ensure that QMF is installed in your environment.
2. Logon to your Control Center machine.
3. Copy SQMQMF $EXEC to your A disk and change filetype to EXEC.
4. Tailor SQMQMF EXEC to contain your specific QMF invocation exec(s). This may be the DSQ2EINV EXEC or a similar routine to invoke QMF.
5. Copy SQMQMF EXEC back to the Control Center code disk.
6. Restart Control Center.

The following figure is a part of the SQMQMF $EXEC which should be modified to fit your environment.

---

[5] For a list of established database nicknames, select Option **G** (General Control Center commands) from the Control Center Main Menu, followed by Option **D** (View SQLMSTR DIRECTRY file), or consult with your local Control Center administrator.

```
/*====================================================================*/
/* SQMQMF $EXEC                                                       */
/*====================================================================*/
/*  MODIFY THE FOLLOWING SECTION for DATABASE DIFFERENCES in          */
/*  invoking QMF.  If there are no database differences, simply       */
/*  change as appropriate for your installation.                      */
/*                                                                    */
/*  If QMF needs to be linked, or in your own QMF interface exec      */
/*  needs to be called, modify the code accordingly.                  */
/*                                                                    */
/*  If the same exec is to be invoked for all databases, then         */
/*  simply uncomment the following instruction.                       */
/*                                                                    */
   parse upper arg dbname .
   dbname = strip(dbname)
   'EXEC &&&&&&&&'   /*  your local exec or QMF supplied one */
   erc = rc
   RETURN
/*                                                                    */
/* OTHERWISE modify the SELECT statement according to the database.   */
/*                                                                    */
   SELECT
    /*** Change for each database if necessary ***/
   when dbname = '&&&&&&&&'   /* specific database */
     then do
       'EXEC &&&&&&&&'    /* your local exec or QMF supplied one.    */
       erc = rc
     end
/*                                                                    */
   otherwise NOP                                                     */
   END /* SELECT */                                                  */
   RETURN                                                            */
/*====================================================================*/
```

Figure 46. Exec to Invoke QMF (SQMQMF $EXEC)

Note: If your installation uses VMLINK to access your QMF code disks, appropriate messages can briefly appear on your screen. You can avoid these messages by issuing an explicit link and access to the QMF code disk instead of using VMLINK.

Fast Path Navigation

The Control Center Fast Path facility allows you to more quickly and easily navigate the many Control Center menus and tools. You can:

- Move from one menu to another,
- Move from CMS to specific Control Center menus,
- Exit directly to CMS from within a Control Center menu.

When you first invoke Control Center from CMS with the fast path facility, the database name found in the user's LASTING GLOBALV file under the tag SQM DATABASE is used. At the same time, each utility initializes itself with the database name used during the last previous execution. Be careful that you do not assume which database will be affected when you use fast path to go directly to a low level utility screen. Either:

- Check the database name in the LASTING GLOBALV file before you invoke the utility, or
- Use the database name field on the utility panel to switch to the desired database, or

- Bring up the Control Center Main Menu, change the DBNAME field, then use fast path to go directly to the desired utility.

How to Use Fast Path

You can use fast path commands to proceed to a specific Control Center screen from another screen or from CMS. From the Control Center menu screens which supply a command line, enter an equal sign (=) followed by the tool or panel designator, and options (if known). For example:

```
=A.S.SA DVERIFY TRCPURGE
```

This example displays the Job Scheduler screen for scheduling a SQLEND ARCHIVE with the DVERIFY and TRCPURGE options as depicted in Figure 47. This illustrates the general rule that if a lowest level panel has associated parameters, you can specify them following the tool or panel designator.

```
 mm/dd/yyyy                      CONTROL CENTER                       hh:mm:ss
*------------------------------ Job Scheduler -----------------------------*
| Command ===>                                                CTRLID: SQMSTTS1 |
| Jobname        ==> ________                                  NODE:  WMAVM1   |
| Job-status         ==> S    (S=Scheduled, A=Active, F=Failed, I=Inactive)  |
| Priority           ==> 5    ( 1 through 9, 1 = HIGH, 9 = LOW )             |
| Next-start         ==> 19970311 23:00   ( format: YYYYMMDD HH:MM )         |
| Window-end         ==> 19970312 08:00   ( format: YYYYMMDD HH:MM )         |
| Schedule-interval  ==> ONCE       ( ONCE or form nnnXX, where XX is:       |
|                                     MI, HR, DY, WK, WD, WE, MO, YR )      |
| Required-database  ==> SQL34DBA         ( Database machine-id )           |
| Required-dbstatus  ==> U                ( U=Up, D=Down, A=Any )           |
| Dependent-jobname  ==> ________         ( Prior job must run first )      |
| Dependjob-result   ==> _                ( S=Successful, F=Failed, A=Any ) |
| Average-runtime    ==> 01:00            ( format: HH:MM )                 |
| Last-jobstart      ==> _____________   Last-jobend  ==> _____________     |
| Notify   ==> ANDYS                                                        |
| Execute  ==> SQM SQL34DBA SQLEND ARCHIVE DVERIFY TRCPURGE                 |
|                                                                           |
|              Enter parameters and press PF6 to schedule                   |
|                                                                           |
*-------------------------------------------------------------SQMSCHDS----*
 PF: 1 Help   2 Detailed Help   3 End (Cancel)   6 Schedule
```

Figure 47. Scheduled Sqlend Archive Using Fast Path

To invoke a specific Control Center tool from CMS, enter SQM, 1 blank space, an "=" sign, followed by the tool/panel designator and any options.

```
SQM =O.A     Issue Operator Command, SHOW ACTIVE
SQM =U.R     Invoke SQLREORG menu
SQM =S.D     Display STATUS of your database
SQM =M.A     Add new Monitor
```

Where Fast Path is NOT Available

Some screens do not have a command line and therefore, do not support fast path navigation. You can in some cases, however, be provided with PF keys to allow you to exit to CMS or return directly to the Main Menu.

Fast Path Help and Hints

To use fast path, enter the equal sign (=) followed by the tool/panel designator in the OPTION field on the Main Menu and press ENTER. From there, you can build on the fast path command by specifying a period (.) followed by the designator that identifies the function displayed on that screen. Continue this until you reach the lowest level screen (=U.CL.S).

If you enter a fast path command without specifying a required parameter, the effect is the same as if you invoke the command through the panels. The panel is redisplayed with a highlighted error message indicating a missing or invalid parameter.

Figure 48 shows the primary tool designators you can use in fast path Navigation.

```
O  - Operator commands              C  - Change sqlmstr userid
S  - Status                         P  - Parameters
SI - Startup (Immediate)            E  - sqlEnd
SS - Startup (Scheduled)            U  - Utilities
A  - Archiving                      R  - Recovery
T  - Tapes                          M  - Monitoring
V  - View message log               VJ - View Job schedule
Q  - QMF                            VJ - View Job schedule
                                    DR - Data Restore
N  - New database setup             AU - control center AUthorization
MS - Master Schedule                G  - General control center commands
SL - Search/List                    MM - Main Menu
X  - eXit to CMS
```

Figure 48. Primary Fast Path Command Designators

Figure 49 on page 141 shows the second level or option designators following the tool to which they apply. Chain an option and its parameters to the tool designator to go directly to the screen you desire.

```
O - Operator Command Tools

  .A  Active                            .AD Address  module-name
  .B  Buffers                           .CN Connect
  .CO Counter *|name                    .CI Counter Internal *|name
  .CP Counter Pool *|n|DIR|UNMAPPED     .C  Dbconfig
  .E  Dbextent                          .D  Dbspace n
  .F  Force uid|AGENT n (DISABLE)       .ID Indoubt
  .I  Invalid                           .IP Initparm
  .L  Log
  .LA Lock Active                       .LD Lock Dbspace ALL|n
  .LG Lock Graph uid|AGENT n            .LH Loghist ALL|n|SERVICE
  .LM Lock Matrix                       .LN CRR Lognames
  .LU Lock User uid|AGENT n|ALL         .LW Lock Wantlock uid|AGENT n|ALL
  .P  Pool ALL|SUMMARY|DELETED|pool     .PO Stop Proc *|proc-name|auth
  .PR PROC *|proc-name AUTHID           .PS Start Proc *|proc-name authid
  .R  Reset *|name                      .RD Reset Indoubt *|ids
  .RI Reset Internal (Counter)          .RN Reset CRR Lognames
  .RP Reset Pool *|n|DIR|UNMAPPED
  .SI Set Parameter name newval
  .SP Set Pool n1|n1-n2 DSn STR|SEQ     .SS Set Saveintv n
  .ST Set Targetws n                    .WS Show Saveintv
  .WT Show Targetws                     .SG Storage
  .S  System                            .SP Pserver *|Group
  .TO Trace Off                         .TS Trace Start
  .U  Users                             .VO Stop Pserver group|name
  .VS Start pserver group|name

S - Database Status Tools

  .D Display status of Database
  .S Display status of all databases for Control Center machine
  .N Display status of all databases on Node
  .A Display status of all databases in SQLMSTR Directry

A - Archiving Tools (parms = DVERIFY, TRCPURGE)

  .I  Immediate Archive
      .SA Sqlend Archive  parms         .A  Archive
      .SL Sqlend Larchive parms         .L  Larchive
      .SU Sqlend Uarchive parms
      .BU BACKUP parms
      .BI BACKUP INCREMENTAL parms

  .S  Schedule Archive
      .SA Sqlend Archive  parms         .A  Archive
      .SL Sqlend Larchive parms         .L  Larchive
      .SU Sqlend Uarchive parms
      .BU BACKUP parms
      .BI BACKUP INCREMENTAL parms

  .C  Cancel archive                    .L  view previous archive Log
  .VJ View database Job schedule        .VT View Tape catalog
  .H  display archive History           .TM Tape catalog Maintenance
  .ST Acquire new SCRATCH TAPE and add to tapes file (VMTAPE or DYNAMT)

T - Tapes Tools

  .V View tape catalog                  .M Make changes to tape catalog
  .S Acquire new scratch tape and add to tapes file (VMTAPE or DYNAMT)

P - Parameters Tools

  .V  View parameters                   .U  Update parameters
  .S  display last used Startup parameters
  .VS View Storage pool specification file (VMDSS)
  .US Update Storage pool specification file (VMDSS)
```

Figure 49 (Part 1 of 3). Secondary Fast Path Command Designators

```
E - Sqlend Tools (parms = DVERIFY TRCPURGE)

  .N  sqlend Normal parms              .S  Schedule sqlend parms
  .Q  sqlend Quick                     .SQ Schedule sqlend Quick

U - Database UTILITY Tools

  .E  add dbExtents                    .A  Add Dbspaces
  .D  Delete dbextents                 .L  List Files
  .RD Reorg Driver tool                .QE Query Extents
  .RI Reorg Index tool                 .RS Reorg Single User
  .SL Search/List                      .M  SQLMAINT
  .B  sqlrBind                         .R  SQLREORG
  .T  sqlTable                         .E  data Extent
  .C  Copy/move dbextents              .CL Database Coldlog
  .D  Directory disk                   .I  Initiate Coldlog
  .L  Log disk                         .S  Schedule Coldlog
  .CO COmmand (CP DET 181)             .RC Reorg Catalog indexes
      .I Immediate command                 .I Initiate sqlcireo
      .S Scheduled command                 .S Schedule sqlcireo

R - Database RECOVERY Tools

  .I  Immediate                        .VJ View database Job schedule
  .S  Schedule                         .H  display archive History
  .C  Cancel                           .L  view Log of previous recovery
  .R  Restart after recovery failure   .VT View Tape catalog
  .TM Tape catalog Maintenance
  .ST acquire new Scratch Tape and add to tapes file (DYNAM/T and VMTAPE)

M - Database MONITORING Tools

  .A ADD a new monitor                .P PURGE (refresh) a monitor report
  .D DELETE a monitor                 .R Display REPORT for a monitor
  .L LIST monitors for this database  .S Display SCHEDULE for monitor(s)
  .M MODIFY an existing monitor       .V View (display) a monitor

Q - Query Management Facility (QMF)

MS - Master SCHEDULE Tools

  .VJ View scheduled Jobs              .VE View Events in SQLMSTR TIMES
  .SJ Schedule a Job                   .SE Schedule an Event in TIMES

DR - Data Restore Tools

  .T  TRANSLATE ARCHIVE                .VJ VIEW JOB SCHEDULE
  .U  UNLOAD DBSPACES                  .S  VIEW DRMACH STATUS
  .R  RELOAD TABLES                    .SR RESET DRMACH STATUS
  .LL LISTLOG                          .D  SHOWDBS
  .AL APPLYLOG

AU - SQLMSTR Authorization

  .A Add                               .D Delete
  .M Mum machine                       .L List
```

Figure 49 (Part 2 of 3). Secondary Fast Path Command Designators

```
G - GENERAL Control Center Commands

  .C  issue Command to CTRLCTR
  .CO close and display CTRLCTR COnsole
  .D  View SQLMSTR Directry file
  .F  FILELIST * * (Defaults to SQLMSTR LOG*)
  .I  Initialize SQLMSTR Control
  .N  NEWPROF (restart CTRLCTR, read SQLMSTR profile)
  .R  CTRNCTR Reader List
  .T  Terminate CTRLCTR
  .S  View/Update SQLMSTR Control File
  .SO Spool CTRLCTR console
  .SS Stop spooling CTRLCTR console
```

Figure 49 (Part 3 of 3). Secondary Fast Path Command Designators

Control Center Communication Path and Database Settings

Depending on your environment, you can have several service machines in operation on one or more CPUs. Each service machine can manage several DB2 Server for VM databases on that CPU. To work with a specific database, you must establish a communication path with the managing service machine and specify the database server machine.

Together, the database name specified in the Database entry field and your established communication path identify the database server you are working with. Figure 50 on page 144 and the following discussion help to illustrate this.

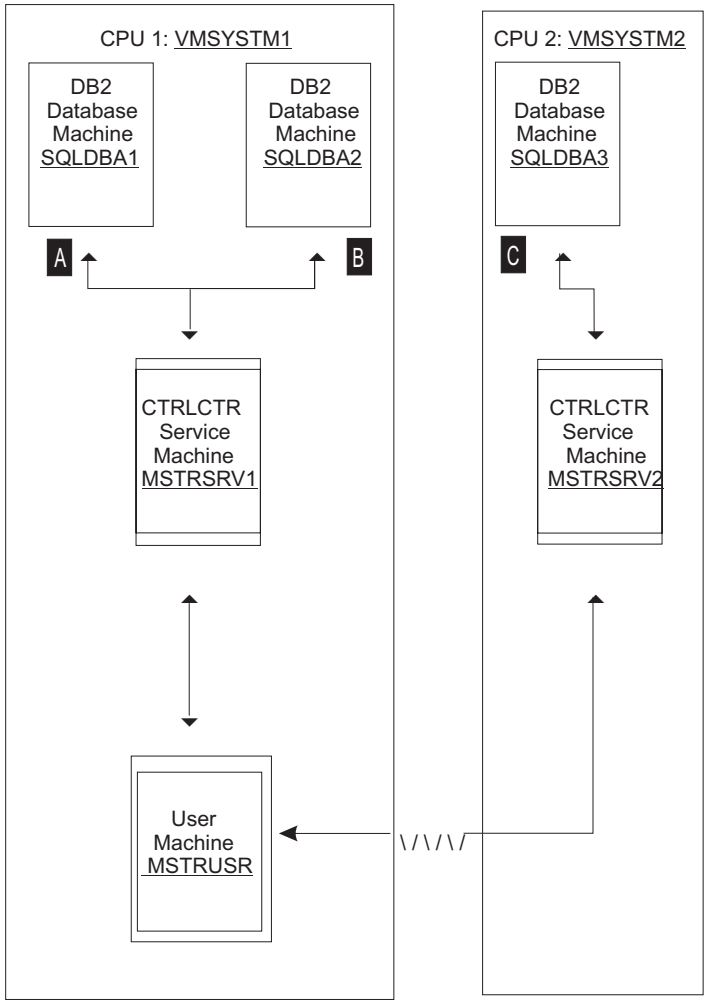


Figure 50. Control Center communication paths

Shown is a two-CPU environment, VMSYSTM1 and VMSYSTM2. Database manager machines SQLDBA1 and SQLDBA2 operate on VMSYSTM1 along with the Control Center service machine MSTRSRV1 and user machine MSTRUSR. The database manager machine SQLDBA3 and the Control Center service machine MSTRSRV2 operate on VMSYSTM2.

Table 9 shows the database field entry and communication path settings necessary to work with each of the databases from user machine MSTRUSR.

Table 9. Database field entries and communication path settings

To work with a Database:	Database Input Field Entry	Communication Path Settings	
		CTRLID	NODE
SQLDBA1	SQLDBA1	A MSTRSRV1	VMSYSTM1
SQLDBA2	SQLDBA2	B MSTRSRV1	VMSYSTM1
SQLDBA3	SQLDBA3	C MSTRSRV2	VMSYSTM2

Control Center Communication Path Setting

Prior to working with a specific database server, you must establish a communication path with the managing service machine. This can be done explicitly using Option **C**, Change CTRLCTR user ID, on the main menu, or implicitly by specifying a new database machine in the Database entry field. See "Control Center Database Setting." In the latter case, Control Center will automatically switch your communication path to the proper service machine.

Once established, your communication path will remain in effect (between sessions and system log-off/log-on) until explicitly or implicitly changed as described above.

Control Center Database Setting

A single service machine is capable of managing several databases. It is necessary for you to identify which database you want to work with by entering the database machine name or defined nickname in the Database entry field.[6]

The database name can be qualified with a node ID such as VMSYSTM1.SQLDBA1 to specify a particular CPU node if duplicate database machine names exist on different CPUs in your environment.

List Panels

Several Control Center tools present information to you in list format. Shown below is the DBSPACE List Utility, which is a part of the Object Search and List tool.

```
 mm/dd/yyyy                    CONTROL CENTER                          hh:mm:ss
*--------------------------- DBSPACE List Utility ----------------------------*
|  Database => SQLDBA                                                        |
|                                                                            |
| A SEL SPNO OWNER    NAME              POOL  PAGES    NACTIVE PIDX PCTF LOCK |
|  --- ---- -------- ------------------ ---- --------  ------- ---- ---- ---- |
| B _     3 PUBLIC   HELPTEXT             1     8192      -1   33   10     P |
|  _     4 PUBLIC   ISQL                  1     1024       3   33   15     T |
|  _     5 PUBLIC   SAMPLE                1      512       3   33    5     P |
|  _    11 PUBLIC   SQLMAINT              3      128       1   33   15     P |
|  _     1 PUBLIC   SYS0001               1    12800      58   60    0     T |
|  _     2 PUBLIC   SYS0002               1     2048      -1    0    1     P |
|                                                                            |
|                                                                            |
|                                                                            |
| C Select:  S = SHOW DBSPACE       L = SHOW LOCK        T = TABLE           |
|            R = REORG   A = ACQUIRE   U = UPDATE STATISTICS    D = DROP     |
| D                                                                          |
| E                             Page 1     of 1                              |
*----------------------------------------------------------------SQMDLST2-----*
 F PF:  1 Help         3 QUIT         4 StatHistory     7 Bkwd       8 Fwd
   9 Sort/Name   10 Sort/Spno   11 Sort/Pages    12 Sort/Pool
```

Figure 51. Control Center List Panel

6 For a list of established database nicknames, select Option **G** (General Control Center commands) from the Control Center Main Menu, followed by Option **D** (View SQLMSTR DIRECTRY file), or consult with your local Control Center administrator.

Some things you should know about *list* panels are:

A The SELect Column—It is here you enter options to be executed against specific rows in the list. Options are entered using the one to two-character option abbreviation highlighted to the left of the option description (see **C**). They can be entered in upper or lower case. Only one option can be entered per row; however, you can enter options on multiple rows. Options are executed sequentially, from top to bottom, when you press ENTER.

B The Item List—The Control Center displays in this area one or more rows of information. Headings identify each column. If more rows exist than can be shown on a single page (see **E**), then use PF7 and PF8 to page backward and forward, respectively.

C List Options—Shown here are the available options.

D The Message Line—Control Center displays information and error messages on this line.

E Page *x* of *y*—The current page ***x*** you're viewing and the total number of available pages ***y*** is shown on this line. Pages can be scrolled backward and forward using PF7 and PF8, respectively.

F Program Function (PF) Keys—PF keys are always shown at the bottom of the panels. What each key does depends on the panel being displayed. However, you can always count on PF1 to provide you with help for the displayed panel and PF3 to back you out to the previous panel.

Unique to *list* panels are "sort" PF keys which allow you to sort the rows by values in specific columns. For example, PF12 Sort/Pool sorts the list by the values in the POOL column.

Response Panels

Response panels display information sent to you by a service machine in response to an action or activity you initiated against a database. Responses are sent as files to your virtual reader that Control Center automatically displays using the CMS PEEK command. CMS PEEK places you in the XEDIT environment, editing the file *spoolid* PEEK A0. The full power of XEDIT is available to you while you peek the *response* file.

For additional information, refer to the PEEK command in the *VM/ESA: CMS Command Reference* manual.

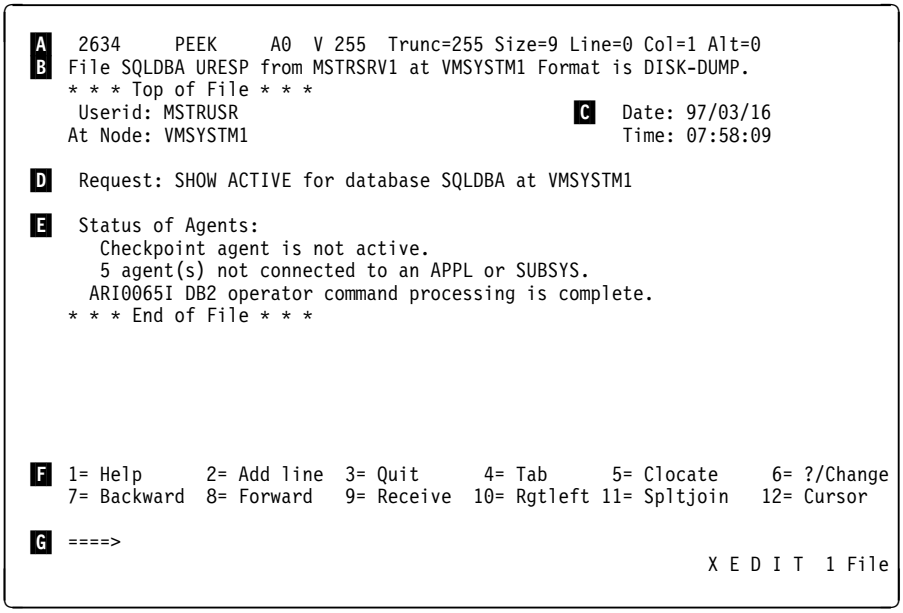


Figure 52. Control Center Response Panel

Some things you should know about *response* panels are:

A spoolid PEEK A0—The name given to the response file by the CMS PEEK command. See “Saving Response Information.”
B MSTRSRV1 at VMSYSTM1— ID and node of the service machine which generated and sent you the response file.
C Date and Time—When the response file was generated.
D Request Information—Your request, in detail, which elicited the response.
E Information Area—The requested information or, in the case of an error, error information.
F Program Function (PF) Keys—Refer to the PEEK command in the *VM/ESA: CMS Command Reference* manual.
G The Command Line—Refer to the PEEK command in the *VM/ESA: CMS Command Reference* manual.

Saving Response Information

When you quit the *response* panel using PF3, the *response* file is discarded from your virtual reader and is no longer available. You can elect to save the file to your disk in one of two ways. PF9 will receive the file to your A-disk using file name *spoolid* PEEK (see **A**). Or, you can enter the XEDIT FILE command followed by the filename, filetype and filemode on the command line.

Chapter 13. Group Authorization Tool

The Group Authorization Tool (SQLADMIN) assists DBAs in managing the access to database objects, simplifies the process of authorization, and shortens the amount of time needed to grant or revoke privileges. It gives DBAs the ability to issue authorizations to groups of users on groups of objects rather than one by one. Individual users can be associated with defined User Groups, and database objects (tables, views, and packages) can be associated with defined Application Groups. Then, GRANTs and REVOKEs can be issued specifying a User Group (grantee) and an Application Group (on objects) using the Group Authorization panel interface.

Installation Considerations

To use the Group Authorization Tool, an exec called SQMGAINS is supplied that installs five tables and creates the SQLADMIN ID in the database where the tool will be run. This exec must be submitted by an ID with DBA authority in the database. An SQLDBSU input file, SQMGAINS LOADBSU, used by SQMGAINS to create the tables and indexes, may need to be modified before execution. The steps to install are:

1. Inspect the SQMGAINS LOADBSU file. This file has the ACQUIRE statement for a PUBLIC DBSPACE of 256 pages in pool 1. The DBA must ensure that an unacquired DBSPACE is available prior to running the exec. The DBA may change any of the pertinent fields (POOL, NPAGES, DBSPACENAME).
2. Replace the changed LOADBSU file on product's code disk.
3. Run the SQMGAINS exec from a DBA ID linked to the database where the tool is to be installed.
4. Change the password of SQLADMIN. The DBSU grants DBA authority to SQLADMIN with a password of SQLADMIN. This password should be changed in accordance with your security procedures.

Group Authorization Tables

The five tables which SQLADMIN uses to manage the Group Authorization functions are listed below. Each one is owned by SQLADMIN. They are:

USERID_GROUP_TAB — User Group Table. This table is used to hold the name, internal ID, and description of a User group.

USERID_TAB — User ID Table. This table has one row for each unique combination of Userid and User Group ID.

APPL_GROUP_TAB — Application Group Table. This table is used to hold the name, internal ID, and description of an Application Group.

OBJECT_TAB — Object ID Table. This table has an entry for each object (object owner, object name) in each Application Group.

GROUP_AUTH_TAB — Group Authorization Table. This table records each group authorization made by SQLADMIN. It records the User Group ID, the Application Group ID and

the specific privileges granted by the SQLADMIN panel interface.

How it Works

The Group Authorization Tool is a stand-alone utility which executes as a user application.

All authorizations are granted by the connect ID "SQLADMIN", and are recorded and maintained by the tool.

By using the “LIST Functions” on page 152 the user can take advantage of the various reports that are available to help manage database access. These listings, or reports, will show Application Groups and the objects found within them (tables, views and packages); User Groups and user IDs associated with specific User Groups and authorities granted to User Groups.

Because all data about User and Application Groups, as well as authorization information, is kept in database tables (described in “Group Authorization Tables” on page 149), the DBA can also query these tables to obtain information about User and Application Groups and other authorization information. For example, if the DBA would like to see a list of objects belonging to an Application Group together with the specific authorizations on those objects, a query including a join statement could produce the desired result. Or, if the DBA chooses to modify a Group's description (see Figure 55 on page 157), an update could be made to the APPL_GROUP_TAB or USERID_GROUB_TAB table to make that change.

Tool Design

The Group Authorization tool is designed such that a user can belong to more than one User Group. A user can have the same privilege granted to an object through multiple User Groups. If the user is dropped from one User Group, the user will NOT lose the privilege to the object since it still has authority through another User Group.

The same logic applies to an object which is in more than one Application Group: if that object is dropped from one group, but privileges on it exist through another application group, those privileges will NOT be revoked from users who have authority on it through another group.

Options Available

ADD a Group

Allows the DBA to create both User and Application Groups. The data created by this function will be maintained in the USER_GROUP_TAB and APPL_GROUP_TAB tables respectively (see “Group Authorization Tables” on page 149).

DROP a Group

Deletes a group entry from the applicable tables (see “Group Authorization Tables” on page 149). In addition, if any privileges have been granted to a User Group which is being dropped, all privileges will be revoked from all users in that group. If an Application Group is dropped, all privileges which were granted on that group will be revoked from all users who were granted those privileges. There are two exceptions to this rule:

1. When dropping a User Group to which some group authorizations have been made, a check is made before revoking each user's privileges to determine that the user has not been granted the same privileges by virtue of belonging to another group which has been granted like privileges. This user will not lose its privileges in this case.
2. When dropping an Application Group which has had privileges granted on it, a check is made before revoking the privileges from each user to whom it was granted, to determine whether the privileges on the object have been granted to that user through another Application Group. The user will not lose its privileges in this case.

ADD/DROP GROUP OBJECTS or USERS

Permits the DBA to populate a group with user IDs (in the case of User Groups) or, tables and views, or packages (in the case of Application Groups). With Application Groups, the group is defined as either a table group (consisting of tables and views only), or a package group (consisting of package names only).

A user can belong to more than one User Group. An object can belong to more than one Application Group. Both group types could consist of only one entity.

When an object is added to an Application Group, if any User Groups have privileges on that group, those privileges will be GRANTED on the newly added object to all users in the User Groups authorized to that Application Group. When an object is dropped from an Application Group, all users holding privileges to that object will have them automatically REVOKED (see exceptions in “DROP a Group”).

When a user is added to a User Group, all authorities which the group currently has are GRANTED automatically to the added user. When a user is dropped from a group, all authorities which the group has are automatically REVOKED from the user (see exceptions in “DROP a Group”).

GRANT or REVOKE Privileges

Provides the capability of granting or revoking privileges to User Groups on individual database objects or on Application Groups; on all objects defined in the group. The capability of granting column UPDATE privileges is not implemented. However, you can create a view with the column updates and then grant UPDATE privilege on the view.

The Authorization panel allows you to enter an individual object or a group of objects on which privileges are to be granted or revoked. An individual object is identified by its owner and object name. A group object is identified by its Application Group name. All authorizations are done under the connect ID of SQLADMIN.

For example, in the SYSTEM.SYSTABAUTH table, the grantor for group authorizations will always be SQLADMIN. An extra grant is done for the User Group name to facilitate the implementation of the tool. If SELECT privilege is given to User Group UGROUP1 on Application Group AGROUP1, one extra GRANT SELECT to UGROUP1 on each object in AGROUP1 is generated by the tool. Likewise, this privilege is revoked from the User Group when the REVOKE function is used.

LIST Functions

These on-line listings, or reports, are available from within the Group Authorization Tool panel environment:

- All User Groups
- All Application Groups
- All Users/Objects within a specific group
- All Application Groups on which a given User Group has been granted privileges, and what those privileges are
- All User Groups to which privileges have been granted on a given Application Group, and what those privileges are.

Invocation

There are two ways to invoke the SQLADMIN tool:

- From the CMS READY screen, enter SQLADMIN, or
- Through the Control Center panel interface, using Option **U** (Database Utilities) on the Main Menu, select Option **GA**.

Once the tool has been invoked, various Group Authorization functions are available as Figure 53 shows.

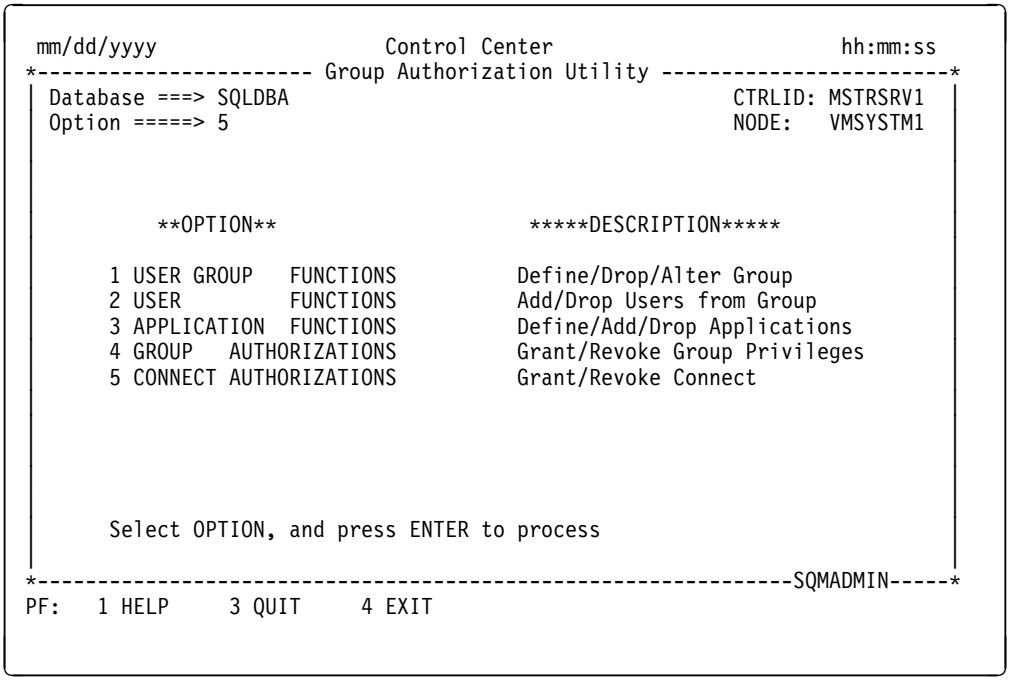


Figure 53. Control Center Group Authorization Utility Menu

Using the Tool

Using the Connect Authorizations Option For a Single ID

The Group Authorization tool is used to define individual user ids or groups of users to Control Center and grant them selected authorizations and access to specified objects. First we will discuss use of the tool for single ids and then its use in defining and authorizing groups.

When you specify 5 for the OPTION in panel SQMADMIN, a panel for CONNECT authorizations is presented. Figure 54 on page 154 is an example of the panel SQMADMCO.

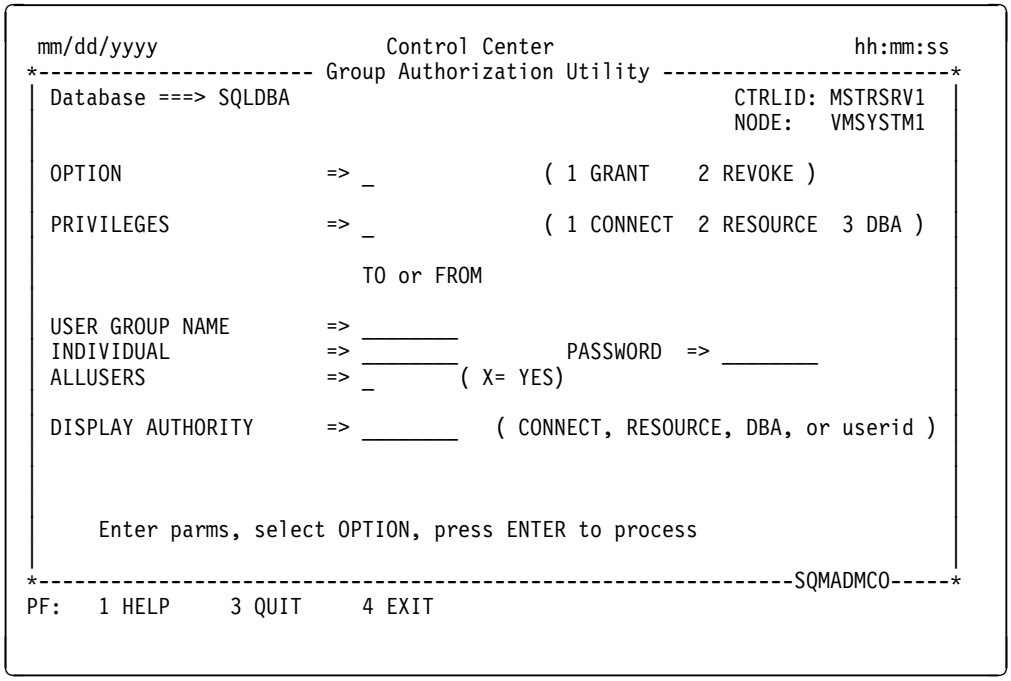


Figure 54. Group Authorization System Authorizations Menu

When using this panel for procedures related to a single user id, you do not use the USER GROUP NAME nor ALL USERS fields. All other functions are basically the same as when used with groups.

Menu Fields

OPTION
Used to specify the type of action to be performed with respect to the PRIVILEGES field. The values are:

- **1** GRANT
- **2** REVOKE

PRIVILEGES
Used to specify the type of authorization to be granted or revoked. The values are:

- **1** CONNECT
- **2** RESOURCE
- **3** DBA

USER GROUP NAME
This field is not used when working with a single user id.

INDIVIDUAL
Name of a specific user to whom to grant or revoke a privilege.

PASSWORD
The password to be used when granting the authority. Note that PASSWORD applies only to an individual id.

ALLUSERS
This field is not used when working with a single user id.

DISPLAY AUTHORITY

CONNECT This value is not used when working with a single user id.

RESOURCE This value is not used when working with a single user id.

DBA This value is not used when working with a single user id.

userid If an INDIVIDUAL id is entered, the id's highest level of authorization will be displayed immediately below the DISPLAY AUTHORITY field.

Using the Connect Authorization Option For Groups

When using this panel for procedures related to groups, you do not use the INDIVIDUAL field. All other functions are basically the same as when used with a single id.

Menu Fields

OPTION Used to specify the type of action to be performed with respect to the PRIVILEGES field. The values are:

- **1** GRANT
- **2** REVOKE

PRIVILEGES Used to specify the type of authorization to be granted or revoked. The values are:

- **1** CONNECT
- **2** RESOURCE
- **3** DBA

USER GROUP NAME
Name of a User Group, as defined to the Group Authorization tool. If a group name is entered, the authorization selected will be assigned to ALL members of the group.

INDIVIDUAL When working with a group, this field is not used.

PASSWORD When working with a group, this field is not used.

ALLUSERS If any non-blank value is entered, the authority is granted to ALLUSERS.

DISPLAY AUTHORITY
When working with groups, this field is used to select a specific authorization report to be generated and displayed online in XEDIT mode. The values are:

- **CONNECT** The report lists those database users who have CONNECT authority only.
- **RESOURCE** The report lists those database users who have RESOURCE authority (CONNECT authority is implied).
- **DBA** The report lists those database users who have DBA authority.

userid When working with groups, a single user id is not appropriate.

Sample Scenarios For Working With Group Authorizations

These are some example steps to take when first using the Group Authorization tool:

1. Define Application Groups
2. Add Objects to the Application Groups
3. Define User Groups
4. Add Users to the User Groups
5. Grant Authorities to the User Groups

To help you get started, we have provided the following suggestions and example scenarios.

Analyze your current database authorization structure. You can start by grouping tables and views according to some common function or element(s). For instance, perhaps you would like to have all payroll, personnel, or accounting tables in their own specific groups. You can then define Application Groups for these various functions.

Let's say you have 5 tables and 3 views that belong in the personnel organization. You might want to define three Application Groups for this organization. The Application Groups will contain the tables and views from the personnel organization.

Application Group Definitions

```
AGROUP1 contains PERSTAB1, PERSTAB2, PERSTAB3, PERSVIEW1
AGROUP2 contains PERSTAB1, PERSTAB4, PERSTAB5, PERSVIEW2
AGROUP3 contains PERSVIEW3
```

Note: **PERSTAB1** is common to both AGROUP1 and AGROUP2, and AGROUP3 has only one view defined to it.

Step 1: Define Application Groups

To define (or add) an Application Group, select Option **3** from the Group Authorization Utility Menu (Figure 53 on page 153). The next menu presented will be the Application Group Menu. Here you must enter the name of the Application Group, together with the option to Add a Group, and the Group Type.

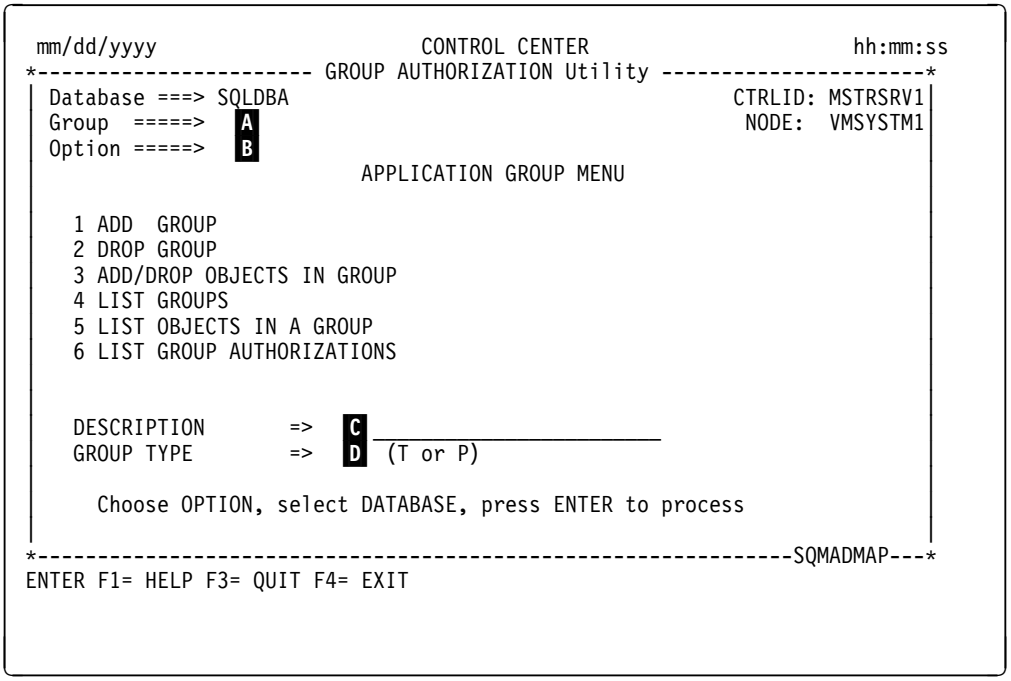


Figure 55. Control Center Application Group Menu

The required, or optional, fields are:

A The Application Group Name (**AGROUP1**)
B The Application Group option, 1-6 (**1**)
C The Group Description field is optional. If entered, it is stored in the database record and can be viewed in the Application Group List report.
D The Group Type identifier; **T** is for tables or views, **P** is for packages.

Step 2: Add Objects to the Application Group

After the Application Group has been added or defined, we need to add the objects (tables and views, or packages) to the group. From the Application Group Menu, select Option **3** to add the objects to **AGROUP1**.

The Object Functions Menu is where tables and views, or packages are added to the Application Group. In Figure 56 on page 158 we have added the personnel tables and view to Application Group **AGROUP1**.

```
 mm/dd/yyyy                      CONTROL CENTER                       hh:mm:ss
*----------------------- GROUP AUTHORIZATION Utility -----------------------*
| Database ===> SQLDBA                                    CTRLID: MSTRSRV1  |
| Group  =====> AGROUP1                                 NODE:  WMAVM7       |
| Option =====> 1                                                           |
|                              OBJECT FUNCTIONS                             |
|                                                                           |
|   1 ADD  OBJECTS                                                          |
|   2 DROP OBJECTS                                                          |
|                                                                           |
|   OBJECTS:    OWNER   OBJECT NAME           OWNER   OBJECT NAME           |
|              --------  ------------------  --------  ------------------   |
|              PERS____  PERSTAB1___________  ________  ____________________ |
|              PERS____  PERSTAB2___________  ________  ____________________ |
|              PERS____  PERSTAB3___________  ________  ____________________ |
|              PERS____  PERSVIEW1__________  ________  ____________________ |
|              ________  ____________________  ________  ____________________ |
|              ________  ____________________  ________  ____________________ |
|                                                                           |
|           Enter Object ids, select OPTION, press ENTER to process         |
|                                                                           |
*---------------------------------------------------------------SQMADMOB-----*
ENTER F1= HELP F3= QUIT F4= EXIT
```

Figure 56. Add Objects to Your Application Group

Step 3: Define User Groups

Once the Application Groups have been set up, define the users of your database to whom privileges must to be granted for the various personnel tables and view identified above.

Let's assume we have 10 users whose IDs are USER1 through USER10. You can now group these users by function or access needs, or whatever criteria you use to determine authorization.

In this case, we will define 3 User Groups (UGROUP1 through UGROUP3). Place the 10 users, User1 through User 10, into any of the 3 defined User Groups:

User Group Definitions

```
UGROUP1 consists of User1, User2, User3, User4, User5, User6
UGROUP2 consists of User1, User7, User8, User9, User10
UGROUP3 consists of User1 through User10
```

Note: **User1** is in all 3 groups, all ten users are in UGROUP3. Any distribution of users is allowed, depending on your authorization needs.

The User Group Menu is where you can add or delete a User Group, populate a User Group, and list User Groups.

```
 mm/dd/yyyy                      CONTROL CENTER                       hh:mm:ss
*---------------------- GROUP AUTHORIZATION Utility ------------------------*
| Database ===> SQLDBA                                        CTRLID: MSTRSRV1 |
| Group  =====> ________                                       NODE:  VMSYSTM1 |
| Option =====> ____                                                           |
|                              USER GROUP MENU                                 |
|                                                                              |
|   1 ADD  GROUP                                                               |
|   2 DROP GROUP                                                               |
|   3 ADD/DROP USERS IN A GROUP                                                |
|   4 LIST GROUPS                                                              |
|   5 LIST USERS IN A GROUP                                                    |
|   6 LIST GROUP AUTHORIZATIONS                                                |
|                                                                              |
|                                                                              |
|   DESCRIPTION       => ____________________________________________          |
|                                                                              |
|           Select OPTION, enter GROUP NAME, press ENTER to process            |
|                                                                              |
*------------------------------------------------------------------SQMADMGP-----*
ENTER F1= HELP F3= QUIT F4= EXIT
```

Figure 57. Menu for User Group Functions

Step 4: Add Users to the User Groups

After the specific User Group has been added, you can populate the User Group by selecting Option **3** from the User Group Menu.

```
 mm/dd/yyyy                      CONTROL CENTER                       hh:mm:ss
*---------------------- GROUP AUTHORIZATION Utility ------------------------*
| Database ===> SQLDBA                                        CTRLID: MSTRSRV1 |
| Group  =====> UGROUP1                                        NODE:  WMAVM7   |
| Option =====> 1                                                              |
|                                 USER MENU                                    |
|                                                                              |
|   1 ADD  USERS                                                               |
|   2 DROP USERS                                                               |
|   3 LIST USERS                                                               |
|                                                                              |
|   USER LIST:    NAME       NAME       NAME       NAME       NAME           |
|              User1___   ________   ________   ________   ________          |
|              User2___   ________   ________   ________   ________          |
|              User3___   ________   ________   ________   ________          |
|              User4___   ________   ________   ________   ________          |
|              User5___   ________   ________   ________   ________          |
|              User6___   ________   ________   ________   ________          |
|              ________   ________   ________   ________   ________          |
|                                                                              |
| Select OPTION, enter USERS if needed, press ENTER to process                 |
|                                                                              |
*------------------------------------------------------------------SQMADMUS-----*
ENTER F1= HELP F3= QUIT F4= EXIT
```

Figure 58. Users Added to UGROUP1

Once the users have been added, we can grant object authority to the User Group **UGROUP1**.

Step 5: Grant Authorities to the User Groups

Now issue grants or revokes to those User Groups associated with the six users who belong to UGROUP1. Using the Authorization Menu, grant SELECT and UPDATE on AGROUP1 to UGROUP1, as shown below.

```
 mm/dd/yyyy                      CONTROL CENTER                       hh:mm:ss
*---------------------- GROUP AUTHORIZATION Utility ----------------------*
| Database ===> SQLDBA                                      CTRLID: MSTRSRV1 |
|                                                            NODE:  VMSYSTM1 |
| OPTION                 => 1              ( 1 GRANT   2 REVOKE )           |
|                                                                           |
| PRIVILEGES             => 1   4          ( 1 SELECT  2 INSERT  3 DELETE   |
|                                            4 UPDATE  5 ALL     6 EXECUTE  |
|                              ON                                           |
| OBJECT OWNER           => ________                                        |
| OBJECT NAME            => ____________________                            |
|                              OR                                           |
| APPLICATION GROUP NAME => AGROUP1                                         |
|                                                                           |
|                              TO or FROM                                   |
| USER GROUP             => UGROUP1                                         |
|                                                                           |
| WITH GRANT OPTION      => _ ( 1 YES)                                      |
|                                                                           |
|     Enter parms, select OPTION, press ENTER to process                    |
|                                                                           |
*-------------------------------------------------------------SQMADMPR----*
ENTER F1= HELP F3= QUIT F4= EXIT
```

Figure 59. Authorization Menu

This will have the effect of granting SELECT and UPDATE privileges on tables PERSTAB1, PERSTAB2, PERSTAB3, and PERSVIEW1 to user IDs User1 through User6.

To continue with this scenario, say you were to also enter:

Grant to UGROUP2

GRANT SELECT, UPDATE on AGROUP1 to UGROUP2

This grants SELECT and UPDATE privileges on tables PERSTAB1, PERSTAB2, PERSTAB3, and PERSVIEW1 to user IDs User1, User7, User8, User9, and User10.

Note: **User1** has been granted SELECT and UPDATE privileges to the same tables in AGROUP1 by virtue of belonging to two different User Groups.

This means that if this revoke were done:

Revoke from UGROUP1

```
REVOKE UPDATE on AGROUP1 from UGROUP1
```

User2, User3, User4, User5 and User 6 would lose Update privileges on the tables in AGROUP1. They would retain the SELECT privilege. However **User1** would retain Update privilege because **User1** still has it by belonging to **UGROUP2**.

This example is only one of very many which can be developed and maintained by using the Group Administration tool. All sorts of variations are possible, and the tool will maintain the structures and authorizations for you.

Performance

There are some performance considerations for doing on-line authorizations. However, none of the functions will take longer than if they were done individually by a DBA through ISQL or some other method. In fact, overall elapsed time should be substantially less when key strokes and preparation time are taken into consideration.

Another performance factor, though probably of little concern, is that two tables (USERID_GROUP_TAB and APPL_GROUP_TAB) have primary indexes which will be incremental in nature. These should be monitored by the DBA for reorganization; however, this should only be a problem in the event that very many groups are defined.

Usage Notes

1. You *can* grant and revoke authorizations to a User Group where users have not yet been added. The authorizations will be made to the Group ID.
2. You can then *ADD* users who will automatically inherit the grants possessed by the Group ID.
3. However, when you drop users from the Group, after that last user is dropped from the group, the Group ID also loses its privileges.
4. Be careful with mixing view and tables together in an Application Group. While you can grant SELECT on a mix of tables and views, you may *not* be able to grant INSERT unless the SQLADMIN ID has specific authority to the view, or the view itself permits INSERT (or UPDATE).

Special Considerations

1. All authorizations are only granted to Groups, not individual user IDs. A way around this is to define a User Group with only 1 user ID in it. You can then grant authorizations to that user group, thus effectively granting to an individual ID.
2. Authorizations can be granted to a User Group for an individual table which is not a member of an Application Group. For example:

Individual Tables

GRANT SELECT on SQLDBA.ACTIVITY to UGROUP1

This will grant the privilege to all users defined in UGROUP1. However, since the grant is to an individual database entity, and not to an object belonging to an Application Group, this authorization is NOT recorded in the tool's tables.

Thus, if any user ID defined in UGROUP1 were later to have its privilege to SQLDBA.ACTIVITY revoked, because that table was defined in an Application Group, the privilege would be deleted from the SYSTEM.SYSTABAUTH table since the Group Authorization tool does NOT know about this individually granted authority.

The way around this limitation is to define the table in its own group. The same way a single user can be defined in a User Group, so too, a single table (or view, or package) can be defined in its own application group.

3. The authorizations which can be granted or revoked are:
   - SELECT, INSERT, UPDATE, DELETE and ALL on Tables/Views
   - The ALL authorization is equivalent to the SELECT, INSERT, UPDATE, and DELETE privileges and does **not** include ALTER, INDEX or REFERENCES.
   - EXECUTE on Packages
   - All of the above authorities can be granted with the GRANT option
4. All grants and revokes are done under the Connect ID of SQLADMIN. This ID will be generated into the database at install time, with a random password. The password is obtained by the program when invoked, and the CONNECT to the ID is done prior to issuing authorities.
5. When granting EXECUTE on packages, the SQLADMIN ID must have the RUN privilege with GRANT option in order to do the GRANT. An error message will be produced if this is not the case.
6. There is no ability to grant referential constraint privileges.
7. If you have current authorizations recorded in SYSTEM.SYSTABAUTH and SYSTEM.SYSPROGAUTH, these entries will not be affected by the tool since all privileges are granted by SQLADMIN. You should, therefore, remove any old or non-SQLADMIN grants as you may deem necessary as you build your authorization scheme and issue GRANTs using the tool.
8. The following is an example of a query you can use to obtain other information from the Group Authorization tables. For example, if you want a list of which users (and their group IDs) have SELECT authorization to all the objects in a particular Application Group, the following query will obtain the desired result:

```
SELECT A.APPL_GROUP_NAME, O.OBJECT_OWNER, O.OBJECT_NAME, G.S_AUTH,
       U.USERID, UG.GROUP_NAME
FROM   SQLADMIN.USERID_GROUP_TAB UG, SQLADMIN.USERID_TAB U,
       SQLADMIN.OBJECT_TAB O, SQLADMIN.GROUP_AUTH_TAB G,
       SQLADMIN.APPL_GROUP_TAB A
WHERE  A.APPL_GROUP_NAME = 'AGROUP5'         AND
       O.APPL_GROUP_ID   = G.APPL_GROUP_ID   AND
       U.GROUP_ID        = G.USERID_GROUP_ID AND
       A.APPL_GROUP_ID   = G.APPL_GROUP_ID   AND
       UG.GROUP_ID       = U.GROUP_ID        AND
       G.S_AUTH > ' '
ORDER BY 1,2,3,5
```

Figure 60. Query Using Join Statement

```
APPL
GROUP      OBJECT     OBJECT                          GROUP
NAME       OWNER      NAME        S AUTH    USERID    NAME
----       ------     --------    ------    ------    ------
AGROUP5    M760595    TABERROR      G       TESTGP3   TESTGP3
AGROUP5    M760595    TABERROR      Y       TESTGP4   TESTGP4
AGROUP5    M760595    TABERROR      G       USER1     TESTGP3
AGROUP5    M760595    TABERROR      Y       USER1     TESTGP4
AGROUP5    M760595    TABERROR      G       USER2     TESTGP3
AGROUP5    M760595    TABERROR      G       USER3     TESTGP3
AGROUP5    M760595    TABERROR      Y       USER4     TESTGP4
AGROUP5    M760595    TABERROR      Y       USER5     TESTGP4
AGROUP5    SQLDBA     ACTIVITY      G       TESTGP3   TESTGP3
AGROUP5    SQLDBA     ACTIVITY      Y       TESTGP4   TESTGP4
AGROUP5    SQLDBA     ACTIVITY      G       USER1     TESTGP3
AGROUP5    SQLDBA     ACTIVITY      Y       USER1     TESTGP4
```

Figure 61. Results of Join Query

Or, if you want to modify a group's description, you can simply update the GROUP_DESC field for the USERID_GROUP_TAB or the APPL_DESC field of the APPL_GROUP_TAB.

9. You must have DBA authority to use the tool and should be initialized to the desired database prior to invoking the tool.

Chapter 14. Job Scheduling Tool

Overview

The Job Scheduling tool is an integrated and flexible tool for scheduling the automatic execution of other Control Center tools. One-time activities such as adding a dbextent can be scheduled to execute once at a specified date and time; repetitive activities such as performing an archive can be scheduled to execute on a routine basis. Jobs can be executed sequentially, where each job is dependent on the completion of the previous (single-threaded execution), or concurrently as independent activities (multiple threaded execution).

Job control features include defining job priority; execution dependencies on database availability, support machine or service machine availability, and completion of other jobs; and a time window for job execution to begin and complete within.

Additionally, "no-process" days can be defined for your installation, such as local holidays or when your site is shut down. The product avoids scheduling jobs on these days and automatically reschedules them to execute on a specified earlier or later day.

How You Might Use the Job Scheduling Tool

As an example of how you might use the Job Scheduling tool, consider the time and work involved to manually initiate each tool to perform a full database archive, immediately followed by several DBSPACE reorganizations when the database returns to multiple user mode, followed by a log archive when all reorganizations complete. Compound this effort by the number of times you want to repeat the process. Once every month? Once every week? Using the Job Scheduling tool you can establish a thread of inter-dependent jobs to complete the same process, from start to finish, automatically. Furthermore, you can schedule the process to run repeatedly, such as every five days, and optionally during off-shift hours.

Advanced Usage: Although it is beyond the scope of this chapter, the Job Scheduling tool can also be used to schedule non-Control Center tools, such as your own internally developed tools, to execute on a Control Center support machine. See Appendix D, “Master Scheduling Tool” on page 507.

Before You Begin

If you plan on scheduling the execution of Database Administration tools, you should:

- Know the machine ID of a previously installed Control Center support machine able to work with the database you're working with (contact your Control Center System Administrator).
- Review how Database Administration tools work. Refer to “Control Center Tools Overview” on page 7.

Who Can Use the Job Scheduling Tools

Use of the Job Scheduling tool and Job Schedule List tool requires Control Center DBA authority or greater.

How Job Scheduling Works

The Job Scheduling tool itself runs on the Control Center service machine. It is here that the product maintains all job schedules, control files, and execution history files; performs job analysis; and initiates job processing, regardless of where the tool will actually execute. The product initiates System Administration tools to execute directly on the target database machine and Database Administration tools to execute on a specified Control Center support machine.

Note:

- Database Administration tools should not be scheduled to execute on the service machine where they will adversely impact the product's ability to communicate with the database machines it manages. These tools should be scheduled to run on a Control Center support machine. Refer to Table 2 on page 11 for a list of tools.
- System Administration Tools should always be scheduled to execute on the Control Center service machine. Refer to Table 1 on page 8 for a list of tools.

The product will notify you when a job executes, if it fails, and if it does not execute within its defined execution window (discussed later). It also keeps a running history of each job execution start time, stop time, and result (success or failure). Between the time a job is initiated and completes on its own, whether successfully or not, the product is largely unaware of the executing tool. For example, if a job running on a support machine were interrupted by some means, the product would be unaware that the job is no longer active. In fact, until specifically told otherwise, it will show the job status as being active. We discuss how to resolve this problem and others later in this chapter.

Most job output is saved on the machine where the job executes and can be listed, viewed, modified, and deleted using the Job Schedule List tool. Output is automatically purged after the number of days specified during the product installation process.

Note:

Job output from the DBSPACE Reorganization Driver and the Reorganization Index Tool is sent to the Control Center service machine and NOT saved.

Further discussions on how Job Scheduling works is provided in the following section and in “Job Scheduling Architecture” on page 179.

Job Scheduling Concepts

This section discusses the basic concepts of Job Scheduling, providing you with enough information to get you started using the job scheduling tool. Later in this chapter we discuss the Job Scheduling Architecture.

Job Dependencies

The Job Scheduling tool allows you to define various dependencies for each job which must be met prior to the job being automatically initiated by the product. Typically, a job will be dependent on the availability of the target database manager, either up, running in multiple user mode or down, available to execute a single user mode tool. This will assure that the job is not initiated at a time when the database is unable to process it successfully.

Complex and time-consuming tasks which require several sequentially executed jobs can be automated by defining the execution of each job to be dependent on the completion of the previous. For example, you can schedule a database archive, followed by a reorganization, followed by another archive to execute as a single thread. See Figure 62.

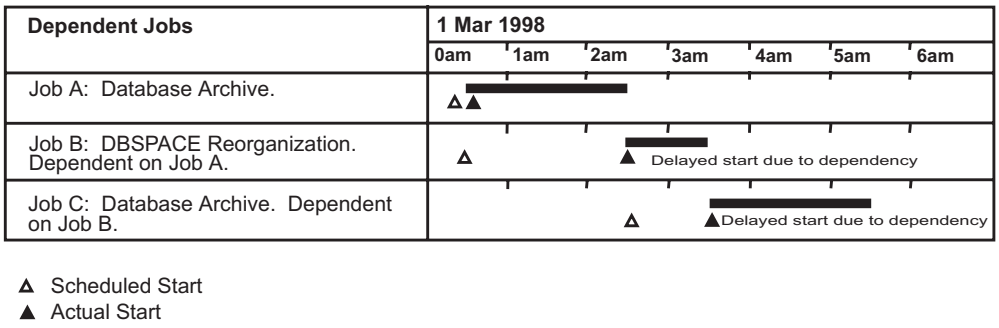


Figure 62. Dependent Job Example

Additionally, you can make one job dependent on the completion status (success or failure) of a previous job, allowing even more complex tasks to be scheduled for automatic execution.

Other job dependencies which effectively control if and when a job is initiated are discussed in the following topics of this section.

Schedule Interval

Repetitive activities such as performing an archive can be scheduled to execute on a routine basis by specifying the time between events. We refer to this as the *schedule interval*.

For example, to schedule a full archive of your production database to occur once a week, you would first specify the date and time of the initial execution. Next, you would specify a schedule interval of "1 week". After the initial successful job execution and every one thereafter, the product will automatically reschedule the job to execute in "1 week".

The example below shows the execution dates of four jobs, all initiated on March 1st, 1997, but each with a different schedule interval. We discuss the product's handling of the March 5th holiday next.

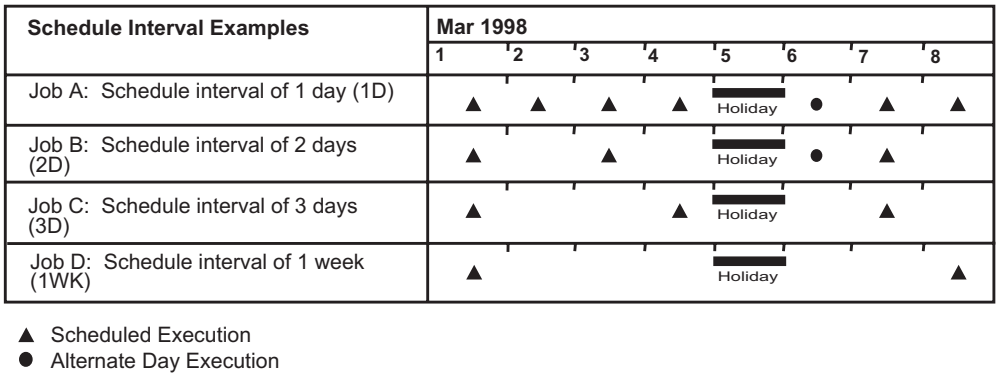


Figure 63. Schedule Interval Examples

Schedule Interval and the SQLMSTR HOLIDAYS File

When the product automatically reschedules a job to execute, it references the SQLMSTR HOLIDAYS file, which resides on the service machine's 191 A-disk, to determine whether the proposed date matches one of the locally established holidays. See “SQLMSTR HOLIDAYS File” on page 183 for a detailed explanation of holiday scheduling.[7]

If a match is found, the product will use the specified alternate date for the job's next execution. The alternate day can be earlier or later than the holiday. In the example, March 5th is a holiday and March 6th is the alternate day. After job A completes on March 4th, it is rescheduled to run on the alternate day, coincidentally a day it normally would have run had there not been a holiday. Job B is also rescheduled to run on the alternate day after it executes on March 3rd. Jobs C and D are not impacted by the holiday.

Operational Note: The SQLMSTR HOLIDAYS file is referenced at the time a job is rescheduled. Any changes made to SQLMSTR HOLIDAYS will not affect those jobs currently scheduled.

Job Priority

The product can initiate several jobs to execute concurrently as long as no two jobs are vying for the same resource; for example, a Control Center support machine. If and when this should occur, it will favor one job over all others, initiating it to execute while the others are put on hold. As one jobs completes, the product will initiate the next as long as all other dependencies for that job are satisfied. You can control which jobs are favored over others by assigning a priority to each job. The job priority, which ranges from 1 to 9, where 1 is the highest, acts as the tie-breaker in these situations.

[7] Generating and updating the SQLMSTR HOLIDAYS file is the responsibility of the Control Center System Administrator.

In the example below jobs A and B are to begin execution at the same time on support machine MSTRSUP. Since job A has the higher priority, it is initiated first and job B is initiated after it completes.

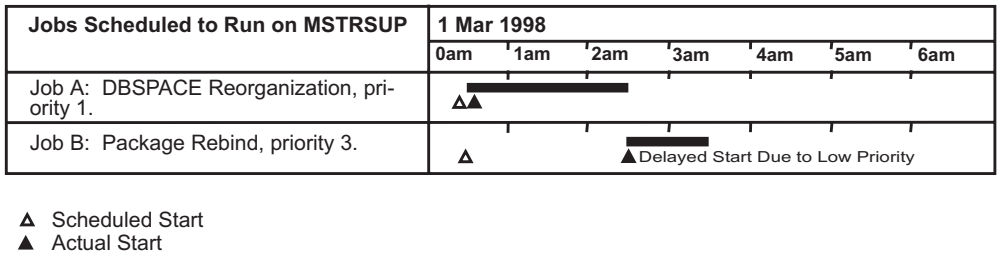


Figure 64. Job Priority Example

Execution Window

The job execution time window defines a period of time within which a job is to begin and complete execution. The product will initiate a job as early as possible within the execution window, after all other dependencies have been met, as long as it determines that the job will complete prior to the end of the window. This allows you to restrict the execution of jobs to periods of low system usage. Conversely, it allows you to prevent the execution of jobs from overlapping into periods of high system usage.

If a job does not execute during the defined execution window, due to system downtime or dependencies not being met, the product will notify you and automatically reschedule the job for the next available window based on the job's defined schedule interval. Jobs scheduled to execute one time only will be placed in an inactive status.

Control Center uses a job's average runtime to determine whether it can complete within the time remaining in the defined execution window. The average runtime is calculated automatically by it based on the five most recent successful job executions. If it determines that there is insufficient time in the current window, the job will automatically be rescheduled.

In the example that follows, three jobs of varying average runtimes are scheduled to run concurrently. All are scheduled to start at 1:00 a.m. and have a defined window end time of 5:00 a.m.. Due to an unplanned event, the system is unavailable from 1:00 a.m. until 4:00 a.m.. When the system comes back online, only one hour remains before the end of the execution window. At that time, it initiates jobs A and B since both have an average runtime of less than an hour and should complete prior to 5:00 a.m.. Job C has an average runtime of one and a half hours and will not be able to complete by the end of the execution window. The product automatically reschedules it to execute during the next available execution window.

Valid start times for jobs A and B are indicated by the bars (**---**).

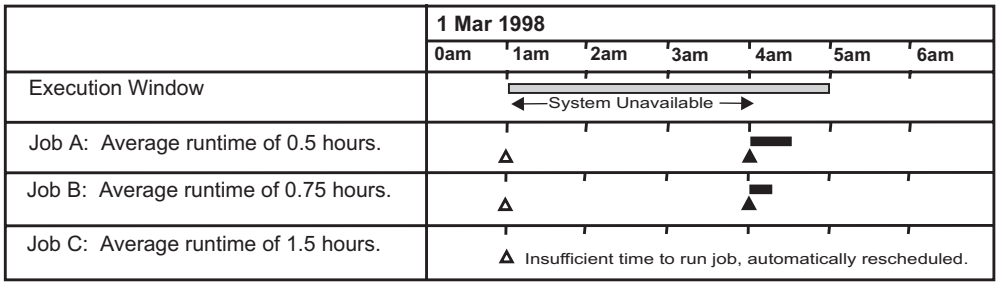


Figure 65. Job Execution Window Example

Job Scheduling Tool

The Job Scheduling tool is available as an option within many of the System Administration and Database Administration tools. It is also a part of the Master Scheduling tool, where it can be used to schedule non-Control Center tools, such as your own internally developed tools. The Master Scheduling tool is discussed in Appendix D.

The Job Schedule List tool is used to list all jobs for a given database, view job output, modify and reschedule jobs, delete jobs, and initiate jobs for immediate execution. Refer to “Job Schedule List Tool” on page 176.

Job Scheduler Panel

The basic Job Scheduling entry panel is shown below. Except for the Server-mach entry field, it is the same regardless of which tool you're scheduling. The Server-mach entry field is not applicable when scheduling tools that execute on the target database machine (single user mode tools).

```
 mm/dd/yyyy                      CONTROL CENTER                        hh:mm:ss
*------------------------------ Job Scheduler -----------------------------*
| Command ===>                                                CTRLID: MSTRSRV |
| Jobname        ==> ________                                  NODE:  VMSYSTM |
| Server-mach    ==> ____________________ (Required for Multi-user mode)     |
| Job-status         ==> S    (S=Scheduled, A=Active, F=Failed, I=Inactive)  |
| Priority           ==> 5    ( 1 through 9, 1 = HIGH, 9 = LOW )             |
| Next-start         ==> 19970419 23:00   ( format: YYYYMMDD HH:MM )         |
| Window-end         ==> 19970420 08:00   ( format: YYYYMMDD HH:MM )         |
| Schedule-interval  ==> 2WK        ( ONCE or form nnnXX, where XX is:        |
|                                      MI, HR, DY, WK, WD, WE, MO, YR )     |
| Required-database  ==> SQLDBA           ( Database machine-id )           |
| Required-dbstatus  ==> U                ( U=Up, D=Down, A=Any )           |
| Dependent-jobname  ==> ________         ( Prior job must run first )      |
| Dependjob-result   ==> _                ( S=Successful, F=Failed, A=Any ) |
| Average-runtime    ==> 01:00            ( format: HH:MM )                 |
| Last-jobstart      ==> _____________   Last-jobend  ==> _____________     |
| Notify   ==> MSTRSRV1                                                     |
| Execute  ==> SQLRBIND SQLDBA                                              |
|                                                                           |
|                                                                           |
*-------------------------------------------------------------SQMSCHDM----*
 PF: 1 Help   2 Detailed Help   3 End (Cancel)     6 Schedule
```

Figure 66. Job Scheduling Tool Entry Panel

Entry Field	Description
Jobname	Uniquely identifies each scheduled job. The jobname is limited to a maximum of eight alphanumeric characters. Special characters must be avoided unless they are valid for filenames and filetypes of CMS files. **Usage Consideration:** 1. Duplicate jobnames are not allowed within the product, regardless of which database they execute against or where they run. 2. Jobnames can not contain an underscore.
Server-mach	Applicable only for multiple user mode tools. Identifies the Control Center support machine where the tool executes. The job schedule is maintained on the Control Center service machine specified by your communication path setting shown in the upper right-hand corner of the panel. Refer to “Control Center Communication Path and Database Settings” on page 143. When the product initiates a multiple user mode job, it sends the appropriate command to the specified support machine where it is executed.

Note!

- Database Administration tools should not be scheduled to execute on the service machine where they will adversely impact the product's ability to communicate with the database machines it manages. They should be scheduled to execute on a Control Center support machine or a user machine. Refer to Table 2 on page 11 for a list of tools.
- System Administration Tools should always be scheduled to execute on the Control Center service machine.
- The Server-mach field should be left blank or contain the Control Center service machine name when the EXECUTE field is a command to be executed on the database console; (SQM SQLDBA SQLEND ARCHIVE).

Job-status
: A single-character flag which indicates the status of the job at any given point in time. When the job is initially entered, the Job-status value should be set to "S" to indicate that it is Scheduled for execution. When it initiates the job, the Job-status will be changed to "A" to indicate that the job is Active. If the job completes successfully, the product will change the Job-status back to "S" and automatically reschedule the job for its next specified execution. If the job fails, the Job-status is changed to "F", which will essentially prevent the job from automatically executing again until someone manually intervenes to correct the cause of the failure and change the Job-status back to "S". Finally, Job-status value "I" is used to indicate that a job is inactive. It can be specified manually to temporarily remove a job from automatic execution.

Priority
: An integer between 1 and 9 which indicates the relative priority of this job compared to other jobs within the product schedule. A value of 1 indicates the highest priority, while a value of 9 indicates the lowest priority. This parameter is only considered by the product during job initiation time. If two or more jobs are scheduled for initiation at the same time and all dependencies have been met for each job, the job priority will indicate which job will be initiated first. If multiple jobs depend on a common resource (such as a specified Control Center support machine), then the jobs will be executed in sequential order based on the specified priorities. See “Job Priority” on page 168 for more information.

Next-start
: Identifies a dependent date and time that must be reached before the job is initiated. It does not guarantee that the job will begin at the specified date/time, but only that the job will not be initiated prior to this date and time. Actual start date/time will vary based on when all job dependencies are met.

  Together, the Next-start and Window-end parameters define the execution time window within which the job is to begin and complete execution. See “Execution Window” on page 169 for more information.

  The format of this parameter is *YYYYMMDD hh:mm*, where *YYYY* is a four-digit year, *MM* is a two-digit month, *DD* is a two-digit day, *hh* is a two-digit number to indicate the hour of the day (01 to 24),

and *mm* is a two-digit number to indicate the minutes (01 to 59). An asterisk, "*", will appear next to the Next-start date when modifying the job and a "Rescheduled because of holiday" message will appear when viewing the job if this job has been scheduled to run on an alternative date as defined in the SQLMSTR HOLIDAYS file. See “SQLMSTR HOLIDAYS File” on page 183.

Usage Consideration: The data entry format must be followed exactly, including leading zeroes, spaces, and colons, to be accepted as valid input.

Window-end

The Window-end parameter is used with the Next-start parameter to define the execution time window within which the job is to begin and complete execution. For more information, see “Execution Window” on page 169.

The data entry format is the same as that for the Next-start parameter (*YYYYMMDD hh:mm*).

Schedule-interval

Specifies the frequency of job executions. Each time the job executes successfully, the product will use this value to automatically reschedule it to execute again. To schedule a job for a single execution, enter the keyword ONCE. All other intervals are specified by using an integer value between 1 and 999 with a two-character keyword to indicate the Unit of Measure.

Valid two-character keywords are:

MI for minutes
HR for hours
DY for days
WK for weeks
WD for weekdays
WE for weekends
MO for months and
YR for years

Example Schedule-intervals are:

23MI Every 23 minutes
3WK Every 3 weeks
2DY Every 2 days
1MO Every month

Refer to Figure 63 on page 168 for a discussion of the SQLMSTR HOLIDAYS file.

Usage Consideration: The NEXT-START and WINDOW-END fields will be used in combination with the SCHEDULED-INTERVAL field to determine the NEW-START time. A job will only run ONCE in a scheduled window.

Required-database

This parameter is optional, but should be used for all jobs that require a database to be available in a specified state (up or down) before it can be executed. The required state is indicated by the related Required-dbstatus parameter.

Required-dbstatus

This parameter is valid only when a database is entered for the Required-database parameter. It indicates what status the Required-database must be in before the product will initiate the job.

Possible values are:

U The database must be Up in multiple user mode before the job can be initiated.

D The database must be Down before the job can be initiated.

A The database can be in ANY state (up or down) for the job to be initiated.

Operational Note: Specifying "A" (Any) will still prevent the job from being initiated if the database is in a status other than up or down (such as archiving).

Dependent-jobname

An optional parameter used to specify another Control Center job that must execute before this job. This parameter allows a group of jobs to be scheduled to execute in sequence, such as an archive job, followed by a reorganization job, followed by another archive. The Dependjob-result parameter is used to specify whether the previous job must complete successfully or not.

Dependjob-result

This parameter is only valid when the Dependent-jobname parameter is used to specify a dependent Control Center job that must execute first.

Possible values are:

S The dependent job must complete Successfully before this job will be initiated.

F The dependent job must Fail before this job will be initiated.

A Any completion status of the dependent job is acceptable for this job to be initiated.

Average-runtime

Initially, this parameter should be set with your estimate of how long the job will run; it is entered in the form *hh:mm*, where *hh* is a two-digit integer indicating *hours* and *mm* is a two-digit integer indicating *minutes*. Afterwards, the product will automatically maintain this parameter by updating it each time the job completes successfully, using the average runtime from the previous five successful runs.

If you change the average runtime value on the panel and press PF6, the RUNTIMES file will be updated with five entries using the new average runtime. This will change the average runtime for the job to the new average runtime selected. The five entries in the RUNTIMES file will show **UPDATED** in the STATUS field.

When all job dependencies have been met, including the Next-start date/time, the product will add the average runtime value to the current time of day and compare the result with the value of the

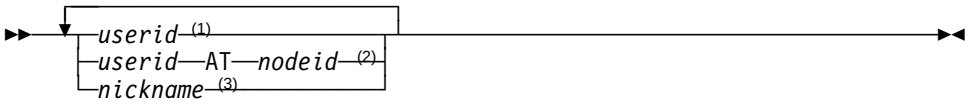
Window-end parameter. If it determines that the job cannot complete prior to the Window-end time, then the job will not be initiated. Instead, it will reschedule the job by adding the Schedule-interval value to the Next-start and Window-end parameters. If the Schedule-interval parameter is ONCE, indicating that it cannot be rescheduled, the job-status value is changed to inactive. The job can then be manually rescheduled or started immediately using the Job Schedule List tool.

Last-jobstart — Maintained automatically by the product. It is updated each time the job is initiated and, under most circumstances, should not be updated manually. It is used by the Job Scheduling tool to determine whether job dependencies have been met.

Last-jobend — Maintained automatically by the product. It is updated each time the job completes and, under most circumstances, should not be updated manually.

Notify — Specifies the users who should be notified when the job executes, if it fails, and if it does not execute within the defined execution window.

The syntax for entering the Notify parameter is given below.



Notes:

1. A local user. Nodeid is assumed to be the same as that of the Control Center service machine.
2. A remote user. The nodeid is different than that of the Control Center service machine.
3. A nickname within the CMS NAMES file on the Control Center service machine's 191 A-disk. Refer to the NAMES command documentation provided with VM/CMS. The nickname format allows multiple users to be associated with a single nickname, which eliminates the limited space restrictions of the parameter input panel.

Execute — The executable command that is processed when the job is initiated. Unless you're scheduling a job using the Master Scheduling tool, this command is built for you automatically based on your input to the various data entry panels associated with the tool being scheduled.

Understanding the command and command syntax requires familiarity with the product's command mode interface. Refer to Appendix G, “Command Mode Interface” on page 523.

Job Schedule List Tool

The Job Schedule List tool is a comprehensive tool for managing all Control Center jobs for a given database. This includes jobs that are scheduled, active, have failed, or are inactive.

Scheduled Job List Panel

Listed are all jobs for the database identified by your communication path and database settings shown near the top of the panel. The communication path can be changed using Option **C** from the Control Center Main Menu, whereas the database setting can be changed on the Control Center Main Menu panel itself. See Control Center Communication Path and Database Settings 143.

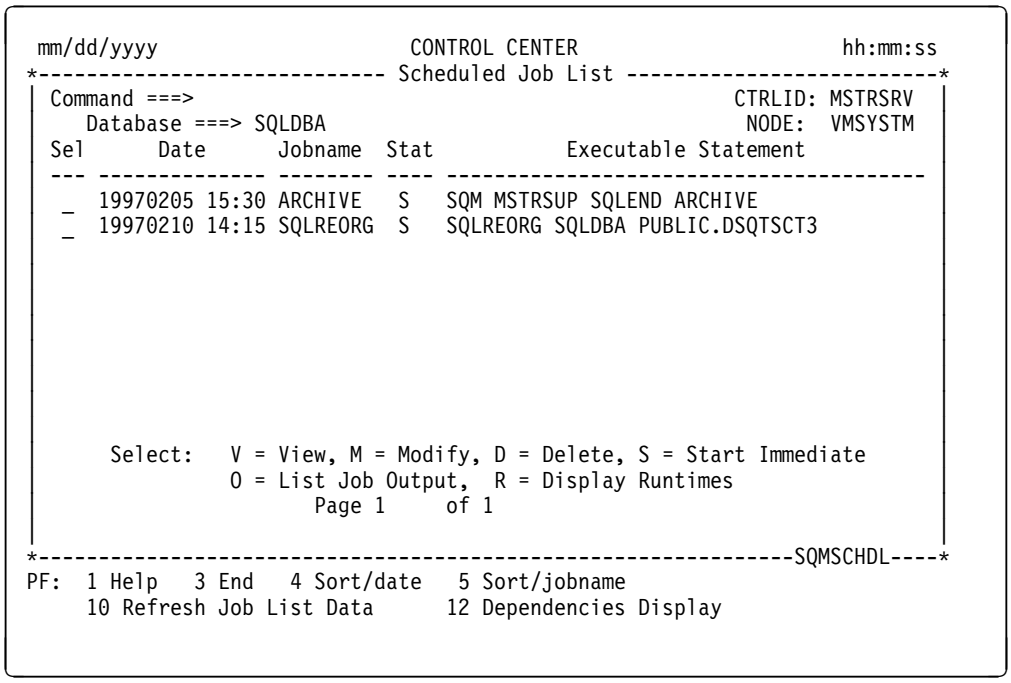


Figure 67. Job Schedule List Tool

Column Heading	Description
Date	The scheduled execution date and time. Actual start date/time will vary based on when all job dependencies are met.
Jobname	The jobname which is unique among all jobs managed by the Control Center service machine.
Stat	The current job status. Possible values are: **S** Job is scheduled for execution. **A** Job is currently active. See “Problem Resolution” on page 184.

F The job failed during the last execution.

When a job fails, the product does not automatically reschedule it for the next schedule interval. Manual intervention is required to correct the problem and change the status back to 'S'.

I The job is inactive. It will not be executed.

You can place a job in an inactive status to temporarily remove it from automatic execution by the product. Jobs scheduled to execute one time only are placed in an inactive status after they execute.

Executable Statement — The executable command that is processed when the job is initiated.

The following column is displayed when PF12 (Dependencies Display) is selected.

Current Dependencies — Shown are the current unsatisfied job dependencies. Keyword *window* indicates that the current date/time is not within the defined execution window.

Select Options

Select options can be invoked against listed jobs.

Select Option	Action
View (V)	Displays current schedule information for the selected job, including the status of each job dependency.
Modify (M)	Invokes the Job Scheduling tool to display the selected job and allow modification of all job parameters.
Delete (D)	Deletes job. You will be prompted to confirm the delete request. **Usage Consideration:** Prior to deleting a job you should delete all job output using the List Job Output (O) select option.
Start Immediate (S)	Initiates immediate execution of job. Current job dependencies will be ignored, allowing the possibility that the job initiation or execution can fail.
List Job Output (O)	Lists job output files. Refer to “Job Output Filelist Panel” on page 178.
Display Runtimes (R)	Displays the start and end times, elapsed time (runtime), and completion status for the fourteen most recent job executions. The runtime history file is not created until the job is run for the first time.

PF Key Selections

PF Key	Action
Sort/date (PF4)	Sorts list by date.
Sort/jobname (PF5)	Sorts list by jobname.
Refresh Job List Data (PF10)	Refreshes all job list information.
Dependencies Display/Executable Statement Display (PF12)	Switches display between the 'Executable Statement' and the 'Current Dependencies' columns.

Job Output Filelist Panel

Job output is saved on the machine where the job executes and can be viewed, modified, and deleted from the Job Output Filelist panel.

```
 mm/dd/yyyy                      CONTROL CENTER                       hh:mm:ss
*----------------------------- Job Output Filelist -------------------------*
| Command ===>                                                CTRLID: MSTRSRV1 |
| Job Name:  DSQWKLY                                           NODE:  VMSYSTM  |
|     FILENAME FILETYPE FM      DATE     TIME                                 |
|   _ DSQTSCT3 LISTING  A1    04/20/97  10:34:20                             |
|   _ DSQTSCT3 UNLIST   A1    04/20/97  10:34:13                             |
|   _ DSQTSCT3 EXEC     A1    04/20/97  10:34:12                             |
|   _ DSQTSCT3 SQLDBSU  A1    04/20/97  10:34:03                             |
|   _ DSQTSCT3 UNLOAD   A1    04/20/97  10:33:47                             |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
|          Select:   V = View, M = Modify, D = Delete                        |
|                              Page 1      of 1                              |
|                                                                            |
*----------------------------------------------------------------SQMOLIST----*
PF:  3 End   4 Sort/date  5 Sort/name
```

Figure 68. Job Output Filelist Panel

Column Heading	Description
FILENAME	Job output filename.
FILETYPE	Job output filetype.
FM	Job output filemode.
DATE	Date when output file was created.
TIME	Time when output file was created.

Select Options

Select options can be invoked against listed job output files.

Select Option	Action
View (V)	Displays the output file using XEDIT.
Modify (M)	Displays the output file using XEDIT; modifications can be made and the source updated. You will be prompted to confirm the update of the file on the target machine.
Delete (D)	Deletes the output file. You will be prompted to confirm the delete action.

PF Key Selections

PF Key	Action
Sort/date (PF4)	Sorts list by file creation date.
Sort/name (PF5)	Sorts list by filename.

Job Scheduling Architecture

The underlying scheduling interrupt capability of the product is implemented using the SQLMSTR TIMES file maintained on the Control Center service machine's 191 A-disk. Each entry in the SQLMSTR TIMES file specifies a date (can be specified for repetitive interrupts), time, and a command that is to be executed at each interrupt. A direct interface to the SQLMSTR TIMES files provided using the Master Scheduling tool, but all other user interfaces are through the Job Scheduling tool.

The Database Monitoring tools do not utilize the job scheduler. These tools are designed to operate independently of all other scheduled activities, and therefore do not require the additional dependency characteristics provided by the Job Scheduling tool.

The job scheduler utilizes several additional files on the Control Center service machine which provide data that eventually flows into the SQLMSTR TIMES file as scheduled interrupts. Prior to the actual entry of a scheduled job into the SQLMSTR TIMES file, a thorough schedule analysis is performed by the job schedule analyzer routine. The analyzer reviews the condition of all databases, the current status of all scheduled jobs, and other job-related factors to determine whether the event should be scheduled to execute, and if so, the proper date and time for initiation.

There will always be an entry within the SQLMSTR TIMES file for every scheduled Control Center job, but the scheduled event will differ depending on the results of the job schedule analysis. If the job is ready for execution (all dependencies have been met), the event for the job will be INITIATION. If the job has any dependencies that remain unsatisfied, then the event for the job will be to RESCHEDULE the job at the end of the currently defined window. (If all job dependencies are met before the event interrupt causes the job to be RESCHEDULED, the event will be changed to START JOB). If the job is currently running, the SQLMSTR TIMES file will contain an interrupt entry for the job to issue a warning message if it runs past the end of the job's specified processing window.

Related Files

Several different files are maintained by the job scheduler on the 191 A-disk of the Control Center service machine to manage scheduled jobs. The scheduling parameters of each job are maintained in a CMS file with a filename equal to the jobname and a filetype of SQMJOB. When a user executes the VIEW, MODIFY, or DELETE functions for a specific job, the SQMJOB file is accessed through the Control Center interface.

The information from each separate SQMJOB file is collected by the job scheduler into a single master schedule file with a filename filetype of SQLMSTR JOBS. This file contains additional information about job dependencies generated by the job schedule analyzer.

The information within the SQLMSTR JOBS file is translated by the job scheduler into specific time/date interrupts that are then inserted into the SQLMSTR TIMES file. This file contains the actual interrupt driver events that will trigger the initiation of a job at the appropriate date and time.

When each job executes, the job scheduler will record the start and stop times, and the results (success or failure). This information is kept in a separate file for each job. The filename of each file will be the applicable jobname and the filetype will be RUNTIMES.

The job scheduler will also utilize the SQLMSTR HOLIDAYS file during job rescheduling to avoid the possibility that jobs will be scheduled for execution on days when the system will be unattended, such as holidays or weekends. The SQLMSTR HOLIDAYS file is maintained on the Control Center service machine's 191 A-disk by the Administrator. It contains future dates which should be *avoided* by scheduled jobs, along with alternate dates which should be used if any job would normally be scheduled on the given date.

SQMJOB File

The scheduling parameters of each job are kept on the service machine's 191 A-disk in separate CMS files named jobname SQMJOB. An SQMJOB file is created when a job is initially scheduled. It contains parameter tags in the format of CMS NAMES file tags, along with the associated parameter values. To avoid the possibility of incorrectly modifying the tag syntax, these files should not be updated manually.

Figure 69 on page 181 is an example of an SQMJOB file for a job with jobname QMFREORG that is scheduled to reorganize a QMF DBSPACE every two months. It uses Control Center support machine MSTRSUP and requires the SQLMACH database virtual machine to be UP in multiple user mode. The SQLREORG command executes against dbname SQLDBA.

```
:Nick.QMFREORG
:Jobname.QMFREORG
:Server_machine.MSTRSUP
:Job_status.S
:Priority.5
:Next_start.19970130 23:00
:Window_end.19970201 08:00
:Schedule_interval.2MO
:Required_database.SQLDBA
:Required_dbstatus.U
:Dependent_jobname.
:Dependent_jobresult.
:Average_runtime.00:53
:Last_jobstart.19961130 23:20
:Last_jobend.19961201 00:17
:Notify.MSTRUSR
:Execute.SQLREORG SQLDBA PUBLIC.DSQTSCT3
:Holiday_flag.
```

Figure 69. Example SQMJOB File

SQLMSTR JOBS File

The SQLMSTR JOBS file will contain information about all the product's jobs, including all current dependency status data. Each record contains all scheduling information for a single job. The SQLMSTR JOBS file is created and maintained on the service machine's 191 A-disk and should *never* be modified manually. Figure 70 provides the layout of parameters within each record of the file.

POSITION(S)	PARAMETER DESCRIPTION
1 - 8	Jobname
10 - 29	Server-machine
30	Server-machine availability flag
31	Holiday file processing flag
32	Job-status
34	Priority
36 - 49	Next-start
50	Start time exceeded flag
52 - 65	Window-end
67 - 71	Schedule-interval
73 - 80	Required-database
82	Required-dbstatus
83	Required-database availability flag
85 - 92	Dependent-jobname
94	Dependjob-result
96	Dependent job completed flag
97 - 101	Average-runtime
103 - 116	Last-jobstart
118 - 131	Last-jobend
133 - 192	Notify
194 - *	Execute

Figure 70. SQLMSTR JOBS File Layout

SQLMSTR TIMES File

The SQLMSTR TIMES file will contain an entry for each scheduled product job. The scheduled executable statement within the TIMES file will be EXEC SQMJOB, not the executable statement contained within the Execute parameter of the job itself. The EXEC SQMJOB statement of the TIMES file will include an action keyword and the name of the scheduled job, as shown in Figure 71 on page 182.

STARTJOB: The STARTJOB action keyword indicates that all job dependencies have been met and the job will be INITIATED by the product on the specified date and time. The QMFREORG job in the example SQLMSTR TIMES file has all dependencies satisfied and will be initiated at 11:00 p.m. on January 30th, 1997.

RESCHEDULE: The RESCHEDULE action keyword indicates that a job has at least one dependency that has not been met. If all dependencies are not met before the specified date and time, the product will automatically reschedule the job for the next available execution window using the Schedule-interval of the job. The SUMINDEX job in the example SQLMSTR TIMES file is waiting for at least one dependency to be met and will be rescheduled to execute at a later date and time if the dependency has not been met by 6:00am on February 25th, 1997. If all dependencies are met before that date and time, the product will change the SQLMSTR TIMES entry from SQMJOB RESCHEDULE to SQMJOB STARTJOB, with a new date and time for the job to be initiated.

WINDOWEND: The WINDOWEND action keyword indicates that a job is currently ACTIVE, and a notification message will be sent to the appropriate product users if the job is still active beyond the date and time specified for the Window-end parameter of the job. The MAINTJOB job in the example SQLMSTR TIMES file is scheduled to notify users at 8:00 a.m. on January 19th, 1997 that the job has run beyond its defined Window-end date and time. If the job completes before the specified value, the product will automatically change the WINDOWEND entry and reschedule the job for its next execution.

```
01/30/97 23:00:00 ........ EXEC SQMJOB STARTJOB QMFREORG
02/25/97 06:00:00 ........ EXEC SQMJOB RESCHEDULE SUMINDEX
01/19/97 08:00:00 ........ EXEC SQMJOB WINDOWEND MAINTJOB
```

Figure 71. Example SQLMSTR TIMES File

RUNTIMES File

The job scheduler keeps track of each job execution within a separate RUNTIMES file for each job. RUNTIMES files are maintained on the service machine's 191 A-disk with a filename equivalent to the jobname and a filetype of RUNTIMES.

Each time a job executes, the job scheduler will add another entry to the job's RUNTIMES file which records the start time, completion time, elapsed time (hours and minutes), and whether the job was successful or failed. This file is used by the job scheduler to calculate the Average-runtime value, using the last five successful entries within the file to update the average runtime of the job. The job scheduler automatically purges old entries from the RUNTIMES file, only maintaining the most recent 50 job executions.

Shown below is an example of a RUNTIMES file.

```
19970111 02:21 19970111 02:59 00:38 FAILED
19970111 23:30 19970112 00:42 01:12
19970118 02:10 19970118 03:27 01:17
```

Figure 72. Example RUNTIMES File

SQLMSTR HOLIDAYS File

The SQLMSTR HOLIDAYS file allows each installation to provide the job scheduler with information about specific days that should not be considered for job execution. For example, if the local installation is closed for an entire week in July, on Easter, and on Christmas Day, the job scheduler can be instructed to avoid scheduling jobs on any of these days. For each day that the job scheduler should avoid, an alternate day must be provided, which will be used instead.

When the job scheduler determines the correct date for the next execution of each job, the SQLMSTR HOLIDAYS file is referenced to determine whether the proposed date matches one of the locally established holidays. If a match is found, the specified alternate date is used. Note that the SQLMSTR HOLIDAYS file is used when Control Center reschedules each job using the Schedule-interval parameter. If the SQLMSTR HOLIDAYS file is updated after a job has already been scheduled for a given date, the job can still execute on the previously scheduled date, although the date can now be defined as a holiday. It is therefore important to update the SQLMSTR HOLIDAYS file as far in advance as is practical to avoid problems.

New with Release 6.1 of Control Center, a Holiday_flag flag has been added to the SQMJOB file to indicate if a job has been re-scheduled because of a holiday in the SQLMSTR HOLIDAYS file. If the Holiday_flag is set to Y, then the job will be scheduled to run on an alternative date. Both the holiday date and the alternative date are defined in the SQLMSTR HOLIDAYS file that is maintained on the CTRLCTR service machine's A disk.

After the job executes it will be rescheduled based on the original next-start date. A sample HOLIDAYS file (SQLMSTR $HOLIDAY) is provided on the code disk.

Additional processing includes:

1. Automatic re-scheduling when a new SQLMSTR HOLIDAYS file is received by the CTRLCTR service machine. If a new SQLMSTR HOLIDAYS file is sent to the CTRLCTR service machine, the scheduler will check all the currently scheduled (status "S") jobs against the new SQLMSTR HOLIDAYS file. If the job has not already been re-scheduled because of a holiday, the job will be re-scheduled to the alternative date.
2. If you indicate "SKIP" as the alternative date in the SQLMSTR HOLIDAYS file, the job will automatically be re-scheduled to run on the next normally scheduled date after the holiday.
3. When modifying a JOB through the job menu, an asterisk, "*", will appear next to the Next-start date to indicate that the job is scheduled to run on an alternative date.

   When viewing a JOB through the job menu, a "Rescheduled because of holiday" message will appear next to the Next-start date date to indicate that the job is scheduled to run on an alternative date.

If you modify an existing JOB through the menu by changing the the next-start date the holiday-flag will be set to blank. The Holiday_flag job parameter will not appear on the job menu and should not be changed by the user.

Figure 73 on page 184 provides an example of the SQLMSTR HOLIDAYS file. The first date on each line represents the holiday to be avoided and the second

date represents the alternate date for jobs to be scheduled. Note that the dates must be given in the form *YYYYMMDD*, where *YYYY* is a four-digit year, *MM* is a two-digit month, and *DD* is a two-digit day. Any characters beyond the second date on each line are considered as comments and will be ignored by the product. Also note that times should not be indicated, only dates.

```
19970101 19970102 * New Year's Day
19970328 19970406 * Good Friday
19970330 19970406 * Easter
19970526 19970527 * Memorial Day
19970704 19970705 * Independence Day
19970901 19970909 * Labor Day
19971127 19971126 * Thanksgiving
19971128 19971126 * Thanksgiving day 2
19971225 19971218 * Christmas
19980101 SKIP     * New Year's Day - skip all jobs
```

Figure 73. Example SQLMSTR HOLIDAYS File

Note in the sample SQLMSTR HOLIDAYS file that there are no restrictions on the alternate date provided for each holiday. For example, jobs scheduled for Good Friday (March 28th) and Easter (March 30th) will both be rescheduled for execution on Saturday, April 6th. Also, jobs scheduled for Labor Day, September 1st, will be rescheduled for one week later, September 9th, while jobs scheduled for Christmas Day, December 25th, will be rescheduled for one week earlier (December 18th).

There are two ways that the SQLMSTR HOLIDAYS file can be created and updated. One method is to stop the service machine, log onto it, create/modify the file (using XEDIT), then restart the service machine. Another method is to use the FILELIST **(F)** option under the General Control Center commands (G) option on the main menu. This option allows the file to be modified from your user ID using XEDIT and then migrate it to the service machine.

Problem Resolution

Correcting Active Job Status

If an executing job is interrupted by some means and is unable to end through the normal job completion process, the job status will not be updated. In effect, the job is *stuck* in an active status. After determining that the job is no longer active, you can use the modify option of the Job Schedule List tool to change the job status.

Resetting Average Runtime

The average runtime is automatically calculated by the product based on the last five successful runs. Except for your initial entry when defining the job, any value you enter in the Average-runtime entry field is ignored. An unusually long job runtime, possibly due to processing problems, can inflate the average runtime to the point where it impacts job scheduling. Correcting the average runtime requires manual update of the jobname RUNTIMES file. This should only be done by the Control Center Administrator.

Chapter 15. Tape Management Tool

Terminology Used in This Chapter

- The term *output media* refers to both *tape* (cartridge or reel-to-reel) and *disk* (CMS file) unless explicitly stated otherwise, as in "tape output medium".
- The terms *tape catalog* and *tapes file* refer to the same structure.
- The use of the term *tapes* when referencing the "tapes file" is not meant to exclude disk output medium.
- The use of the term *tapes* in the Control Center panel interface is not meant, in most cases, to exclude disk output medium.

Before continuing, you should review these topics:

- “Tape Management” on page 24.
- “Database TAPES File” on page 127.

Overview

Control Center uses the database TAPES file to maintain a list of output media assigned for usage during database archives, log archives, and trace activities. At database startup time and during processing of these activities, it references and updates this file as necessary. If Data Restore is enabled and used, then the TAPES file will also include media information for Data Restore BACKUP, Incremental BACKUP and TRANSLATE.

A separate database TAPES file is maintained by the product for each database machine it manages. TAPES files are generated during the database setup process and saved on the service machine's 191 A-disk as file *database* TAPES, where *database* is the database machine ID. Using the Tape Management tool, you can view and modify the database TAPES file.

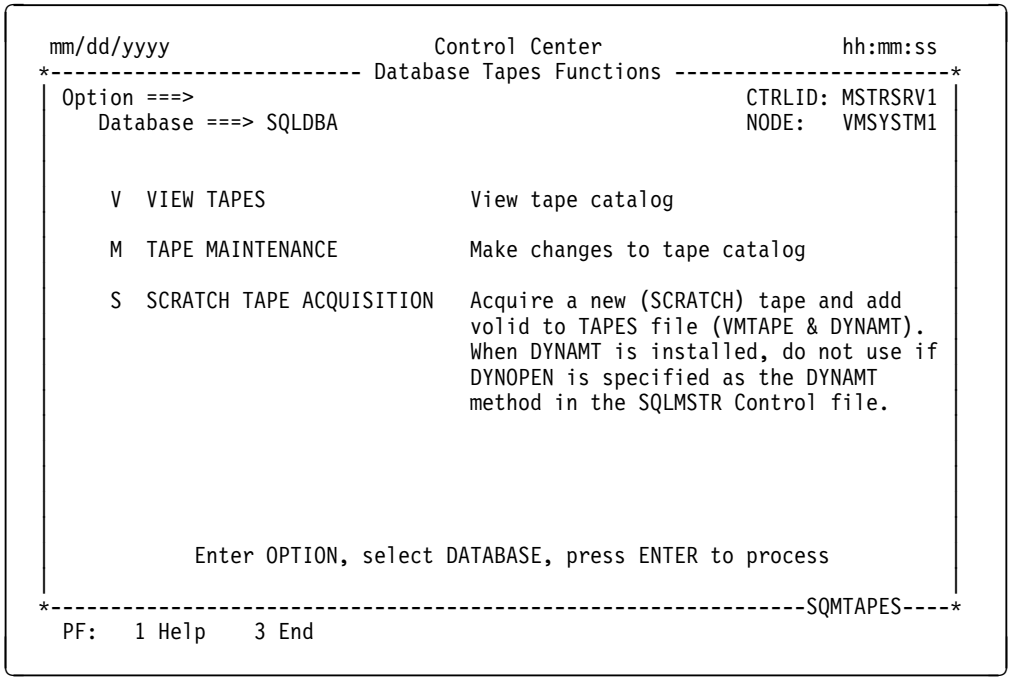


Figure 74. Tape Management Option Selection Panel

Shown in Figure 74 is the option selection panel of the Tape Management tool. The database you're currently working with is identified by your communication path and database settings shown near the top of the panel. The communication path can be changed using Option **C** from the Control Center Main Menu, whereas the database setting can be changed on this panel. For more information, see “Control Center Communication Path and Database Settings” on page 143.

VIEW TAPES (V)

Select this option to view the current database TAPES file.

The display panel used by the VIEW TAPES Option **V**, is similar to the entry panel used by the TAPE MAINTENANCE Option **M**, described next.

TAPE MAINTENANCE (M)

Select this option to modify the database TAPES file. Make required changes, paging forward and backward using PF7 and PF8, and then select PF10 to process. You will be prompted to confirm the update of the TAPES file on the service machine's 191 A-disk.

Usage Considerations:

Extreme caution should be exercised when changing the database TAPES file. Except for problem resolution, changes in the TAPES file should be limited to adding output media and deleting unused output media.

- Changing the output medium (from disk to tape or vice versa) will require updating the Archiving/Tracing parameters in the database parameter file using the Database Parameters tool. Refer to “Archiving/Tracing Parameters” on page 220. All changes will take effect during the next database startup.

- The product issues FILEDEFs and LABELDEFs for database archives at database startup time. Changes to database archive output medium will not take effect until the next database startup.
- Changes to log archive and trace activities take immediate effect.
- It references and updates the STATUS field during processing. Except for problem resolution, you should not modify STATUS values.

Note:

Under no circumstances should you update the TAPES file while a database archive, log archive, or trace activity is in progress. The product references and updates the TAPES file during these activities. Your changes may corrupt status information required by Control Center, which can lead to serious errors and loss of data.

```
 mm/dd/yyyy                    CONTROL CENTER                          hh:mm:ss
*------------------------- Database TAPES File Update ----------------------*
| Command ===>                                                CTRLID: MSTRSRV1 |
|  Database => SQLDBA                                          NODE:  VMSYSTM1 |
|                                          VOLID/                             |
| SERIES  TYPE     DATE   TIME     STATUS  FILENAME  FILETYPE  FM   CUU       |
|  100    ARCHIVE  00000  00:00:00 FILDEF  VOL100                            |
|  100    LOG      00000  00:00:00 UNUSED  SQLDBA1 03269101 F 501            |
|  100    LOG      00000  00:00:00 UNUSED  SQLDBA1 03269102 F 501            |
|  200    ARCHIVE  97086  11:09:58 FILLED  VOL200                            |
|  200    LOG      97086  11:34:56 FILLED  SQLDBA1 03279103 G 502            |
|  200    LOG      00000  00:00:00 UNUSED  SQLDBA1 03269105 G 502            |
|  300    ARCHIVE  97086  11:49:02 FILLED  VOL300                            |
|  300    LOG      00000  00:00:00 UNUSED  SQLDBA1 03269106 H 503            |
|  300    LOG      00000  00:00:00 UNUSED  SQLDBA1 25039107 H 503            |
|                                                                            |
|                                                                            |
|   Make changes, place D in SERIES to DELETE , press PF10 to process        |
|                            Page 1     of 1                                 |
|                                                                            |
*----------------------------------------------------------------SQMTP20-----*
 PF:  1 Help   3 End    4 Add Tape   7 Bkwd     8 Fwd     10 Process updates
```

Figure 75. Tape Management Tool, Tape Maintenance Panel

Table 10. TAPES File Format

Value	Description
SERIES	A three-digit number indicating the output media series. Output media with the same series number are considered by Control Center to belong together as a single logical group.
TYPE	Assigns output media to a specific usage. Possible values: **ARCHIVE** Use for full database archiving activities. **ssINCBKd** Use for each Incremental Backup. "ss" is the subseries number and "d" is blank (indicating the primary) or "2" (indicating the secondary) Incremental Backup. **LOG** Use for database log archiving activities. **TRACE** Use for database tracing activities. **BACKUP** Use for primary user archives using Data Restore BACKUP. **BACKUP2** Use for secondary user archives using Data Restore with dual BACKUP. **TRANS** Use for translation of database archives into Data Restore BACKUP format. **TRANSDSK** Use for the disk location of the 3 Data Restore work files needed for translation (SYS0001, DIRWORK, & HEADER). **UNLOADID** User generated id for unloading DBSPACES using Data Restore. Used in the UNLTAPES file only.
DATE	The Julian date (*yyddd*) when output media was last used by Control Center. Should be equal to 00000 when tape is unused.
TIME	Time (*hh:mm:ss*) when output media was last used by Control Center. Should be equal to 00:00:00 when tape is unused.
STATUS	Control Center uses this field to track the status of the output media. Common status values include: **UNUSED** Output media is not being used. **FILDEF** Output media currently being used. **MOUNTD** Output media is currently mounted. **FILLED** Output media has been used.
VOLID	Tape volume serial number if output media is TAPE.
FILENAME	CMS filename if output media is DISK.
FILETYPE	CMS filetype if output media is DISK.
FM	CMS filemode (disk access mode) if output media is DISK.
CUU	*Virtual* disk address (CUU) if output media is *DISK* and *TYPE* is ARCHIVE, LOG or TRACE. *Real* disk address (CUU) if output media is *DISK* and *TYPE* is a *Data Restore* function (*TRANSLATE, BACKUP2, INCBK, BACKUP, or UNLOAD*).
CI	Century Indicator (1 = 20th Century, 2 = 21st Century) This is not displayed on the panels.

Adding a Record

Select **PF4** key to add a new record to the end of the list, as shown below. You must provide a valid SERIES and VOLID/FILENAME by typing over the default values before the new record will be accepted by the product. The list will be re-sorted after you process your changes **(PF10)**.

```
 300    ARCHIVE  97086  11:49:02 FILLED   VOL300
 300    LOG      00000  00:00:00 UNUSED   SQLDBA1 03269106 H 503
 300    LOG      00000  00:00:00 UNUSED   SQLDBA1 25039107 H 503
 ADD    ARCHIVE  00000  00:00:00 UNUSED


  Make changes, place D in SERIES to DELETE , press PF10 to process
                           Page 1     of 1
*-------------------------------------------------------------SQMTP20-----*
 PF:  1 Help   3 End    4 Add Tape   7 Bkwd     8 Fwd     10 Process updates
```

Figure 76. Adding a Record to the Database TAPE File

Deleting a Record

To delete an entire record, type a `D` in the SERIES field of the target record. The Control Center identifies deleted records by changing the SERIES value to DEL, as shown below.

```
 200    LOG      00000  00:00:00 UNUSED   SQLDBA1 03269105 G 502
 300    ARCHIVE  97086  11:49:02 FILLED   VOL300
 300    LOG      00000  00:00:00 UNUSED   SQLDBA1 03269106 H 503
 DEL    LOG      00000  00:00:00 UNUSED   SQLDBA1 25039107 H 503

  Make changes, place D in SERIES to DELETE , press PF10 to process
                           Page 1     of 1
*-------------------------------------------------------------SQMTP20-----*
 PF:  1 Help   3 End    4 Add Tape   7 Bkwd     8 Fwd     10 Process updates
```

Figure 77. Deleting a Record in the Database TAPE File

SCRATCH TAPE ACQUISITION (ST)

This option acquires a scratch tape and automatically adds it to the database TAPES; it is applicable only when using VMTAPE or DYNAMT.

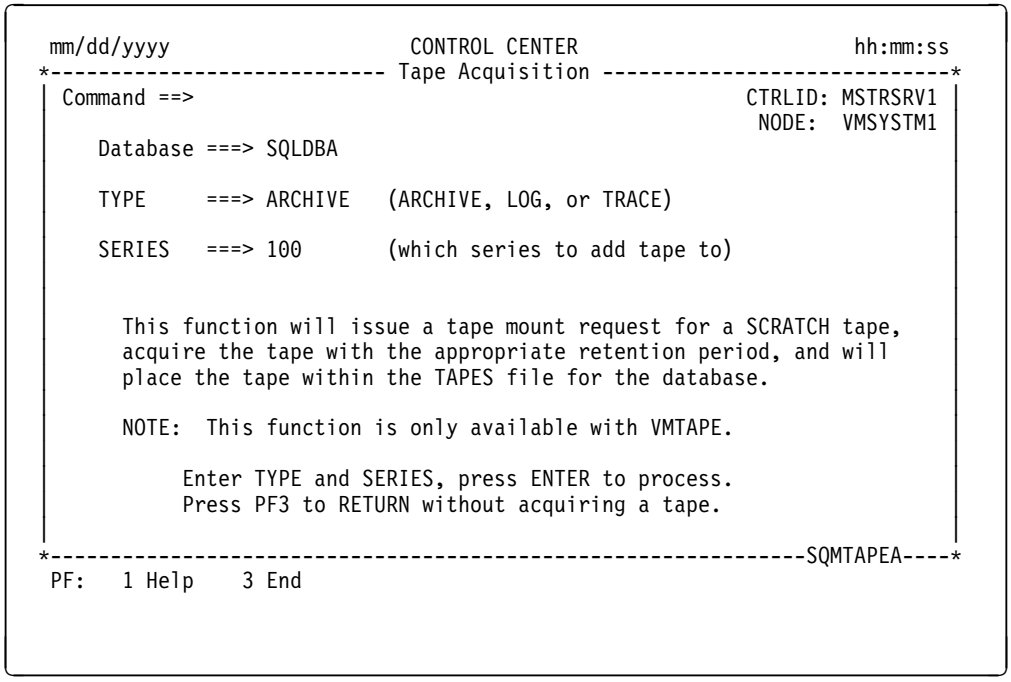


Figure 78. Tape Management Tape Acquisition Panel

Entry Field	Description
TYPE	Assigns tape to a specific usage. Possible values: **ARCHIVE** Use for full database archiving activities. **LOG** Use for database log archiving activities. **TRACE** Use for database manager for tracing activities.
SERIES	Tape will be added to this series in the TAPES file.

DYNAMT Tape Usage

DYNAMT has two mount commands available to handle tape processing. DYNOPEN lets users create and request tapes based upon data set name, while DYNMOUNT lets users create and request tapes from a user's personal tape library. Both mount methods are supported by this version of Control Center, however, the information below is important based upon the DYNAMT method you choose for tape usage.

DYNOPEN

To use DYNOPEN, the steps below are required for correct operation:

1. DYNOPEN must be entered as the DYNAMT method in the SQLMSTR Control file.

2. Data set names are required for all DYNOPEN tape processing. Enter the appropriate dataset name within the Tape and Data Restore Parameter sections of the Database parms file.
3. The Scratch-tape-option value must be set to "Y" (YES) in the Tape Parameters section of the Database Parms file.
4. The Logtape-premount value must set to "Y" (YES) in the Tape Parameters section of the Database Parms file.

DYNMOUNT

To use DYNMOUNT, the following is required for correct operation:

1. You can either enter DYNMOUNT or leave the DYNAMT method blank in the SQLMSTR Control file.
2. Data set names are not required DYNMOUNT tape processing. If your location has implemented the use of data set names, then enter the appropriate dataset name within the Tape and Data Restore Parameter sections of the Database parms file.
3. The Scratch-tape-option value must be set to "N" (N0) in the Tape Parameters section of the Database Parms file.
4. The Logtape-premount value must set to "Y" (YES) in the Tape Parameters section of the Database Parms file.

Data Restore Tape Types

Tape TYPE

When discussing tape files, TYPE will refer to the indicator in the database TAPES file that describes the function for which the output tape/file is intended. TYPE does NOT describe the physical characteristics of the media (for example, 3820 tape).

If Data Restore is enabled and used, then the TAPES file will also include media information for Data Restore BACKUP, Incremental BACKUP and TRANSLATE.

Backup Media

The tapes for the "BACKUP" tape type will be used for the primary Data RestoreBACKUP. If "Dual-backup = Y" in the database PARMS file, the secondary BACKUP will use the tapes indicated by the BACKUP2 tape type. An archive or a BACKUP executed with the FULL option does not affect the tape type used for the function.

Translate Media

Output media for TRANSLATE tapes and work files resides in the database TAPES file in the same series as the associated archive. One TRANSLATE is allowed per database TAPES file series. The file or tapes to be used for the TRANSLATE are indicated by using a type of TRANS.

While there can be one or more tapes designated for the TRANSLATE (TRANS), only one entry (TRANSDSK) will be used to indicate where the work files will be kept.

The three work files created by Data Restore. (SYS0001, DIRWORK, HEADER) can only be written to disk; these files will all be written to the same disk. Because all the work files are written to the same disk, only one entry in the database TAPES file (TRANSDSK) will be used to indicate the name and location of the work files. The TRANSDSK entry includes the FILENAME and link address of the work files. Only the file name will be significant for the TRANSDSK entry. The actual filetype used will correspond to the type work file that will be FILEDEFed and LABELDEFed. The file mode will be determined at the time of the TRANSLATE. This strategy will allow more than one TRANSLATE work file output on a disk. See Figure 80 for an example of the filedefs specified.

TRANSLATE may require more tapes than those used in the archive. If predefined tapes are used in the database TAPES file, be sure to include more tapes to handle the TRANS entry.

```
100  TRANS    00000  00:00:00 UNUSED SCRATCH
100  TRANSDSK 00000  00:00:00 UNUSED DB2VM510 TRANSDSK *  400
```

Figure 79. Example of TAPES file entries for TRANSLATE

The preceding figure shows how the TRANSDSK entry in the database TAPES file will be used when performing a FILEDEF on the translate work files.

```
FILEDEF SYS0001 DISK  DB2VM510 SYS0001 T ( RECFM F BLOCK 4096
FILEDEF HEADER  DISK  DB2VM510 HEADER  T ( RECFM F BLOCK 4096
FILEDEF DIRWORK DISK  DB2VM510 DIRWORK T ( RECFM F BLOCK 0512
```

Figure 80. Example FILEDEFS for TRANSDSK entry

The TRANSLATE (new backup) tapes/file will use cuu 185, and the archive tapes (ARIARCH) will use cuu 181.

Because the TRANSLATE output tapes are taken from the same series as the archive being translated, the database TAPES file must have a series that matches the series in the ARCHHIST file.

For example, if you have a series 100 through 600 and you delete series 600, you can still have an entry for series 600 in the ARCHHIST file. You can TRANSLATE the archive for the deleted series 600 because the tapes still exist, but there will be no entry in the TAPES file to indicate what tapes to use.

Incremental Backup Media

DB2 Server for VM treats an Incremental Backup as a normal USER ARCHIVE. However, an Incremental Backup is not useful unless the file or tapes used for the reference Full Backup are still valid. If a Full Backup's tapes are overwritten by a subsequent BACKUP, then all the Incremental Backups that reference that Full Backup will no longer be usable.

When Control Center manages archive and log archive output media in the database TAPES file, it uses a series which holds one archive and one or more log archives. A similar strategy is used to maintain the relationship between the Full Backup and its one or more Incremental Backups.

In order to prevent tapes from being overwritten prematurely, each series in the database TAPES file will also include INCBK and INCBK2 tape types for one or

more Incremental Backups. The Incremental Backup is grouped together with the series used for the Full BACKUP/archive. In this way, a Full Backup or Incremental Backup is not overwritten until the series is used again for a new normal or Full Backup. However, Incremental Backup output is different than log archive output in several ways:

1. Multi-volume Incremental Backups are allowed.
2. Dual Incremental Backups are allowed.
3. The number of tapes required to hold the output from each Incremental Backup can increase as the number of changed pages in the database increases.

In order to uniquely identify the tapes that belong to an Incremental Backup within a series, each set of Incremental Backup tapes are identified with their own tape type.

Incremental Backup INCBK Tape Type: The INCBK tape type is used to identify the set of tapes/file to be used for each Incremental Backup within a series.

The INCBK will be in the form "ssINCBKd" where:

ss is the subseries number between 01 and 99 that will be used to identify all the tapes available for a single Incremental Backup within the archive-series. For example, all the tapes for the first Incremental Backup in series 100 would be identified by the tape type 01INCBK. The second set of Incremental Backup tapes would be identified by 02INCBK.

After a Full Backup successfully executes using series 100, the first Incremental Backup would use the 01INCBK tapes in archive-series 100, the second Incremental Backup would use 02INCBK, etc.. The subseries numbers must be sequential and start with 01.

INCBK Indicates the tapes are for an Incremental Backup.

d indicates the tapes are reserved for either the primary or secondary backup of a DUAL backup. If the value is blank (01INCBK), then the tapes will be used for the primary Incremental Backup. If the value is 2 (01INCBK2), then the tapes will be used for the secondary (dual) Incremental Backup. If "Dual-backup = N" in the database PARMS file, then only the primary tapes will be used.

Figure 81 on page 194 shows part of a logmode A database TAPES file using Incremental Backup. Series 100 consists of a set of dual backup entries for primary BACKUP and secondary backup and three sets of dual Incremental Backup. Each tape belonging to an Incremental Backup within series 100 is grouped together by the subseries prefix. If the database was using logmode L, then a log archive would have been performed prior to each BACKUP and and Incremental Backup.

```
                100  ARCHIVE  00000  00:00:00 UNUSED VM0010
                100  ARCHIVE  00000  00:00:00 UNUSED VM0011
1               100  BACKUP   98032  16:37:21 FILLED VM0012
                100  BACKUP   98032  16:41:08 FILLED VM0012
                100  BACKUP2  98032  16:37:21 FILLED VM0014
                100  BACKUP2  98032  16:41:08 FILLED VM0015
                100  TRANS    00000  00:00:00 UNUSED VM0040
                100  TRANS    00000  00:00:00 UNUSED VM0041
                100  TRANSDSK 00000  00:00:00 UNUSED TRAN100 TRANSDSK * 196
2               100  01INCBK  98033  17:50:01 FILLED VM0016
                100  01INCBK2 98033  17:50:02 FILLED VM0017
3               100  02INCBK  98034  17:30:01 FILLED VM0018
                100  02INCBK  00000  00:00:00 UNUSED VM0019
                100  02INCBK2 98034  17:30:02 FILLED VM0020
                100  02INCBK2 00000  00:00:00 UNUSED VM0021
4               100  03INCBK  98035  17:50:10 FILLED VM0022
                100  03INCBK  98035  17:50:15 FILLED VM0023
                100  03INCBK  00000  00:00:00 UNUSED VM0024
                100  03INCBK2 98035  17:50:11 FILLED VM0025
                100  03INCBK2 98035  17:50:17 FILLED VM0026
                100  03INCBK2 00000  00:00:00 UNUSED VM0027
                200  ARCHIVE  00000  00:00:00 UNUSED VM0028
                200  ARCHIVE  00000  00:00:00 UNUSED VM0029
5               200  BACKUP   98036  16:37:21 FILLED VM0030
                200  BACKUP   98036  16:46:58 FILLED VM0031
                200  BACKUP2  98036  16:37:21 FILLED VM0032
                200  BACKUP2  98036  16:41:08 FILLED VM0033
                200  TRANS    00000  00:00:00 UNUSED VM0042
                200  TRANS    00000  00:00:00 UNUSED VM0043
                200  TRANSDSK 00000  00:00:00 UNUSED TRAN200 TRANSDSK * 196
6               200  01INCBK  98037  17:50:01 FILLED VM0034
                200  01INCBK2 98037  17:50:02 FILLED VM0035
                200  02INCBK  98038  17:30:01 FILLED VM0036
                200  02INCBK  00000  00:00:00 UNUSED VM0037
                200  02INCBK2 98038  17:30:02 FILLED VM0038
                200  02INCBK2 00000  00:00:00 UNUSED VM0039
                200  03INCBK  00000  00:00:00 UNUSED VM0040
                200  03INCBK  00000  00:00:00 UNUSED VM0041
                200  03INCBK  00000  00:00:00 UNUSED VM0042
                200  03INCBK2 00000  00:00:00 UNUSED VM0043
                200  03INCBK2 00000  00:00:00 UNUSED VM0044
                200  03INCBK2 00000  00:00:00 UNUSED VM0045
```

Figure 81. Example of a Logmode A Database TAPES File Using Incremental Backup

1 — On Sunday a dual Full Backup was performed using series 100.

2 — On Monday, an Incremental Backup was performed using the tapes in series 100 and subseries 01 (01INCBK).

3 — On Tuesday, an Incremental Backup was performed using subseries 02 (02INCBK).

4 — On Wednesday, two tapes were used for both the primary and secondary Incremental Backup using subseries 03 (03INCBK and 03INCBK2).

5 — On Thursday, a new Full Backup was executed using series 200.

6 — On Friday, a new set of Incremental Backups are executed in series 200 starting with subseries 01 (01INCBK).

The ARCHIVE and BACKUP2 entries in the database TAPES file do not do not distinguish between FULL and normal backups because the tapes can be used for either. After the backup is completed, the ARCHHIST file will indicate if the full option was used.

Number of LOG ARCHIVE entries in a series

There must be enough log archive entries in each Logarch-series to handle all log archives between Full Backup or ARCHIVE FULL in the Incbackup-series. If the database is logmode L, then a log archive is performed before each USER ARCHIVE (BACKUP or Incremental Backup).

Incbackup-series and Tape Rotation with Incremental Backup: Incremental Backup has its own 'Incbackup-series' in the database PARMS file to control which set of tapes will be used for the next Incremental Backup.

The Incbackup-series in the database PARMS file indicates the series and subseries in the database TAPES file that will be used to supply tapes for the next Incremental Backup. The Incbackup-series is presented in the form "series.subseries" where:

series The series indicates the archive-series of the last archive or BACKUP. The series portion of the Incbackup-series will be the same as the Logarch-series and is incremented after any archive or BACKUP completes. The series is incremented and the subseries reset to 01 each time any normal BACKUP, Full Backup, normal archive or ARCHIVE FULL completes.

subseries The subseries indicates which set of Incremental Backup tapes to use within the series. The subseries will be a number between 01 and 99. The first subseries after an archive or backup will always be 01. For example, Incremental Backup series 100.01 indicates that the first set of Incremental Backup tapes (01INCBK) in series 100 are to be used.

The subseries is incremented **only** after an Incremental Backup successfully completes.

The Archive-series, Logarch-series and Trace-series are **not** incremented when an Incremental Backup completes.

The Incbackup-series will be maintained in the database PARMS file even when "Incbackup-enabled = N".

After an Incremental Backup completes, the subseries will be incremented even though there may not be an INCBK entry in the database tapes file for that Incbackup-series.

The user must setup the database TAPES file so that there will be enough entries of each tape type to accommodate the expected backup/archive workload. However, there will be times when a user has performed more than the usual number of Incremental Backups, and there may not be anymore INCBK entries in the database TAPES file to satisfy a regularly scheduled Incremental Backup. Control Center provides several options to ensure that archiving can continue.

If another Incremental Backup is requested when there is no INCBK entry for the current Incbackup-series then one of the following actions will occur:

1. If you are using SCRATCH tapes for your Incremental Backup, then Control Center will automatically add a new INCBK SCRATCH entry in the database

TAPES file for the current Incbackup-series. The Incremental Backup process will continue.

2. If the "Auto-full" option in the database PARMS file is set to Y (Yes), then Control Center will automatically switch the Incremental Backup to a Full Backup. The Auto-full parameter will be ignored if the AUTOFULL or NOAUTOFULL command option is used.
3. If you have indicated the AUTOFULL command option when setting up the Incremental Backup through the menu or on the command line, then Control Center will switch the Incremental Backup to a Full Backup regardless of the "Auto-full" value in the database PARMS file.
4. If you have indicated the NOAUTOFULL command option then Control Center will

   not switch the Incremental Backup to a Full Backup regardless of the "Auto-full" value in the database PARMS file. The Incremental Backup will be cancelled.
5. If none of the above options are chosen, the Incremental Backup will be cancelled.

If the Incremental Backup is cancelled, you can use the Tape Maintenance menu to add the new INCBK entry and then request another Incremental Backup.

The user must understand the implications of switching an Incremental Backup to a Full Backup. A Full Backup can take considerably longer than an Incremental Backup and can affect the execution of other scheduled database maintenance activity.

There can be times when switching or not switching makes more sense. The AUTOFULL/NOAUTOFULL command option offers the user the flexibility to control the Incremental Backup without changing the overall Incremental Backup strategy. In order to keep the Incremental Backup tapes together with its reference backup, Control Center will increment the Incbackup-series whenever a BACKUP or archive completes.

A BACKUP is a User Archive and when the database is using logmode L, log archives will be initiated by the database.

Number of LOG ARCHIVE entries in a series

There must be enough log archive entries in each Logarch-series to handle all log archives between Full Backup or ARCHIVE FULL in the Incbackup-series. If the database is logmode L, then a log archive is performed before each USER ARCHIVE (BACKUP or Incremental Backup).

Chapter 16. Database Operation Interface Tools

Overview

The Operator Command Interface tools provide an interface between your virtual machine and a DB2 Server for VM database for performing database operator commands including SHOW commands, FORCE, COUNTER, RESET, and TRACE commands, and operator commands included with the VMDSS feature.

Although the Operator Command Tool and the Direct Operator Command Tool allow you to perform many of the same operator commands, there are key functional and operational differences which set them apart.

Table 11. Operator Command Tool/Direct Operator Command Tool Comparison

Functional Differences	**Operator Command Tool**	**Direct Operator Command Tool**
Issue operator commands to remote databases	YES	NO
Supports FORCE, TRACE, and RESET operator commands	YES	NO
Operational Differences		
Control Center Panel Mode Interface	YES [8]	NO
Requires Control Center authority	YES	NO
Requires a DB2 Server for VM agent	NO	YES

- Before using the trace facility through Control Center, you should review the database Archiving/Tracing parameters and Startup parameters. See “About the Database Parameters Tool” on page 213.
- If outputting trace information to tape, you should also review the database TAPE parameters in the same section.

Operator Command Tool

The Operator Command Tool provides an interface between your virtual machine and a local or remote database manager for performing database operator commands including SHOW commands, FORCE, COUNTER, RESET, and TRACE commands, and operator commands included with the VMDSS feature.

[8] The Operator Command Tool can also be invoked from CMS using the Control Center command mode interface. Refer to Appendix G, “Command Mode Interface” on page 523.

```
 mm/dd/yyyy                      CONTROL CENTER                       hh:mm:ss
*--------------------------- DBA Operator Commands --------------------------*
| Option ===>                                                CTRLID: MSTRSRV1  |
|    Database ===> SQLDBA                                    NODE:   VMSYSTM1  |
|                                                                              |
|     ************************ SHOW COMMANDS ************************         |
|   A  ACTIVE                                 AD ADDRESS module-name           |
|   B  BUFFERS                                C  DBCONFIG                      |
|   CN CONNECT                                D  DBSPACE n                     |
|   E  DBEXTENT                               I  INVALID                       |
|   ID INDOUBT                                IP INITPARM                      |
|   LA LOCK ACTIVE                            LD LOCK DBSPACE ALL|n           |
|   LG LOCK GRAPH uid|AGENT n                 LM LOCK MATRIX                   |
|   LU LOCK USER uid|AGENT n|ALL              LW LOCK WANTLOCK uid|AGENT n|ALL |
|   L  LOG                                    LH LOGHIST ALL|n|SERVICE        |
|   LN CRR LOGNAMES                           P  POOL ALL|SUMMARY|DELETED|    |
|   PR PROC  *|proc-name AUTHID               SP PSERVER  *|GROUP             |
|   S  SYSTEM                                 SG STORAGE                       |
|   U  USERS                                                                   |
|                                                                              |
|      Enter OPTION and PARMS, select DATABASE, press ENTER to  process        |
|                                                                              |
*------------------------------------------------------------------SQMOP10------*
 PF:   1 Help    3 End     8 Fwd (More & VMDSS)
```

Figure 82. DBA Operator Commands Panel

The DBA Operator Commands panel is shown in Figure 82. Additional options, which are applicable only if your database includes the VMDSS feature, are presented on a separate panel accessed by pressing **PF8** key.

The database you're currently working with is identified by your communication path and database settings shown near the top of the panel. The communication path may be changed using Option **C** from the Control Center Main Menu, whereas the database setting may be changed on this panel. For more information, see “Control Center Communication Path and Database Settings” on page 143.

Commands are issued against the specified database by entering the one or two-character command abbreviation along with required parameters, if any, and pressing the ENTER key. For example, the SHOW ACTIVE command can be issued by entering **A** in the option input field and pressing ENTER. The database response will be presented to you in a *response* panel (see “Response Panels” on page 146).

Refer to the *DB2 Server for VSE & VM Operation* manual for a complete description of each operator command.

The VMDSS DBA Operator Commands panel is shown in Figure 83 on page 199. This screen is presented on a separate panel when **PF8** key is pressed from the main DBA Operator Commands panel.

```
 mm/dd/yyyy                      CONTROL CENTER                       hh:mm:ss
*--------------------------- DBA Operator Commands ---------------------------*
| Option ===>                                                 CTRLID: MSTRSRV1  |
|    Database ===> SQLDBA                                     NODE:   VMSYSTM1  |
|                                                                               |
|      ******************** MISCELLANEOUS COMMANDS ********************        |
|    F  FORCE uid|AGENT n (DISABLE)  Disable is optional                        |
|    CO COUNTER *|name                      R  RESET *|name                     |
|    RD RESET INDOUBT *|ids                 SI SET PARAMETER name newval       |
|    TS TRACE START                         TO TRACE OFF  NOCL                  |
|    PS START PROC *|proc-name|authid       PO STOP PROC *|proc-name|authid     |
|    VS START PSERVER group|name            VO STOP PSERVER group|name         |
|    RN RESET CRR LOGNAMES                                                      |
|                                                                               |
|      ************************ VMDSS COMMANDS *************************        |
|    CI COUNTER INTERNAL *|name             RI RESET INTERNAL *|name           |
|    CP COUNTER POOL *|n|DIR|UNMAPPED       RP RESET POOL *|n|DIR|UNMAPPED     |
|    SP SET POOL n1|n1-n2 DSn STR|SEQ       SS SET SAVEINTV n                  |
|    WT SHOW TARGETWS                                                           |
|                                                                               |
|     Enter OPTION and PARMS, select DATABASE, press ENTER to  process          |
|                                                                               |
*-------------------------------------------------------------------SQMOP15-------*
PF:   1 Help    3 End     7 Bkwd    8 Fwd
```

Figure 83. VMDSS DBA Operator Commands panel

For more details on the specific VMDSS operator commands, refer to the *DB2 Data Spaces Support* manual.

Database Tracing

There are two ways to initiate the trace facility under Control Center: during database startup or by issuing the TRACE operator command. Both methods are controlled by the database Tracconv, Tracdbss, Tracdrrm, Tracdsc, Tracebuf, Tracrds, Tracstg, and Tracwum parameter settings. These parameters allow you to specify what database operations are to be traced, and at what levels. The section “Database Startup Parameters” on page 222 describes how to set these parameters.

How the trace facility is activated is dependent on the database Tracing parameter. If Tracing is set ON, the trace facility will be activated at database startup. If set OFF, it can be activated on demand by issuing the TRACE START command. In either case, tracing can be turned off by issuing the TRACE OFF command.

The output media for trace information is defined by the database Archiving/Tracing parameter settings. See “Archiving/Tracing Parameters” on page 220.

Advantages of Use

Using the Operator Command Interface provides you with several advantages over using other database interface products such as ISQL.

- Operator commands may be issued to local and remote databases from a single console.
- Control Center interfaces with a database through the database console and therefore does not use a database agent. This is important when the database is busy, experiencing intensive database locking, or when all agents are active.

- The FORCE command, which is only available from the database console, may be issued to a local or remote database using Control Center.

Direct Operator Command Tool

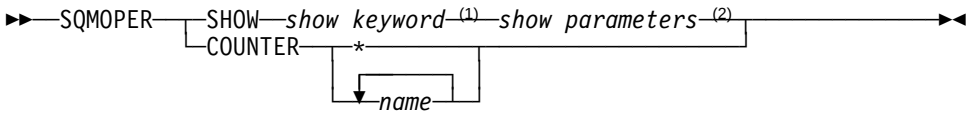
The Direct Operator Command Tool allows you to issue database COUNTER and SHOW commands to a database from the CMS command level. Results are displayed at your console.

Before Issuing SQMOPER

Before issuing the SQMOPER command, these steps must be completed:

- The database production disk must be linked and accessed.
- The Control Center disk must be linked and accessed.
- The database must be started.
- Either the DBINIT EXEC or SQLINIT EXEC provided with the Control Center and DB2 Server for VM products, respectively, must be run to establish the required links and define the name of the database machine.

SQMOPER



Notes:

1 ACTIVE, ADDRESS, BUFFERS

2 Parameters may or may not be applicable for the specified *show keyword*.

Figure 84. Sample Output from SQLOPER SHOW ACTIVE Command

The SQMOPER command issues a database COUNTER or SHOW command to the database machine. Output is displayed by placing you in the XEDIT environment, editing the file SQMOPER OUTPUT; see Figure 85 on page 201. The first line of output is generated by Control Center; it includes the name of the database machine. Output from the database is shown on the remaining lines.[9]

[9] If no database output is shown check the COUNTER or SHOW command for correct syntax.

```
 SQMOPER  OUTPUT   A1  V 80  Trunc=80 Size=8 Line=0 Col=1 Alt=0
====>
1=HLP 2=SPL 3=FILE 4=Q  5=TAB  6=SPLIT 7=U20 8=D20 9=JOIN 10=R30 11=L30 12=CC1
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
00000 * * * Top of File * * *
00001  Database: SQLDBA                                  03/26/97 10:24:30
00002
00003  Status of agents:
00004     Checkpoint agent is not active.
00005     User Agent: 1 User ID: MSTRUSR1 is R/O  APPL  38F0
00006        Agent is processing an operator command.
00007     4 agent(s) not connected to an APPL or SUBSYS.
00008  ARI0065I Operator command processing is complete.
00009 * * * End of File * * *
```

Figure 85. Sample Output from SQMOPER Command

Operational Note: The full power of XEDIT is available to you while you view the output, including the XEDIT subcommand FILE. However, the SQMOPER OUTPUT file is automatically erased when you exit from XEDIT. Therefore, to save the displayed output you must issue the XEDIT FILE subcommand using a different file name.

Chapter 17. Database Status Tool

Overview

Control Center status tools provide you with current detail and summary status information pertaining to your database machine. The status information provided will help you to quickly determine the condition of your database and whether or not any action is required. In addition, status information can be summarized to single panel displays for quick systemwide assessments of multiple databases residing on one or more systems managed by one or more Control Center service machines.

Database Status Display Options

The Database Status selection panel shown in Figure 86 on page 204 gives you four options for displaying database status information. The options will allow you to retrieve status for one database or for a group of databases. Option **D**, for example, will provide you with the status for a single database, whereas Option **A** will provide you with status information for all databases listed in the SQLMSTR DIRECTRY file (refer to Chapter 11, “Managing the Environment” on page 113 for details regarding this file). These options were developed to meet a wide range of operational support requirements. For example, an operations support staff responsible for overall systems availability might be more interested in Option **A** for a total systems status summary, whereas a database administrator might be more interested in Option **D**, providing information about a specific database.

The status information provided by each Option (D,S,N,A) is the same, with the difference being the specific database or list of databases displayed.

Display Status for a Database (D)

Provides detailed status information for a specific database. From this panel you will be able to quickly determine the current condition of your database, as well as determine if any specific action is required. Additional panel options (Select and Query) will give you quick access to other product tools that can provide you with more detailed status information, as well as operational control tools (database starting, termination, operator commands). Refer to the sections below for more information regarding the Select and Query options.

Display Status by Control Center Machine (S)

With this option you can quickly review the status of all your databases controlled by a specific service machine. The product's entry field can be typed over to quickly request status information from other service machines.

Display Status by System Node (N)

Provides status information for all databases that are managed and controlled by all service machines on a specified system node. To change the system node, simply tab over to the system node entry field and type in the new system node.

Display Status Using Database List (A)

This option provides status information for all databases that are identified in your SQLMSTR DIRECTRY file or another specified filename and filetype in the same format.

Specifying a Specific List of Databases to Status: As shown in Figure 86, the SQLMSTR DIRECTRY file indicated for option A can be changed to another CMS file. The format of this file *must* match that of the SQLMSTR DIRECTRY file, as documented in Chapter 11, “Managing the Environment” on page 113. In this file you would include a list of all database machines that you want status information about.

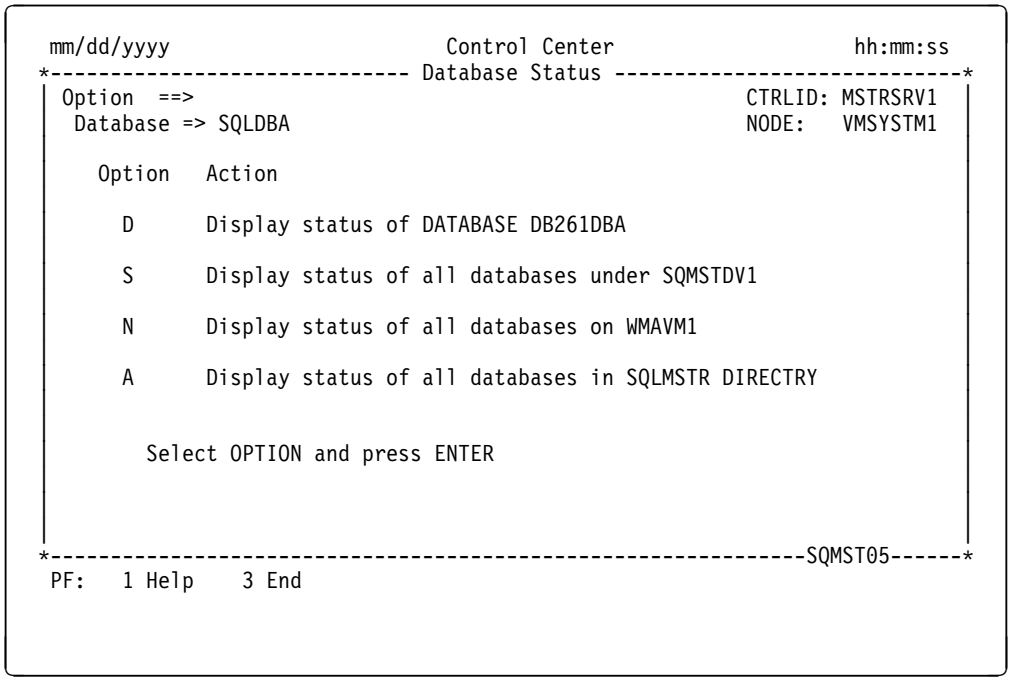


Figure 86. Database Status Tool, Option Selection Panel

Who is Authorized to Use the Status Display Panels

Retrieval of database status information requires User, Operator, or greater (DBA or Control Center Administrator) authorization for each database that you are requesting status information about. Once the status is displayed, additional panel options (database startup, termination, ...) may require DBA authorization for each database selected. Adding Control Center authority for a database can be done using the Database Parameters tool. Refer to “About the Database Parameters Tool” on page 213.

Advanced Usage: The product provides a default set of tools that are accessible by User, Operator, or DBA-level authorizations. These defaults can be changed so as to add or delete User, Operator, or DBA authorizations to specific tools. For more details on how your authorization levels can be modified, refer to Appendix E, “Authorizations” on page 511.

About the Master Database Status Display Panel

The Master Database Status panel (format 1), shown in Figure 87, provides status summary information for databases as specified by your selection on the previous panel (Figure 86 on page 204).

The primary display panel (format 1) will provide you with the current status of each database, the date and time when the status had last changed, the date and time when the last message was received by the product from the database, the date and time when the database last started, and the time/result of the last SHOW LOG, with the LOGPCT field calculated as the amount of the current log being used divided by the database's archive threshold (LOGUSED / ARCHPCT, with ARCHPCT matching the archpct database startup parameter; refer to Chapter 18, “Database Startup and Termination Tools” on page 213).

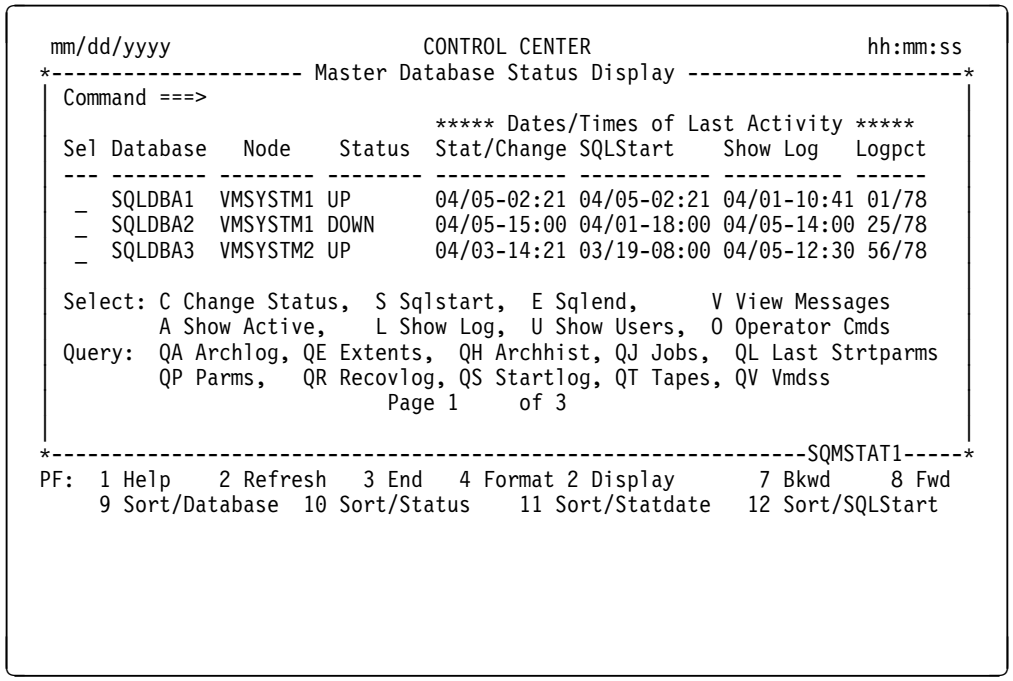


Figure 87. Master Database Status Panel (Format 1)

Format 2 Status Display Panel

PF4 can be used to toggle to and from a second panel, shown in Figure 88 on page 206, which provides additional status information about each database currently displayed including the last message received from the database's console.

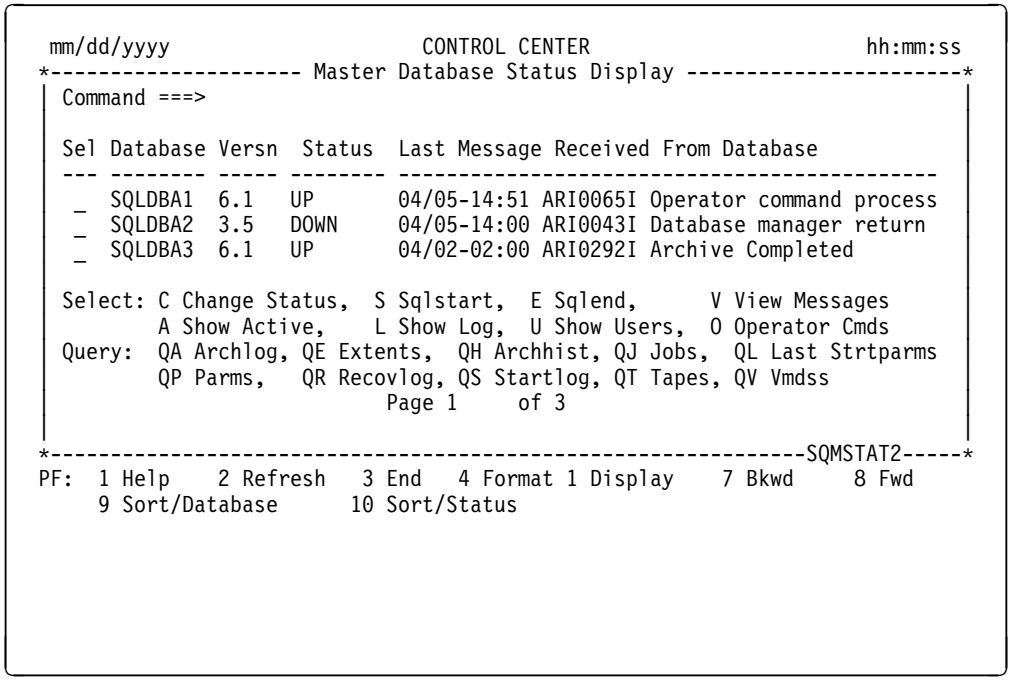


Figure 88. Master Database Status Panel (Format 2)

Select Options

Select Options C, S, E (refer to Figure 88) are available from both the format 1 and format 2 status display panels. These options provide you with quick access to product tools to invoke specific database actions such as database startup or termination.

Option	Description
Change Status (C)	Use this option to change the DBSTATUS and SQMODE entry fields. These status fields are used by the product during automated database operations. If these fields do not correctly reflect the current status of your database, then you may be required to change them. More information about these entry fields and the Change Status (C) option is presented later in this chapter.
Sqlstart (S)	If you determine that the database is down and can be restarted, selection S will issue a start immediate command. Refer to Chapter 18, “Database Startup and Termination Tools” on page 213.
Sqlend (E)	Should you need to terminate a DB2 Server for VM database's operation, selection E will invoke the database termination tool. Refer to Chapter 18, “Database Startup and Termination Tools” on page 213.
View Messages (V)	This option will invoke the View Message Log (refer to “View Message Log Tool” on page 485). This tool provides you with a view of all database

	console messages that have been produced to date on a specified database machine's console.
Show Active (A), Show Log (L), Show Users (U)	Selections A, L, and U will invoke the specific database operator commands indicated (refer to Chapter 16, “Database Operation Interface Tools” on page 197).
Operator Cmds (O)	Selection O will invoke the Database Operation Interface tool (refer to Chapter 16, “Database Operation Interface Tools” on page 197).

Query Options

Query options (QA, QE, ... refer to Figure 88 on page 206) are available from both the format 1 and format 2 status display panels. These options make additional information about your database's status easily accessible from a single panel to aid in the problem determination and resolution process.

Option	Description
Archlog (QA), Archhist (QH)	Displays all database console messages that were produced by a database machine during its last (most current) archive event (log or full). The QH option will display the database's archive history file, which is a chronological listing of all archive activity for a given database (refer to Chapter 19, “Database Archiving and Recovery Tools” on page 233).
Extents (QE)	This option will display current database extent space utilization information in storage pool order (refer to “Query DBEXTENTS/STORPOOL Mapping Tool” on page 493).
Jobs (QJ)	This option will list all jobs associated (active, inactive, or failed) with a specified database (refer to “Job Schedule List Tool” on page 176).
Last Strtparms (QL), Parms (QP)	Option QL will display the database startup parameters that were specified at DB2 Server for VM database startup. Option QP will display the startup parameters as specified in the database PARMS file (refer to “About the Database Parameters Tool” on page 213).
Recovlog (QR)	This option will display the most current database recovery console file. All database console messages produced during the most current database recovery event will be displayed in this file (refer to Chapter 19, “Database Archiving and Recovery Tools” on page 233).
Startlog (QS)	Displays all database console messages that were produced during the last time the database started up (refer to Chapter 18, “Database Startup and Termination Tools” on page 213).

Tapes (QT)	Will invoke the product's database tape management tool for a specified database machine. From this option all tapes (tapes or disks) assigned to a database machine for the purposes of archiving are displayed (refer to Chapter 15, “Tape Management Tool” on page 185).
VMDSS (QV)	Displays the current VMDSS (VM Data Spaces) parameter settings that are used during database startup processing (refer to “About the Database Parameters Tool” on page 213).

PF Key Selections

Various PF keys are available on both the format 1 and format 2 Master Status Display panels which provide you with panel display sorting and refresh capabilities.

PF Key	Description
Refresh (PF2)	This option will refresh the status information currently displayed on format 1 or format 2 Master Status Display panels.
Sort/Database (PF9)	This option will sort the format 1 and format 2 Master Status display panels by database machine name.
Sort/Status (PF10)	This option will sort the format 1 and format 2 Master Status display panels by database machine status.
Sort/Statdate (PF11)	This option will sort the format 1 and format 2 Master Status display panels by date and time of last status change (most current to oldest).
Sort/SQLStart (PF12)	This option will sort the format 1 and format 2 Master Status display panels by date and time of the last database startup time (most current to oldest).

Changing a Database's Status

The product keeps several parameters in memory about the status of a database. This information is used as part of the knowledge base for many of the automated operations that Control Center performs. When a database starts up under product control, the status is always set to NORMAL. NORMAL is an internal status control value and is not displayed to you on the format 1 or format 2 master display panels as NORMAL, but rather as UP. Other status values are set when the product receives information from the applicable database that indicates what condition it is in.

For example, if a TAPE MOUNTED message is received from a database, the product must know what status the database is in so that the proper action can be performed. If the database is archiving, then the TAPE MOUNTED message will cause it to issue a CUU reply to a database archiving message. If the TAPE MOUNTED message is received while the database is in recovery, then a different response would be issued from the product.

Who Can Change a Database's Status

Only those persons with DBA-level authority or Control Center administrators will be able to change the DBSTATUS and SQMODE status entry fields.

When to Change the Database's Status

A database's status may occasionally become invalid (due to system crashes or other failures) and require manual resetting. You may, for example, need to reset the status information before the product will allow you to issue a database restart or any other database command.

The display and changing of the database status information may be done by selecting option C from the format 1 or format 2 master status display panels (refer to Figure 87 on page 205). The panel shown in Figure 89 will display the current DBSTATUS and SQMODE settings for your database. You should exercise caution in changing this information when the database is operational or a database process is active. The product tools depend heavily on these values and results will be unpredictable if they are changed while a process is executing (such as archives or add extents).

```
 mm/dd/yyyy                    CONTROL CENTER                        hh:mm:ss
*------------------------------ Database Status -----------------------------*
| Command ==>                                                CTRLID: MSTRSRV1  |
|                                                             NODE:  VMSYSTM1  |
|    Database ===> SQLDBA         Database Version ===> 6.1                   |
|                                Last Status Change ===> 03/18 14:35          |
|                                                                             |
|    SQMODE   ===> NORMAL                                                     |
|    DBSTATUS ===> NORMAL                                                     |
|                                                                             |
|           Both status fields should be NORMAL if database is up and         |
|         running in normal MULTI-USER mode.                                  |
|                                                                             |
|           To change status, enter new SQMODE and/or DBSTATUS                |
|         and press PF10 to process.  This should ONLY be done for            |
|         correction purposes, CTRLCTR normally maintains these.              |
|                                                                             |
|           Press PF3 to RETURN without changing status.                      |
|                                                                             |
*-----------------------------------------------------------------SQMST10------*
 PF:   1 Help    3 End    10 Change Status (Process)
```

Figure 89. Control Center Change Database Status Panel

About SQMODE and DBSTATUS

SQMODE and DBSTATUS are two status parameters kept by the product for each database. SQMODE is used to keep track of the base process being performed when multiple subprocesses are required to complete the base process. DBSTATUS is used to keep track of the current step or subprocess being performed.

Table 12 on page 210 provides some examples of possible values for the SQMODE and DBSTATUS parameters, along with a description of the corresponding database activity.

Table 12. Example Database Status Values

SQMODE	DBSTATUS	DATABASE MEANING
ADDSPACE	DOWN	SQLADBSP done, start archive
ARI242xx	xxxx	Logmode being changed from L to Y, A, or N
ARI244xx	xxxx	Logmode being changed from Y, A, or N to L
ARI280xx	xxxx	Logmode being changed from A to Y or N
CANCEL	ARCHIVExxxx	Archive CANCEL issued
NORMAL	ADDSPACE RUNNING SQLADBSP	Adding DBSPACEs
NORMAL	ARCHIVEDOWN REQUESTED TAPEWAIT	Waiting for tape mount to do a SQLEND ARCHIVE
NORMAL	ARCHIVEUP REQUESTED TAPEWAIT	Waiting for tape mount to begin a full DB archive
NORMAL	ARCHIVExxxx RUNNING TAPEWAIT	Implicit archive running, tape not mounted yet
NORMAL	ARCHIVExxxx RUNNING TAPEWRITE	Archive running, tape being written to
NORMAL	DOWN	Not operational
NORMAL	LARCHIVEDOWN REQUESTED TAPEWAIT	Waiting for tape mount to begin SQLEND LARCHIVE
NORMAL	LARCHIVEUP REQUESTED TAPEWAIT	Waiting for tape mount to begin a log archive
NORMAL	LARCHIVExxxx RUNNING TAPEWAIT	Implicit larchive running, tape not mounted yet
NORMAL	NORMAL	Up and available for use
NORMAL	STARTUP ARCHIVE LABELDEF	Starting, doing LABELDEFs
NORMAL	STARTUP LOG LABELDEF	Starting, doing LABELDEFs
NORMAL	STARTUP TRACE LABELDEF	Starting, doing LABELDEFs

Resetting DBSTATUS and SQMODE

In the event that the DBSTATUS and SQMODE entry fields need to be changed to match the true status of your database machine, then you will need to invoke the Change Status (**C**) option from either the format 1 or format 2 Master Status Display panels. Refer to Figure 87 on page 205. Figure 89 on page 209 will be displayed with the current Control Center DBSTATUS and SQMODE settings.

Preparing for Database Restart: Change the DBSTATUS setting to DOWN and the SQMODE setting to NORMAL, then press PF10 to process the change.

Telling Control Center the Database is Up and Running: Change the DBSTATUS setting to NORMAL and the SQMODE setting to NORMAL, then press PF10 to process the change.

Database Startup Processing and Control Center Status: The DBSTATUS and SQMODE settings will be automatically re-synchronized during database startup

processing. Therefore, another method for correcting incorrect DBSTATUS and SQMODE settings for a database that is down is to recycle (IPL, autolog, log-off/log-on) the database machine. During the startup process the SQMODE and DBSTATUS entry fields will be correctly reset.

Chapter 18. Database Startup and Termination Tools

Overview

The Database Startup and Termination tools are used to view and update the database parameters file and VM storage pool specifications file for a given database; view the database startup parameters used during the last startup of a database; and to start and stop database operation.

Before continuing, you should review these topics:

- The database parameters file. See “Database PARMS File” on page 128.
- The authorization scheme. See “Authorization” on page 26.
- The database machine startup parameters. Refer to manual *DB2 Server for VM System Administration*.
- The Job Scheduling tool, if you plan on scheduling the startup or termination of a database. Refer to Chapter 14, “Job Scheduling Tool” on page 165.

About the Database Parameters Tool

Shown in Figure 90 is the option selection panel of the Database Parameters tool. It is reached by selecting **P - Database Parameters** on the Main Menu. The communication path can be changed using Option **C** from the Control Center Main Menu, whereas the database setting can be changed on this panel. For more information, see “Control Center Communication Path and Database Settings” on page 143.

```
 mm/dd/yyyy                    CONTROL CENTER                       hh:mm:ss
*--------------------------- Database Parameters ---------------------------*
| Option ===>                                          CTRLID: MSTRSRV1      |
|    Database ===> SQLDBA                               NODE:  VMSYSTM1      |
|                                                                            |
|      V  VIEW                   View database parameters                    |
|      U  UPDATE                 Update database parameters                  |
|      S  STARTUP                Display startup parameters used             |
|                                during last database startup                |
|     VS  VIEW SPECS             View Storage Pool Specifications File       |
|                                 (VMDSS Support)                            |
|     US  UPDATE SPECS           Update Storage Pool Specifications File     |
|                                 (VMDSS Support)                            |
|                                                                            |
|            Enter OPTION, select DATABASE, press ENTER to process           |
|                                                                            |
*-----------------------------------------------------------------SQMDPARM----*
 PF:   1 Help    3 End
```

Figure 90. Database Parameters Selection Panel

Display Fields

Database Parameters File

Control Center maintains a set of database startup and control parameters for each database machine it manages. Parameters for each database are saved in separate files named *database* PARMS, where *database* is the database machine ID, and are kept on the service machine's 191 A-disk. Database parameters are separated into the groups listed below:

1. Authorization/Notification parameters
2. Utility parameters
3. Tape parameters
4. Archiving/Tracing parameters
5. Database startup parameters
6. Data Restore parameters

Initially, parameter values are established during the DB2 Server for VM database installation process. Afterwards, they can be updated using the UPDATE **U** option. Several parameters are updated automatically by Control Center during normal operation and under most circumstances should not be updated manually.

Parameters can be viewed and updated at any time; however, most of the parameters updated will not take effect until the next database startup. The parameters used during the last database startup are available for viewing at any time using the STARTUP **S** option.

About the Storage Pool Specification File

Control Center maintains a set of storage pool specifications for each database machine it manages which allows you to control the DB2 VM Data Spaces feature at the individual storage pool level. Specifications for each database are saved in separate files named "*database* ARISPOOL", where *database* is the database machine ID, and are kept on the service machine's 191 A-disk.

Three VMDSS specifications can be specified for storage pools:

- Whether Data Spaces Support or the standard database DASD I/O system is to be used (Data Space Support is the default)
- The working storage residency priority, for those pools that use Data Spaces (a residency priority of 3 is the default)
- Whether or not striping is used (striping is the default)

Initially, when Control Center creates the storage pool specification file, it uses the information provided by the SHOW POOL operator command to determine the proper settings. Afterwards, they can be updated using the UPDATE SPECS **US** option.

These settings can also be changed dynamically while the database is running by issuing specific database operator commands. Dynamic changes, however, do not change the settings in the database's storage pool specifications file, the file used at database startup time.

For more information on the VMDSS, refer to the *DB2 Data Spaces Support* manual.

VIEW (V)

Select this option to view the current parameter settings saved in the database parameters file.

The display panels used by the VIEW **V** option are similar to the entry panels used by the UPDATE **U** option. See "UPDATE (U)."

Usage Consideration: The displayed parameter settings are those that will be used during the next database machine startup; use the STARTUP **S** option to view the parameters used during the last database startup.

UPDATE (U)

Select this option to update one or more parameters in the database parameters file. Updated parameters will take effect the next time the database machine is started.

For each of the general parameter groups, a separate data entry panel is presented. Scrolling from one panel to the next is done using PF8 to go forward and PF7 to go backward. You can change existing parameters through normal editing means. Update of the parameters file on the Control Center service machine's 191 A-disk is completed by selecting PF2 to process your changes. You will be prompted to confirm the update action.

Authorization/Notification Parameters: The initial data entry panel displayed is that of the Authorization/Notification parameters shown in Figure 91.

```
 mm/dd/yyyy                    CONTROL CENTER                      hh:mm:ss
*------------------------ Database Parameter Display ------------------------*
| Command ==>                                              CTRLID: SQMSTPD1  |
|  Database => SQLDBA     VERSION  => 6.1.0             NODE: VMSYSTM1       |
|  ************************ Authorization Parameters ********************** |
|  Administrators  => D128980 DUKEWE M760595 SQSUPPD1 DUKEWE AT WMAVM7       |
|    S818738                                                                 |
|  Operators       => SQMDEMO3                                               |
|                                                                            |
|  Users           => MILBURNJ                                               |
|                                                                            |
|  Notify-error    => DUKEWE D128980                                         |
|                                                                            |
|  Notify-severe   => DUKEWE D128980                                         |
|                                                                            |
|  Notify-dbstatus => DUKEWE D128980                                         |
|                                                                            |
|  Notify-operator =>                                                        |
|                                                                            |
|                               Page  1 of  9                                |
|                                                                            |
*----------------------------------------------------------------SQMDP10----*
 PF: 1 Help   3 End    7 Bkwd (Data Restore parms)    8 Fwd (Utility parms)
```

Figure 91. Database Parameter Display

The data entry field for each Authorization/Notification parameter spans two lines, allowing a list of user IDs to be entered. Each user ID must be separated by a

space and a single token should not be split over the two lines (the end-of-line character is translated into a blank by the Control Center).

The syntax for specifying authorization/notification parameters is given below.

Notes:

1. A local user. The nodeid is assumed to be same as that of the Control Center service machine
2. A remote user. The nodeid is different than that of the Control Center service machine.
3. A nickname within the CMS NAMES file on the Control Center service machine's 191 A-disk. Refer to the NAMES command documentation provided with VM/CMS. The nickname format allows multiple users to be associated with a single nickname, which eliminates the limited space restrictions of the parameter entry panel.

Parameter	Description
Version	Current version, release and modification level of DB2 Server for VM (6.1.0). This parameter will be updated as necessary when the database starts.
Administrators	Users who are allowed to execute any Control Center commands against the specified database. These users will be able to initiate archives and recoveries and to start and stop the database. Two authorized administrators are shown in Figure 91 on page 215. MSTRUSR1 is a local user and MSTRUSR2 is a remote user.
Operators	Users who are allowed to execute a subset of the Control Center DBA commands to the specified database. Commands which they will be authorized to execute include initiating an archive and starting and stopping the database. They will not be authorized to execute DBA commands such as adding a dbextent.
Users	Users who are allowed to issue the SHOW, COUNTER, and RESET commands to the specified database. Referring to Figure 91 on page 215, MSTRUSR1 is a local user and USRNICKS is a CMS NICKNAME. USRNICKS must be included in the Control Center NAMES file on the Control Center service machine's 191 A-disk. This is the preferred method of specifying many users when using the entry panel, which has a limited amount of space for each parameter value.

Notify-error	Users who are to be notified when an error is received from the specified database.
Notify-severe	Users who are to be notified when a severe error is received from the specified database.
Notify-dbstatus	Users who are to be notified whenever the status of the specified database changes (UP, DOWN, ARCHIVE). This parameter will also notify designated users of monitor error conditions. See "Who is Notified" on page 347.
Notify-operator	VM operator who should be notified when serious problems occur within the specified database.

Utility Parameters

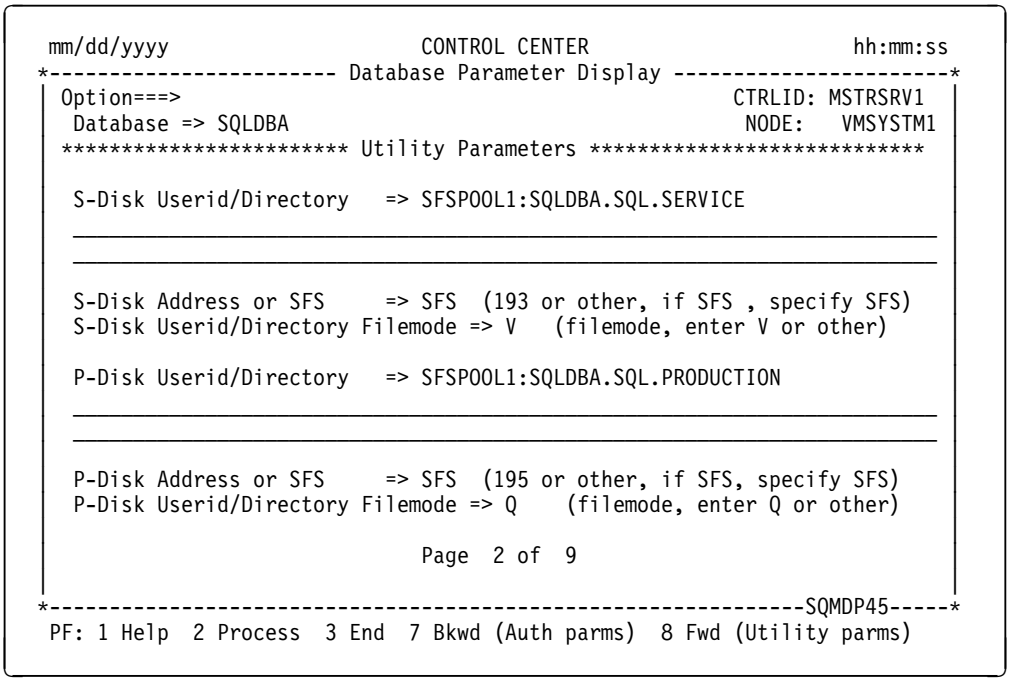


Figure 92. Data Entry Panel for Utility Parameters

Parameter	Description
S-Disk Userid/Directory	The virtual machine name (user ID) that owns the service disk for this database. If the database is installed with SFS, specify the complete directory name.
S-Disk Address or SFS	The virtual address of the service disk. For SFS, type "SFS" instead of the virtual address.
S-Disk/Directory Filemode	The filemode used to access the database service disk or directory containing the service code. The default value is V.

P-Disk Userid/Directory	The virtual machine name (user ID) that owns the production disk for this database. If the database is installed with SFS, specify the complete directory name.
P-Disk Address or SFS	The virtual address of the production disk. For SFS, type "SFS" instead of the virtual address.
P-Disk/Directory Filemode	The filemode used to access the database production disk or directory containing the production code. The default value is Q.

```
 mm/dd/yyyy                    Control Center                      hh:mm:ss
*----------------------- Database Parameter Display -----------------------*
| Option ===>                                              CTRLID: CCDEMO   |
|  Database => ACTCLONE                                    NODE:   YOURNODE |
|  ********************** Utility Parameters ***************************** |
|                                                                           |
|  Archive-addspace          => N        ( Y or N, Archive after SQLADBSP)  |
|  Archive-addextent         => N        ( Y or N, Archive after SQLADBEX)  |
|                                                                           |
|  Sqlend-quick              => N        ( Y or N, SQLEND QUICK processes)  |
|                                                                           |
|  Uarchive-enabled          => N        ( Y or N, User Archiving enable)   |
|  Drmstr-enabled            => N        ( Y or N, Data Restore enable)     |
|                                                                           |
|  Submit-routine            => ________ ( UARCHIVE submit routine)         |
|  Cancel-routine            => ________ ( UARCHIVE cancel routine)         |
|  Spool-console-option      => N        ( Y or N, Spool Database Console)  |
|                                                                           |
|                                                                           |
|                              Page  3 of  9                                |
|                                                                           |
*-------------------------------------------------------------------SQMDP46-----*
 PF: 1 Help    2 Process  3 End  7 Bkwd (Utility Parms 1)  8 Fwd (Tape Parms)
```

Figure 93. Data Entry Panel for Utility Parameters - Part 2

Parameter	Description
Archive-addspace	A "Y" or "N" value (YES or NO) to indicate whether an archive should automatically be invoked whenever the Add DBSPACE tool is executed. The Add DBSPACE tool is executed in single user mode on the database machine and is not recorded in the database log. If a recovery is done to a previous archive, the added dbspaces will not be recovered by the database.
Archive-addextent	A "Y" or "N" value (YES or NO) to indicate whether an archive should automatically be invoked whenever the Add DBEXTENTS or Delete DBEXTENTS tool is executed. These tools execute in single user mode and are not recorded in the database log.
Sqlend-quick	A "Y" or "N" value (YES or NO) to indicate whether the SQLEND QUICK command should be used to terminate the database for all single user mode

activities. If "N" is specified, then the SQLEND command without the QUICK option will be used, which will allow all active LUWs within the database to complete before the database is brought down. The QUICK option will stop the database immediately and roll back all active LUWs.

Uarchive-enabled — (Y,N). "Y" (YES) instruct Control Center that USER ARCHIVES are allowed for this database. "N" (NO) will not allow a USER ARCHIVE to proceed. This value and the Drmstr_enabled value must be Y in order to execute an USER ARCHIVE using Data Restore BACKUP. If Uarchive-enabled = "Y" and Drmstr-enabled = "N", then the user is responsible for creating the required submit-routine and Cancel-routine files and setting up the User archive process as described in Appendix F, “User Archiving” on page 515.

Submit-routine — The name of an executable command (EXEC or MODULE) that should be executed after the database comes down following the SQLEND UARCHIVE command. Do not fill in this value if Drmstr_enabled = "Y".

Cancel-routine — The name of an executable command (EXEC or MODULE) that should be executed if the CANCEL ARCHIVE command is issued by a user. Do not fill in this value if Drmstr_enabled = "Y".

Spool-console-option — A "Y" or "N" value (YES or NO) to indicate whether the database machine console is to be spooled during operation. The default is no spooling.

Tape Parameters

```
 mm/dd/yyyy                    Control Center                    hh:mm:ss
*----------------------- Database Parameter Update ------------------------*
| Command ==>                                              CTRLID: MSTRSV1  |
|  Database => SQLDBA                                      NODE:   VMSYSTM1 |
|  ********************** Tape Parameters ***************************       |
|                                                                           |
|  Tape-density        => XF     ( e.g. 4K, 38K )                          |
|                                                                           |
|   NOTE:   The following parameters apply to DYNAM/T, EPIC and VMTAPE       |
|                                                                           |
|  Tape-retention      => ____   ( Tape retention <days> )                  |
|  Scratch-tape-option => N ( Always use SCRATCH Y/N, if DYNOPEN, enter Y )  |
|          ( archive, log, & trace dataset names for tape label checking    |
|  Tape-archive-dsn    => ____________________________________________      |
|  Tape-log-dsn        => ____________________________________________      |
|  Tape-trace-dsn      => ____________________________________________      |
|          ( scratch pool name for database )                               |
|  Scratch-pool        => ____________________________________________      |
|  Logtape-premount    => N     ( N with DISK or DMSTVI,  Y with DYNAMT )    |
|                                                                           |
|                               Page  4 of  9                               |
|                                                                           |
*-------------------------------------------------------------SQMDP50-----*
 PF: 1 Help 2 Process 3 End   7 Bkwd (Utility Parms 2)  8 Fwd (Archive Parms)
```

Figure 94. Data Entry Panel for Tape Parameters

Parameter	Description
Tape-density	The tape density to specify with the TAPE MOUNT command during archiving and recovery activity (any valid VM filedef tape density, (XF)).

These parameters are applicable only when using DYNAM/T, EPIC, or VMTAPE.

Parameter	Description
Tape-retention	Tape retention period in days.
Scratch-tape-option	(Y,N). Indicates whether you want to *always* use SCRATCH tapes for log archives, database archives or the primary BACKUP rather than predefined volids. If you use DYNOPEN, you must specify 'Y' to always use SCRATCH tapes. **Note:** The "Backup2-scratch-tape" parameter is used to indicate whether scratch is to be used for the secondary BACKUP (BACKUP2).
Tape-archive-dsn	Dataset name that should be specified for new scratch tapes. Use the naming convention for the DSN identified to the tape management system catalog.
Tape-log-dsn	Dataset name that should be specified for new log scratch tapes. Use the naming convention for the DSN identified to the tape management system catalog.
Tape-trace-dsn	Dataset name that should be specified for new trace scratch tapes. Use the naming convention for the DSN identified to the tape management system catalog.
Scratch-pool	Specifies a special SCRATCH TAPE POOL to be used by this database for all archives, and Data Restore backups, and unload output. This value is optional.
Logtape-premount	Set to "Y" if Control Center is to issue tape mount requests for log archive tapes. By allowing Control Center to issue the tape mount, the log archive tape can be mounted in preparation for a log archive (premounted). During the period of time between when the log mount was issued and when the physical tape drive is attached, normal database activities can continue, including the start of new LUWs. **Operational Note:** If log archive media is "DISK" you must specify "N". If you are using the exit DMSTVI, specify "N". If you are using DYNAM/T, you must specify "Y".

Archiving/Tracing Parameters

```
 mm/dd/yyyy                    CONTROL CENTER                      hh:mm:ss
*------------------------ Database Parameter Update ------------------------*
| Command ==>                                                 CTRLID: MSTRSRV1 |
|  Database => SQLDBA                                         NODE:   VMSYSTM1 |
|  ******************** Archiving/Tracing Parameters ********************     |
|                                                                              |
|   Archive-media     => TAPE        ( TAPE or DISK )                          |
|   Archive-blksize   => 28672       ( e.g. 28672 )                            |
|   Archive-series    => 100         ( e.g. 200 )                              |
|                                                                              |
|   Logarch-media     => TAPE        ( TAPE or DISK )                          |
|   Logarch-blksize   => 28672       ( e.g. 28672 )                            |
|   Logarch-series    => 100         ( e.g. 100 )                              |
|                                                                              |
|   Trace-media       => TAPE        ( TAPE or DISK )                          |
|   Trace-blksize     => 28672       ( e.g. 28672 )                            |
|   Trace-series      => 100         ( e.g. 200 )                              |
|                                                                              |
|                                                                              |
|                                                                              |
|                                Page  5 of  9                                 |
|                                                                              |
*------------------------------------------------------------------SQMDP55-----*
 PF: 1 Help  2 Process  3 End    7 Bkwd (Tape parms)   8 Fwd (Startup parms 1)
```

Figure 95. Data Entry Panel for Archiving/Tracing Parameters

Parameter	Description
Archive-media	(TAPE, DISK). Indicates the type of media to be used for DB2 Server for VM database archive or the Data Restore primary BACKUP or Incremental Backup. The media type must match the type of ARCHIVE media defined in the database TAPES file. **Usage Consideration:** Database archives to DISK are not fully supported by DB2 Server and VM and must be used with caution.
Archive-blksize	The block size that should be used for full database archives.
Archive-series	The tape series that will be used for the *next* database archive or BACKUP. This value must match one of the series listed in the database TAPES file. When the database starts up, or before a Data Restore backup executes, the archive-series is used to determine what tapes or files will be FILEDEFed for the archive or BACKUP FILEDEF. **Operational Note:** Control Center automatically changes the archive-series after an archive or BACKUP in order to rotate through the available tape series.
Logarch-media	Enter "TAPE" or "DISK" to indicate whether log archives should be done to tape or disk. If log archiving is not to be performed, this parameter should be left blank.

Logarch-blksize	This parameter indicates the block size that should be used for log archives.
Logarch-series	The tape series that is to be used for the *next* log archives. **Usage Consideration:** This value must match the previous value of the Archive-series parameter when a successful full ARCHIVE was performed. Therefore, it must be different from the current Archive-series. For initial setup, the value given here is not significant until the first full archive is done. **Operational Note:** Control Center automatically changes this value after an SQLEND in order to rotate through the available tape series.
Trace-media	Enter "TAPE" or "DISK" to indicate whether trace data should be written to tape or disk. This entry field should be left blank if tracing is not done.
Trace-blksize	Indicates the block size that should be used for trace output.
Trace-series	Indicates the tape series that will be used for trace output.

Database Startup Parameters: Field-specific help is available on this panel by placing the cursor at the first position of an entry field and selecting PF5. Help is displayed near the bottom of the panel.

Refer to the *DB2 Server for VM System Administration* or *DB2 Server for VSE & VM Operation* manuals for a complete description of each DB2 Server for VM database startup parameter.

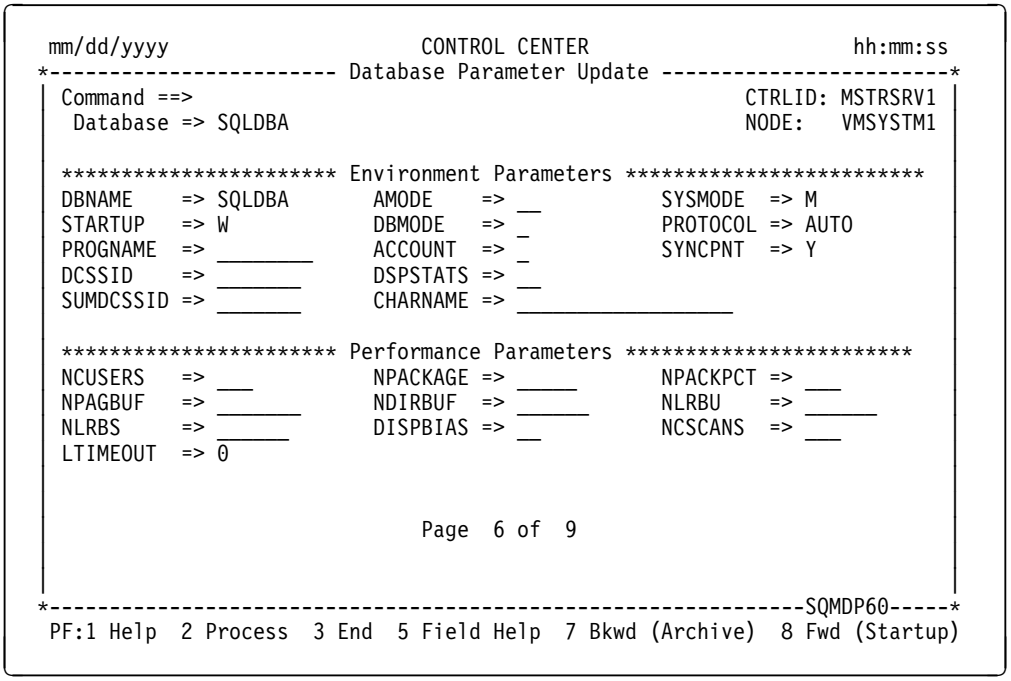


Figure 96. Database Parameter Update Panel - Part 1

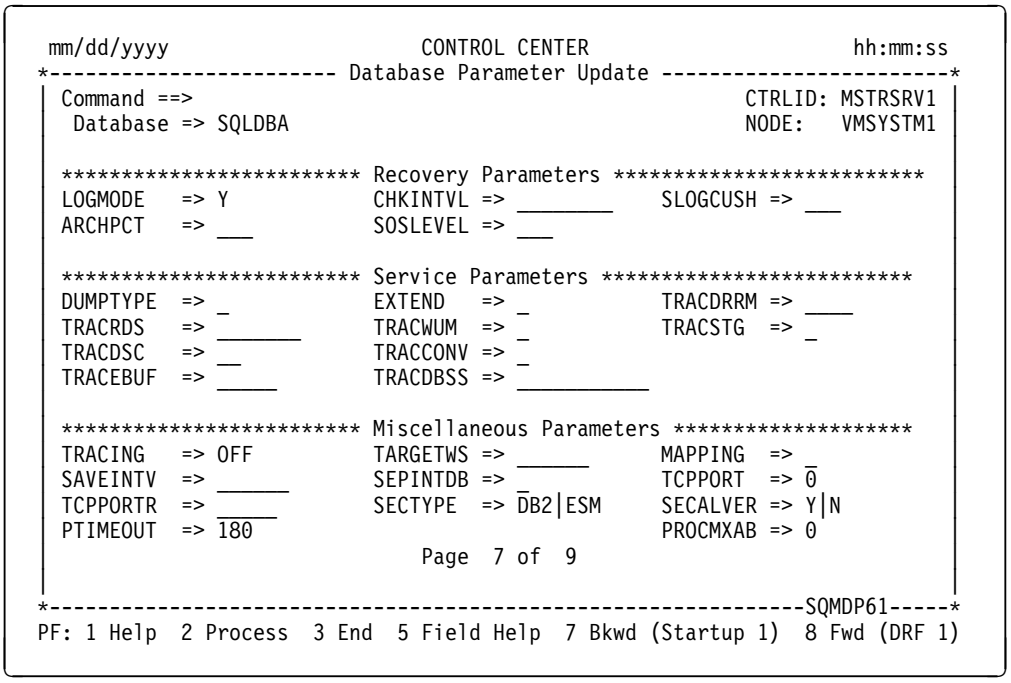


Figure 97. Database Parameter Update Panel - Part 2

The VMDSS parameters are explained in the *DB2 Data Spaces Support* manual.

The output media for trace information is defined by the database Archiving/Tracing parameter settings. See “Archiving/Tracing Parameters” on page 220.

Data Restore Parameters in the Database PARMS File

```
 mm/dd/yyyy                    CONTROL CENTER                      hh:mm:ss
*----------------------- Database Parameter Update -----------------------*
| Command ==>                                                CTRLID: MSTRSRV1 |
|  Database => SQLDBA                                        NODE:   VMSYSTM1 |
| ************************ Data Restore Parameters ********************** |
|                                                                           |
|  Drmstr-enabled        => N        ( Y or N, Default = N)                 |
|  Data-restore-machine  => ________ ( Machine running Data Restore Userid) |
|  Drtape-premount       => N        ( Y or N, Default = N )                |
|  Data-restore-lang     => S001     ( Default = S001 )                     |
|  Dual-backup           => N        ( Y or N, Default = N )                |
|  Backup-wrksize        => 2048     ( Default = 2048 )                     |
|  Backup2-media         => ____     ( TAPE or DISK )                       |
|  Backup2-scratch-tape  => N        ( Y or N, Default = N )                |
|                                                                           |
|  Incbackup-enabled     => N        ( Y or N, Default = N )                |
|  Incbackup-series      => 100.01   ( nnn.nn format )                      |
|  Auto-full             => N        ( Y or N, Default = N )                |
|  Current-increference  =>                    ( Date & Time Information )  |
|                                                                           |
|                              Page  8 of  9                                |
|                                                                           |
*-------------------------------------------------------------SQMDP66-----*
 PF: 1 Help 2 Process 3 End   7 Bkwd (Startup parms 2)  8 Fwd (DRF 2 parms)
```

Figure 98. Data Restore Parameters

Parameter	Definition
Drmstr-enabled	(Y,N). Y (YES) indicates that Data Restore has been enabled for this database and you wish to perform Data Restore functions with this database. N (NO) indicates that you do not wish to execute Data Restore functions with this database. This value must be Y in order for BACKUP to proceed. If you wish to perform another type of USER ARCHIVE even though you have Data Restore installed, set this value to N. Setting this value to "Y" does not require that you perform Data Restore functions. If the Uarchive-enabled value is "N" (NO) then Data Restore BACKUP will not be executed.
Data-restore-machine	Is the machine id (USER ID) of the machine that will run the Data Restore functions for this database. This Data Restore machine will be managed by one Control Center service machine. A Data Restore machine can control more than one database but both database and Data Restore machines must be controlled by the same Control Center service machine.
Drtape-premount	(Y,N). This value indicates whether Control Center MUST premount the first tape before a Data Restore function is executed. If you have a tape management system that uses the DMSTVI tape exit, then mounting of the first tape of a multi-volume series can be handled by the DMSTVI tape exit automatically. The value should be set to "N". If you do not have a tape management system

	(native CMS), then tape mount requests will be made automatically for you based on your values in the SQLMSTR Control file. If your tape management system does not include the DMSTVI MODULE, or if the automatic tape mounts are not working on your system, then set the value to "Y". Regardless of the tape exit available, CMS handles all subsequent multi-volume tape mounts based on information in the FILEDEF and LABELDEF information for the tapes.
Data-restore-lang	(S001, S002, S003, S004, S005, S006). Indicates what language will be used by Data Restore to issue messages. The value will be assigned to the "LANG=" option in the SYSIN file. The default is "S001" (American English, AMENG).
Dual-backup	(Y,N). N (NO) indicates that only a single BACKUP Incremental Backup is to be performed. For a BACKUP, the ARCHIVE tapes defined in the database TAPES file for the current "Archive-series" will be used for the primary BACKUP. For Incremental Backup, the "INCBK" tapes defined in the database TAPES file for the current "Incbackup-series" will be used for the primary Incremental Backup. If Y (YES) is chosen, then a DUAL BACKUP or DUAL Incremental Backup will be performed. The BACKUP2 tapes for the current "Archive-series" will be used for the secondary BACKUP and the "INCBK2" tapes for the current "Incbackup-series" will be used for the secondary Incremental Backup. The default value is N.
Incbackup-enabled	(Incremental Backup enabled Y,N). Y (Yes) indicates Control Center will manage both Full Backup and Incremental Backup. Indicating Y does not require that Incremental Backup be used; however, Control Center will only execute a BACKUP with the FULL or INCREMENTAL option. This will allow Control Center to keep a Full Backup and its Incremental Backups together in the same tape series and prevent a Full Backup from being overwritten prematurely. Full Backup is the default backup type. When "Incbackup-enabled" = "N", Control Center will not execute an Incremental Backup and the "Current-increreference" parameter will be set to blank when a normal BACKUP or archive completes. When "Current-increreference" is blank, a Full Backup or an ARCHIVE FULL must be executed before Control Center will allow an Incremental Backup.

Incbackup-series	(Incremental Backup series.) Indicates the series and subseries in the database TAPES file that will be used to supply tapes for the Incremental Backup process. The Incbackup-series is presented in the form "**series.subseries"**". The subseries indicates which set of Incremental Backup tapes to use within the series. For example, Incremental Backup series 100.01 indicates that the first set of Incremental Backup tapes (01INCBK) in series 100 are to be used. **Only** the subseries is incremented after an Incremental Backup successfully completes. The series is incremented and the subseries reset to "01" each time any normal BACKUP, Full Backup, normal archive or ARCHIVE FULL completes. The series portion of the Incbackup-series will be the same as the Logarch-series. The Incbackup-series will exist in the PARMS file even when "Incbackup-enabled" = "N".
Current-increference	(Current incremental reference). Indicates the timestamp of the last successful Full Backup or ARCHIVE FULL. The Current-increference timestamp will be included in the database ARCHHIST entry for subsequent Incremental Backup. This parameter should not be modified by the user. If "Incbackup-enabled" = "N" and a normal BACKUP or archive has completed, then this parameter will be blank. When this parameter is blank, a Full Backup or ARCHIVE FULL must be executed before Control Center will allow an Incremental Backup.
Auto-full	(Y,N). "Y" (Yes) indicates that an Incremental Backup will be changed to a Full Backup if there are no more incbackup-series and SCRATCH tapes are not used. If Incremental Backups are performed to tape and you are using SCRATCH tapes, then a new SCRATCH tape entry will be added to the database TAPES file for the current Incbackup-series. When Auto-full = "N" (No), then the Incremental Backup will be cancelled if the current Incbackup-series does not exist in the database TAPES file. If the "AUTOFULL" option is used with the BACKUP command then it will over-ride the "Auto-full" value in the database PARMS file.
Backup2-scratch-tape	(Y,N) Indicates if SCRATCH tapes will always be used for the secondary backup or Incremental Backup of a dual backup.

Backup2-media	(TAPE, DISK). Indicates the expected type of media to be used by Data Restore for the secondary BACKUP or Incremental Backup ("BACKUP2"/"INCBK2"). The value must match the type of media defined in the database DRTAPES file or an error will occur during the BACKUP. This value is only used if "Dual_backup" is "Y".
Backup_wrksize	Indicates the "WRKSIZE=" value that will be used for BACKUPs and Incremental Backups. The default is 2048.
Tape_backup_dsn	Indicates what Data Set Name to include in the LABELDEF command (ARCHIV) for the primary BACKUP or Incremental Backup.
Tape_Backup2_dsn	Indicates what Data Set Name to include in the LABELDEF command (ARCHIV2) for the secondary BACKUP or Incremental Backup.

```
 mm/dd/yyyy                    Control Center                     hh:mm:ss
*----------------------- Database Parameter Update ------------------------*
| Command ==>                                              CTRLID: MSTRSRV1 |
|  Database => SQLDBA                                      NODE:   VMSYSTM1 |
|  ************************ Data Restore Parameters **********************  |
|                                                                           |
|   Translate-media        => TAPE     ( TAPE or DISK )                     |
|   Translate-scratch-tape => Y        ( Y or N, Default = N )              |
|                                                                           |
|        ( Data Set Names for Translate, Unload, and Backup tapes )         |
|   Tape-translate-dsn     => DB261DBA.TRANSLATE                            |
|   Tape-unload-dsn        => DB261DBA.UNLOAD                               |
|                                                                           |
|   Tape-backup-dsn        => DB261DBA.BACKUP                               |
|   Tape-backup2-dsn       => DB261DBA.BACKUP2                              |
|                                                                           |
|                                Page  9 of  9                              |
| The requested command was issued                                          |
*-------------------------------------------------------------SQMDP67-----*
 PF: 1 Help  2 Process  3 End   7 Bkwd (DRF Parms 1)   8 Fwd (Auth Parms)
```

Figure 99. Data Restore Parameters

Parameter	Definition
Translate-media:	(TAPE or DISK) Indicates the expected type of media to be used by Data Restore for the TRANSLATE function. The value must match the type of media defined in the database TAPES file or an error will occur during the TRANSLATE. TRANSLATE-MEDIA must be set to TAPE, when TRANSLATE-SCRATCH-TAPE = Y. This field can be left blank when "DRMSTR-ENABLED" = "N".

Translate-scratch-tape: (Y or N, default = N) Y (Yes) indicates that scratch tapes should be used for TRANSLATE. N (NO) indicates that scratch tapes should NOT be used for TRANSLATE.

Tape-translate-dsn: Indicates what data set name to include in the LABELDEF command for the TRANSLATE tapes. This field is optional.

Tape-unload-dsn: Indicates what data set name to include in the LABELDEF command for the UNLOAD tapes. This field is optional.

STARTUP (S)

Displays the startup parameters that were used by the database when it was last started (which can differ from those in the current database PARMS file if it has been updated since this time). Included will be any default values supplied by the database.[10]

The date and time that the database was started will also be shown.

VIEW SPECS (VS)

Select this option to view the current settings saved in the storage pool specifications file.

The display panel used by the VIEW SPECS **VS** option is similar to the entry panel used by the UPDATE SPECS **US** option. See “UPDATE SPECS (US).”

Usage Considerations:

1. The displayed parameter settings are those that will be used during the next database machine startup. Changes made dynamically while the database is running are *not* reflected in the specifications file.
2. You can go directly from the VIEW SPECS panel to the UPDATE SPECS panel by selecting PF5.

UPDATE SPECS (US)

Select this option to update one or more parameters in the storage pool specifications file. Updated parameters will take effect the next time the database machine is started.

Update of the specifications file on the service machine's 191 A-disk is completed by selecting PF2 to process your changes. You will be prompted to confirm the update action.

The storage pool specifications data entry panel is shown in Figure 100 on page 229.

[10] Default values supplied by the database are not saved in the database parameters file. Therefore, they cannot be viewed using the VIEW (V) option.

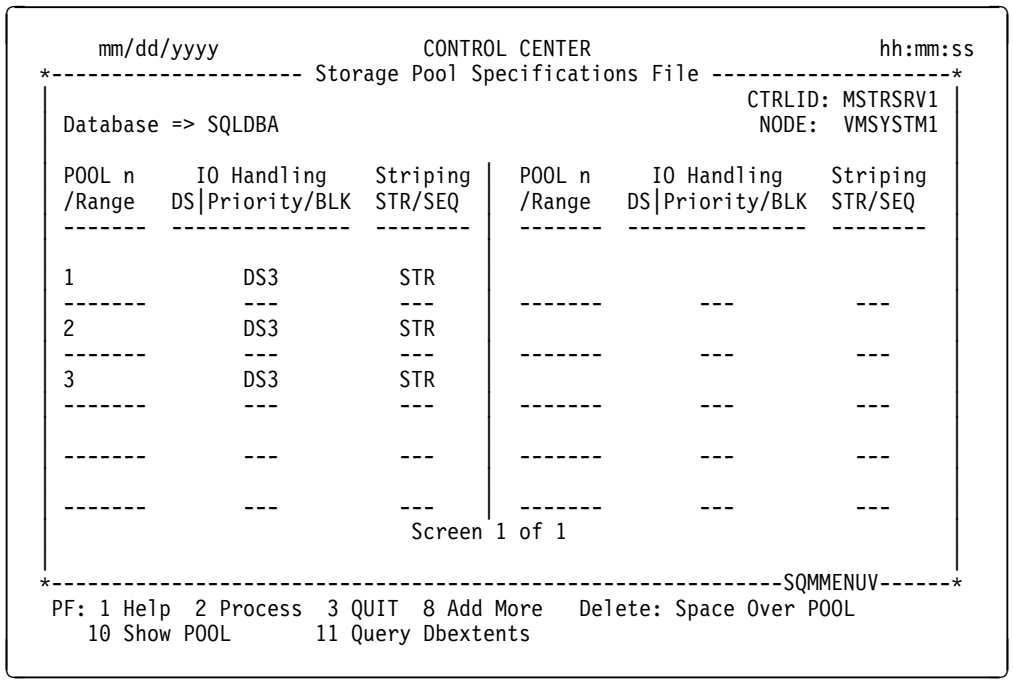


Figure 100. Data Entry Panel for Storage Pool Specifications

Parameter	Description
POOL n/Range	A valid pool number (n; where n=1 to 999), or pool range (nnn-mmm; where nnn, mmm=1 to 999). Use only positive values for pool numbers. To delete a pool or pool range, space over the pool number or pool range entry.
IO Handling DS|Priority/BLK	Specifies whether VM Data Spaces Support is ON or OFF for the specified storage pool or pool range (ON is the default). Possible values are: **DS** Data spaces support will be used for the specified storage pool range. This is the default. **Priority** Used to set the storage residence priority for the specified storage pool or pool range. Valid Priority range = 1 to 5 (3 is the default). If not indicated, the default will be used. The priority value should be entered immediately after the **DS** term as shown in Figure 100. **BLK** Standard DB2 DASD IO will be used. In other words, data spaces support is turned OFF.

Striping STR/SEQ

Specifies whether striping is ON or OFF for the specified storage pool or pool range (ON is the default).

Possible values are:

STR Indicates that data striping has been turned ON for the specified storage pool or pool range. Data will be striped, or evenly distributed across all dbextents within a given storage pool. This is the default.

SEQ Indicates that striping has been turned off for the storage pool or pool range. Data will be allocated to one dbextent at a time until that dbextent is filled, then the next dbextent in sequence will be used.

About the Database Startup Tool

Use the Database Startup tool to initiate the immediate startup of a database or to schedule the startup of a database for a later date and time using the Job Scheduling tool.

Database Startup - Immediate (SI)

Select this option to immediately issue a database startup command to the specified database. Startup parameters will be taken from the database parameter file and storage pool specifications file (if it exists), which reside on the service machine's 191 A-disk. The startup command will be rejected by Control Center if the current status of the database is not "Down."

Database Startup - Scheduled (SS)

Select this option to schedule the database to start up at a later date and time.

Database Termination Tool

Use the Database Termination tool to stop a database. The Database Termination selection panel, shown in Figure 101 on page 231, is presented after choosing Option **E** (SQLEND Database) from the Control Center Main Menu.

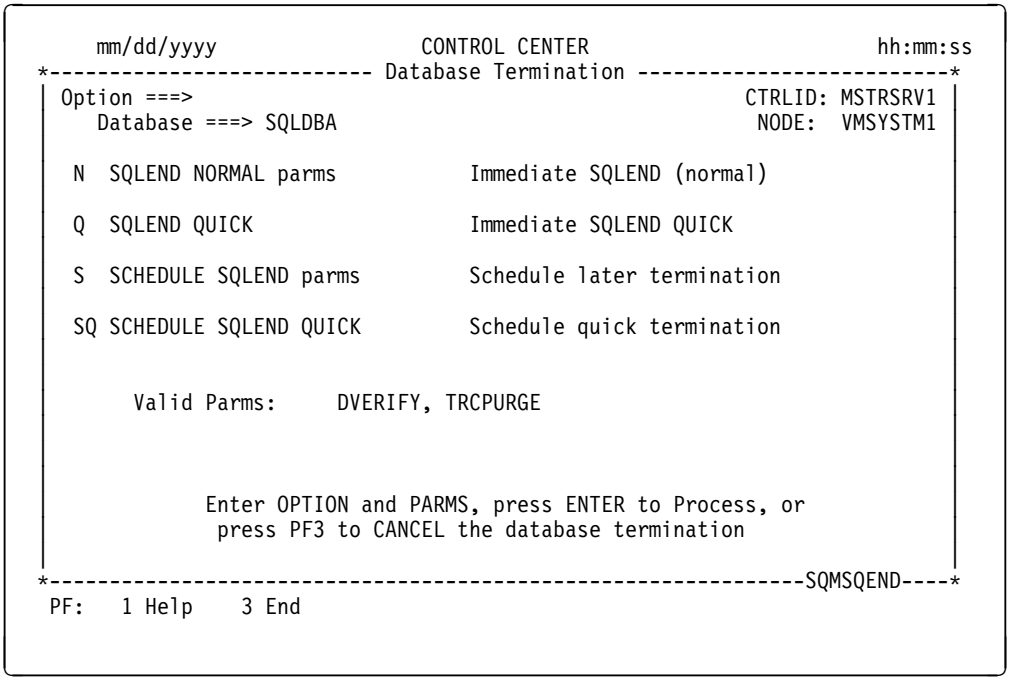


Figure 101. Database Termination Option Panel

SQLEND NORMAL (N)

Selecting this option will allow all current LUWs to complete prior to stopping the database and prevent any new units of work from beginning. Active LUWs can delay the stopping of the database.

Optional parameters:

DVERIFY Causes the database system to verify its directory before ending. If a directory error is found, the authorized Control Center administrators for the database will receive a message.

TRCPURGE If specified, database will purge the contents of the trace buffer at shutdown. An output trace file will not be created for the trace buffer in this case.

SQLEND QUICK (Q)

Selecting this option will immediately stop the database. Any active LUWs will be rolled back when the database is brought back up.

SCHEDULE SQLEND (S)

Select this option to schedule the database to stop at a later date and time. At the scheduled time, all current LUWs will be allowed to complete prior to the database being stopped and any new units of work will be prevented from beginning. Active LUWs can delay the stopping of the database.

Optional parameters:

DVERIFY Causes the database system to verify its directory before ending. If a directory error is found, the authorized Control Center administrators for the database will receive a message.

TRCPURGE If specified, the database will purge the contents of the trace buffer at shutdown. An output trace file will not be created for the trace buffer in this case.

SCHEDULE SQLEND QUICK (SQ)

Select this option to schedule the database to stop at a later date and time. Any active LUWs will be rolled back when the database is brought back up.

Chapter 19. Database Archiving and Recovery Tools

Overview

There are many options available with DB2 Server for VM to manage the backup and recovery process for a database. Control Center has been designed to be flexible enough to support any archiving methodology which is suitable for a particular environment or even a specific database. You must decide what type of archiving and recovery will be used for each database, as well as define these methodologies to the product for proper automation of both processes. This chapter provides you with all necessary information required to automate your archive and recovery processes using Control Center.

There are several types of archiving available with DB2 Server for VM. You must first decide on the methodology that best meets the needs and requirements of your environment. The *DB2 Server for VM System Administration* manual contains information that will help you decide on the best archiving methodology environment.

Before You Begin

Before you begin the archive and recovery automation process, consider or perform the following:

- Review Chapter 15, “Tape Management Tool” on page 185.
- Logmode (Y,A,L,N) to be used by each of your databases.
- Online (database up) or SQLEND (database down) archives.
- Log archive medium type (tape or disk) if using logmode **L**
- Full archive medium type (tape or disk) if using the database archive utility.
- I/O blocking size to be used for archive output media.
- Number of tapes required for a full archive if using DB2 Server for VM archiving.
- Use of specific pre-allocated tapes or scratch tapes if using tapes as output media and a tape management product is used (VMTAPE, EPIC, DYNAM/T or other).
- Minidisks required to support archiving if using disks as your output media (you will need the cuu for each minidisk to be used).
- Number of log archives (logmode **L** to be taken between your full database archives.
- Number of back-level (not most current) full archives to be kept (available for recovery).

Telling Control Center How to Manage Your Database Archive

Prior to starting an actual Control Center automated DB2 Server for VM database archive event, the product will need to know certain information about the archive about to take place: what archive media will be used, if a tape management system like VMTAPE or DYNAM/T is being used, and so on. Most, if not all, of this information should have been created during your Control Center and DB2 Server for VM database installations and is located in these files:

- SQLMSTR CONTROL
- SQMOUNT EXEC
- SQMSTAPE EXEC
- Database PARMS
- Database TAPES

SQLMSTR CONTROL

This file contains information that describes to the product the interfaces to tape management systems like VMTAPE and DYNAM/T . This file is created during the service machine's installation and is located on its 191 A-disk. Refer to Chapter 4, “Installing the Service Machine” on page 37 and Chapter 11, “Managing the Environment” on page 113 for specific information regarding this file.

SQMOUNT EXEC

This module is created during the service machine's installation process and will be used by the product to request database tape mounts. Refer to Chapter 4, “Installing the Service Machine” on page 37 for specific information regarding this file.

SQMSTAPE EXEC

If a Data Restore backups are used then the SQMSTAPE EXEC can be used to execute some mounts requests depending on values in the database PARMS file. See the Post Installation section of Chapter 4, “Installing the Service Machine” on page 37 for more information about this file.

Database PARMS File

The database PARMS file contains information specific to your database machine. The PARMS file contains information that will instruct the product as to archive medium type to be used, I/O blocking size for the specified media type, and the current series of tapes or disks to be used for the next log and archive to be done. For specific information regarding this file refer to “About the Database Parameters Tool” on page 213.

Changing the Database PARMS File

The CHKINTVL, DISPBIAS, DUMPTYPE, and LTIMEOUT parameters can be modified dynamically from the operator console without stopping and restarting the database. It is important, however, to remember that any changes made to the remaining database PARMS file will not take effect until the corresponding database machine is terminated and then restarted. This is because information regarding a database archive is defined at database startup time. Therefore, changes made to the PARMS file while the database machine is running the database will not be used in the event of an 'online' (database up) or SQLEND (database down) archive activity.

Database TAPES File

The database TAPES file must be created during the process of the database setup with the product. It is critical to note that this file is used at database startup time to correctly establish required CMS FILEDEF and LABELDEF information that will be used in the event of a database archive. It is equally critical to recognize that this file is used by the product during archives to either tape or disk.

The database TAPES file maintains information about two types of archives: log archive and full archive. The full archive output media identifies media (tape or disk) to be used during the database full (not log) archive events. Log archive output media identifies media (tape or disk) to be used during the database log (logmode **L** only) archive events. For specific information regarding this file, refer to Chapter 11, “Managing the Environment” on page 113.

Changing Archive Output Media Types

Changes to full archive output media types (disk to tape or tape to disk) will not take effect until the next product database startup. However, changes to the log archive output media, such as the addition or deletion of media, take immediate effect and will not require the database to be stopped then restarted. For both full and log archives, the media type (tape or disk) *cannot* be changed while the database machine is running, and will require that the database be stopped and restarted. Changes to media types while the database is running could cause database archive *failures*.

Archive Media Types (Tape and Disk)

There are three logmodes (**Y,A,L**) available for multiple user mode (MUM) operation of a database manager. The product fully supports each of these logmodes and their associated archiving methodology to either tape or disk. Furthermore, you can specify that your log archive media be tape and that your full database archive media be disk, or vice versa. The only restriction is that all log archives are of the same media type (tape or disk), and all full archives are of the same media type.

About Archiving to Disk

The product supports the ability to perform database archives to disk. This includes full database archives and log archives. To implement, database TAPES file and the database PARMS file must be defined as described in Chapter 18, “Database Startup and Termination Tools” on page 213. The PARMS file will have disk for Archive_media and/or Logarch_media rather than tape. The blksize parameters should reflect the minidisk block size.

Figure 102 is an example of the PARMS file parameters for archiving to disk.

```
:Archive_media.disk
:Archive_blksize.4097
:Archive_series.100
:Logarch_media.disk
:Logarch_blksize.4097
:Logarch_series.300
```

Figure 102. Example Disk Archiving Database Parameters

Note: You should not use a virtual address of less than 200 as the product detaches 181 thru 185 during DBSTART (database startup and other functions.

Figure 103 is an example of the PARMS file parameters used when full database archives are done to tape and log archives are done to disk (logmode **L**).

```
:Archive_media.tape
:Archive_blksize.28672
:Archive_series.100
:Logarch_media.disk
:Logarch_blksize.4097
:Logarch_series.300
```

Figure 103. Example Disk Log Archive Database Parameters

All entries within the TAPES file must match the format which is appropriate for the entries within the PARMS file. Disk archiving entries are identical to those for tape archiving with the exception of the volid. The volid for disk archiving should be expressed as a filename, filemode, and cuu address. The Tape Management tool should be used to update the database TAPES file.

```
100  ARCHIVE  00000  00:00:00 UNUSED ARCHFN1 ARCHFT1 H 500
100  LOG      00000  00:00:00 UNUSED LOGFN1 LOGFT1 I 501
```

Figure 104. Example Disk Archive Tapes File

The example in Figure 104 indicates that a full database archive for series 100 will use CMS file ARCHFN1 ARCHFT1, located on minidisk H, which is accessed by the database machine with virtual address 500. A log archive for series 100 will use CMS file LOGFN1 LOGFT1, located on minidisk filemode I, which is accessed by the database machine with virtual address 501.

Full Archives to Disk Cannot Span Minidisks

A single archive to disk cannot span multiple minidisks. Therefore, a full database archive must fit on a single minidisk, and a log archive must fit on a single minidisk. There is no multivolume support available for disk similar to that available for tape.

Archive Disks Must Be Linked and Accessed

Another requirement of disk archiving under the product is that each output minidisk must be linked and accessed in read/write mode by the database machine prior to starting the database. The link statements can be added to the database virtual machine VM directory and the access statements can be added to the PROFILE EXEC on the database machine. It is your responsibility to assure that the minidisks are linked to the proper filemodes and virtual addresses as defined in the database TAPES file.

Disk Archiving and Tape File Rotation

In order to support the rotation from one tape series to the next, all *full archives* must be executed with the SQLEND ARCHIVE command. This is so the product can issue CMS FILEDEF and LABELDEF commands that will direct the next archive series to the next SQLEND ARCHIVE command. *Log archives* can be executed while the database remains up and running. The database will prompt the product for a new output FILEDEF when a log archive is performed. This will allow the LARCHIVE function to be done during multiple user access of the database.

You can choose to perform full database archives to tape and do log archives to disk, both to tape, both to disk, or full database archive to disk and log archives to tape. The single minidisk restriction will only apply to the archive types that are done to disk. Therefore, if you choose to perform full database archives to disk, the entire database must be smaller than the available minidisk (essentially limited to one volume). Likewise, if you choose to perform log archives to disk, the entire log file must be smaller than the available minidisk (again limited to one volume).

A separate minidisk should be provided for each archive and log archive file defined in the TAPES file. This would be restricted by the number of filemodes that can be linked by the database machine at a given time. If this is too restrictive, you can define different archive output files on a single minidisk, as long as the minidisk is large enough to handle the multiple files.

Log Archives to Disk

The product maintains the database TAPES file. If you are performing log archives to disk, the database will not allow the same log archive output file name to be used twice. Therefore, the database will generate a new filename for the log archive with filename equal to the database name, and a filetype of the form *mmddyyxx*, where *mmddyy* is the month, day, and year and *xx* is a number between 01 and 99. Control Center will then replace the current filename and filetype in the TAPES file with this new one. This will assure that the uniqueness will be maintained.

This implies that you will still be required to create log entries in the database TAPES file for as many archive log files tapes as you will need between full archives. The FILEMODE and CUU entries will still be used by Control Center to direct the archive output, but the filename and filetype entries will be updated automatically when the log archive is executed. The initial filenames and filetypes supplied within the TAPES file are therefore only required as placeholders. The

product will change them to the suggested names provided by the database when the log archive occurs. Examples of the TAPES file before and after a log archive under this scenario are given in Figure 105 on page 238 and Figure 106 on page 238.

Disk Log Archive Cleanup

Control Center will use the unique filename provided by the database for each disk log archive. Therefore, as files are created on the log archive output disk each time a log archive occurs, eventually the log disk will become full unless old log archives are erased.

The product provides for automatic cleanup of outdated log archive files on disk. The cleanup process is implemented within the tape series rotation function that occurs when a full database archive is performed. Before a new tape series is started, the old disk log archive files are erased for reuse.

At the completion of a full database archive, the product will automatically switch to another group of tapes within the TAPES file for the next archive, to prevent the next archive from writing over the previous archive. Within the group of tapes that was used for the just completed archive, the log archive files will no longer be valid for recovery purposes. It will therefore use the filenames of each log archive file in that group of tapes to perform erases on the target disk.

```
100  ARCHIVE  97001  11:02:03 FILLED ARCHFN1 ARCHFT1 H 500
100  LOG      00000  00:00:00 UNUSED LOGFN1 LOGFT1 I 501
100  LOG      00000  00:00:00 UNUSED LOGFN2 LOGFT2 I 501
200  ARCHIVE  00000  00:00:00 UNUSED ARCHFN2 ARCHFT2 H 500
200  LOG      00000  00:00:00 UNUSED LOGFN3 LOGFT3 I 501
200  LOG      00000  00:00:00 UNUSED LOGFN4 LOGFT4 I 501
```

Figure 105. TAPES File BEFORE Log Archive

```
100  ARCHIVE  97001  11:02:03 FILLED ARCHFN1 ARCHFT1 H 500
100  LOG      97002  05:20:37 FILLED SQLDBA  01029201 I 501
100  LOG      00000  00:00:00 UNUSED LOGFN2 LOGFT2 I 501
200  ARCHIVE  00000  00:00:00 UNUSED ARCHFN2 ARCHFT2 H 500
200  LOG      00000  00:00:00 UNUSED LOGFN3 LOGFT3 I 501
200  LOG      00000  00:00:00 UNUSED LOGFN4 LOGFT4 I 501
```

Figure 106. TAPES File AFTER Log Archive

User Archiving Without Data Restore

This chapter covers topics concerning the archive and recovery of a database and its log using DB2 Server for VM-provided utilities. The product does, however, provide the capability to interface with other non-DB2 Server for VM archive tools. You can, for example, have at your installation Data Dump Restore (DDR) jobs that provide you with minidisk backups of your database's minidisks. The product will allow you to easily interface with any such user archive processes and will even record information pertaining to the user archive, such as start and stop times. For

more information regarding Control Center and user archiving, refer to Appendix F, “User Archiving” on page 515.

User Archiving With Data Restore

Control Center will automate and manage the execution of User Archives using Data Restore. See Chapter 21, “Data Restore BACKUP” on page 265 for a complete description of the backup process.

Database Archiving Tool

The panel shown in Figure 107 is the interface for the Database Archiving tool. Provided are various archiving options, including the ability to schedule an archive or initiate one immediately. This panel is reached by selecting Option **A** on the Control Center Main Menu.

```
 mm/dd/yyyy                    CONTROL CENTER                       hh:mm:ss
*----------------------------- Initiate Archive -------------------------------*
| Option ===>                                                CTRLID: MSTRSRV1   |
| Database ===> SQLDBA                                       NODE:   VMSYSTM1   |
|                                                                               |
|    ************************ ARCHIVE COMMANDS ************************        |
|    SA SQLEND ARCHIVE parms         Full archive, data base down               |
|    SL SQLEND LARCHIVE parms        Log archive, data base down                |
|    SU SQLEND UARCHIVE parms        User archive, data base down               |
|    A  ARCHIVE parms                Full archive, data base up                 |
|    L  LARCHIVE                     Log archive, data base up                  |
|                                                                               |
|    ****************** Data Restore BACKUP COMMANDS ******************        |
|    BU BACKUP  parms                Data Restore Backup                        |
|    BI BACKUP INCREMENTAL parms     Data Restore Incremental Backup            |
|                                                                               |
|    Valid SQLEND Parms:    DVERIFY, TRCPURGE                                   |
|    Valid Incremental Backup Parms: AUTOfull NOAUTOfull                        |
|                                                                               |
|              Enter OPTION and PARMS, press ENTER to Process                   |
|                                                                               |
*-------------------------------------------------------------------SQMAR10-------*
 1 Help    3 End (Cancel)
```

Figure 107. Database Archiving Options Panel

There are several types of archiving available with the database. The types available and the archiving process under the product are described next.

Initiate/Schedule Archive (I/S)

The Archiving option of the panel interface will provide all functions which relate to database archiving. The first two functions offered are archive Initiation and Scheduling. Both functions provide a choice of what archive command you want to be performed.

When an archive is initiated under the product, no further action will be required by you, unless the option chosen is SQLEND UARCHIVE (user-defined archive). If the SQLEND option is selected, the database will be brought down to perform the

archive. The UARCHIVE processing and functions that you must perform are discussed in Appendix F, “User Archiving” on page 515.

Operational Note: The following situation applies **only** if the SQLEND QUICK parameter has been set to 'Y' in the database parameters file (see “Utility Parameters” on page 217.) If users are connected to the database when the SQLEND command is issued, Control Center will detect this and issue the SQLEND QUICK command to bring the database down immediately. To prevent this impact to users, you should issue the SHOW USERS command prior to initiating an archive.

Who Can Run An Archive

Only persons identified as having administrator authority (specified using the Database Parameters tool) for a given database, or Control Center administrators, can initiate database archives. Those persons identified as administrators, for other databases or persons with user or operator authority cannot initiate database recoveries.

Scheduling Your Archive

If you selected the Scheduling option, then after selecting the type to be performed, the Job Scheduling tool is invoked.

Archive Options

Archive commands available with the product include ARCHIVE, LARCHIVE, SQLEND ARCHIVE, SQLEND LARCHIVE, and SQLEND UARCHIVE. Each SQLEND choice can also include the DVERIFY option (which verifies the integrity of the database directory when the database is brought down), and the TRCPURGE option (which purges contents of a trace buffer at shutdown).

LARCHIVE: If the LARCHIVE command is issued without SQLEND, the product will initiate the archive while the database remains operational. Consider the LARCHIVE command rather than the SQLEND LARCHIVE command. This will initiate the log archive process while the database remains up and will reduce the database unavailability. Since a multivolume tape LABELDEF is not issued for log tapes, the product will properly mount and use each successive log tape for log archives between full database archives. **Note:** Control Center was designed to work with a single tape per log archive.

The impact to interactive database users during a log archive with the LARCHIVE command is minimized by the product. When you request the LARCHIVE, it will first issue the tape MOUNT request prior to issuing the LARCHIVE command to the database. When the tape is mounted, the product issues the LARCHIVE command and the log archive begins. When the LARCHIVE command is issued, the database will prohibit any new Logical Units of Work (LUWs) from beginning until the archive begins. If the LARCHIVE command is issued before the tape is mounted, there can be a lengthy delay to database users until the tape is mounted. The product eliminates the tape mount delay for explicit log archives.

Implicit archives will be initiated automatically by the database when the Archpct of the log file is reached. Logmode **A** will cause an implicit full archive, and logmode **L** will cause an implicit log archive. Since this is initiated by the database, the product will not be able to request the tape mount prior to the database archive command. This will result in a significant delay to interactive users while waiting for the tape

mount to occur. You should therefore initiate an explicit log archive command if the SHOW LOG command indicates that the log file is near the Archpct value.

There are several ways that you can prevent an implicit archive from occurring during the busiest interactive usage of a database. One way would be to schedule a log archive (or full archive for logmode **A**) at intervals that will normally prevent the Archpct from being reached. Another way would be to schedule a database monitor routine that will periodically check the log percent (using SHOW LOG) and will explicitly initiate the archive if the log is near the Archpct value.

ARCHIVE: Although it can be executed with Control Center, the ARCHIVE command should not be used without SQLEND (while the database remains operational). Due to the tape LABELDEF commands being issued only when the database starts up, an ARCHIVE without bringing the database down will not reset the tapes for the next archive. If two consecutive archives are done without bringing the database down, then the second archive will use the same tapes as the first archive, eliminating back-level protection.

UARCHIVE: For User Archives that do not use Data Restore, the UARCHIVE command must be issued with the SQLEND keyword. There can be many user-defined methods of archiving a database outside of DB2 Server for VM, that is, use of VMBACKUP, DDR, VMBARS, SYBACK. These options are all done outside of Control Center control. Only Data Restore backups are explicitly supported by Control Center however, the product does aid in starting the process, and in bringing up the database upon completion of the user archive (whether successful or not). For complete details regarding user archiving, refer to Appendix F, “User Archiving” on page 515.

Cancel Archive (C)

The Cancel option can be used to cancel an active archive. Due to the difficulty in determining what step the archive is in, cancellation cannot be performed gracefully without impact to the database. The Cancel command will therefore force a failure of the active archive by issuing a detach command for the tape drive or disk device being used for the database archive process. This will cause a WRITE error on that hardware device and cause the database to crash. For DB2 Server for VM initiated archives, it will then be your responsibility to update the TAPES file and the database status prior to starting the database back up.

For user archives, a user archive cancel process will invoke a customer/user-created cancel routine. The cancel routine process will end with a user archive failed message sent to the database, which will then cause the product to bring up the database and respond to the database, indicating that the user archive failed. For more information on this routine and user archiving, refer to Appendix F, “User Archiving” on page 515.

View Job Schedule (VJ)

The View Job Schedule option displays all scheduled jobs that relate to the specified database. From this display, you will also be able to view, modify, or delete any listed jobs. Refer to “Job Schedule List Tool” on page 176 for additional information.

History (H)

The History option displays the archive history log for the specified database. This log contains the date, time, and tape usage information for all archive and recovery activity. The file is the key component used by the product to automate the database recovery process.

The ARCHHIST file is continually updated by appending any activities that affect database recoverability. These activities include archives, recoveries, logmode changes, and COLDLOGs. The file is edited to remove entries which are no longer applicable (when a new series gets overwritten). Figure 108 provides an example of the ARCHHIST file for database SQLDBA.

```
1997-11-12 15.23.58 EXPLICIT SQLEND ARCHIVE beginning, logmode L
1997-11-12 15.24.01 LOG ARCHIVE completed, series 200.03 volid GP1247 @11/12/97 15.24.01
1997-11-12 15.28.40 FULL ARCHIVE  - filled, series 300 volid GP1248 @11/12/97 15.31.13
1997-11-12 15.31.13 FULL ARCHIVE completed, ( FULL ) series 300 volid GP1249
                    @11/12/97 15.31.13 DRTRANS

1997-11-13 09.37.15 EXPLICIT SQLEND LARCHIVE beginning, logmode L
1997-11-13 09.38.39 LOG ARCHIVE completed, series 300:ehp2 volid GP1250 @11/13/97 09.38.08

1997-11-13 14.02.56 EXPLICIT LOG ARCHIVE (BEFORE UARCHIVE) beginning, logmode L
1997-11-13 14.06.06 LOG ARCHIVE completed, series 300 volid GP1251 @11/13/97 14.05.05
1997-11-13 14.15.37 BACKUP  filled, series 400 volid QU1351 @11/13/97 14.06.06
1997-11-13 14.16.55 BACKUP2 filled, series 400 volid QU1352 @11/13/97 14.06.06
1997-11-13 14.20.30 FULL ARCHIVE completed, ( DUALFULLBACKUP ) series 400 volid USER ARCHIVE
                     @11/13/97 14.06.06

1997-11-13 14.57.37 IMPLICIT LARCHIVE beginning, logmode L
1997-11-13 14.59.31 LOG ARCHIVE completed, series 400 volid GP1252 @11/13/97 14.55.00

1997-11-20 16.32.51 EXPLICIT LOG ARCHIVE (BEFORE UARCHIVE) beginning, logmode L
1997-11-20 16.34.24 LOG ARCHIVE completed, series 400 volid GP1250 @11/20/97 16.34.01
1997-11-20 16.43.06 BACKUP2 filled, series 400.01 volid QU1365 @11/20/97 16.34.24
1997-11-20 16.44.48 BACKUP  filled, series 400.01 volid QU1364 @11/20/97 16.34.24
1997-11-20 16.48.28 FULL ARCHIVE completed, ( DUALINCBACKUP ) series 400.01 volid USER ARCHIVE
                    @11/20/97 16.34.24 INCREF 11/13/97 14.06.06

1997-11-24 17.08.26 EXPLICIT LOG ARCHIVE (BEFORE UARCHIVE) beginning, logmode L
1997-11-24 17.09.35 LOG ARCHIVE completed, series 400.01 volid GP1253 @11/24/97 17.08.03
1997-11-24 17.16.42 BACKUP2 filled, series 400.02 volid QU1369 @11/24/97 17.09.35
1997-11-24 17.20.05 BACKUP  filled, series 400.02 volid QU1368 @11/24/97 17.09.35
1997-11-24 17.24.17 FULL ARCHIVE completed, ( DUALINCBACKUP ) series 400.02 volid USER ARCHIVE
                    @11/24/97 17.09.35 INCREF 11/13/97 14.06.06

1998-02-02 16.26.46 EXPLICIT LOG ARCHIVE (BEFORE UARCHIVE) beginning, logmode L
1998-02-02 16.28.16 LOG ARCHIVE completed, series 400.02 volid GP1254 @02/02/98 16.27.01
1998-02-02 16.37.21 BACKUP2 filled, series 100 volid QU1373 @02/02/98 16.28.15
1998-02-02 16.41.08 BACKUP  filled, series 100 volid QU1372 @02/02/98 16.28.15
1998-02-02 16.47.00 FULL ARCHIVE completed, ( DUALFULLBACKUP ) series 100 volid USER ARCHIVE
                    @02/02/98 16.28.15

1998-02-10 09.12.03 EXPLICIT SQLEND LARCHIVE beginning, logmode L
1998-02-10 09.14.20 LOG ARCHIVE completed, series 100 volid GP1255 @02/10/98 09.13.38
```

Figure 108. Example ARCHHIST File Showing Incremental Backup, Full Backup, and DB2 Archives

Log (L)

The Log option displays the latest archive log for the specified database. This log contains the console output that was produced during the last archive. This can be used for failure analysis when an archive is unsuccessful.

Figure 109 on page 243 provides an example of the ARCHLOG file for an archive under Control Center. All **highlighted** entries in the example are responses made automatically by the product. All other entries are messages received from the database machine during the archive process.

```
00:31:46 SQLDBA SQLEND ARCHIVE issued from MSTRSRV1 AT VMSYSTM1
00:31:46  ARI0028I The database manager is terminating.
00:31:46  ARI0065I Operator command processing is complete.
00:31:46  ARI2008I Archive is about to be started.
00:31:46  ARI0254I The database manager is initiating a log archive.
00:31:46           When it is completed, the database manager will
00:31:46           process the database archive request.
00:31:50  ARI0293I Archive is starting.
00:31:51  ARI0239I External labeling of this archive is:
00:31:51               Type:      log  archive
00:31:53               Timestamp: 01-10-97  00:31:46
00:31:53  ARI0252I     Medium:    disk  SQLDBA  01109701 F
00:31:56  ARI0246D The above information describes the log archive
00:31:56           about to be done.  Enter either:
00:31:56             CONTINUE   to proceed using the output medium
00:31:56                          indicated, or
00:31:56             CHANGE     to change this medium.
00:31:56 CHANGE
00:31:57  ARI0263D To direct the log archive to tape, enter tape followed
00:31:57           by the tape address (CUU) to be used.
00:31:57           To direct the log archive to disk, enter disk followed
00:31:57           by the disk file name,  file type, and  file mode.
00:31:57           If you chose disk, the default file is:
00:31:57           SQLDBA  01109201 F
00:31:57 disk = = G
00:31:58  ARI0239I External labeling of this archive is:
00:31:58               Type:      log  archive
00:32:00               Timestamp: 01-10-97  00:31:46
00:32:00  ARI0252I     Medium:    disk  SQLDBA  01109701 G
00:32:02  ARI0246D The above information describes the log archive
00:32:03           about to be done.  Enter either:
00:32:03             CONTINUE   to proceed using the output medium
00:32:03                          indicated, or
00:32:03             CHANGE     to change this medium.
00:32:03 CONTINUE
00:32:08  ARI0292I Archive is completed.
00:32:12  ARI0293I Archive is starting.
00:32:12  ARI0239I External labeling of this archive is:
00:32:12               Type:      database  archive
00:32:13 Tape mount issued for: VM1203 181
00:32:13               Timestamp: 01-10-97  00:32:08
00:32:13  ARI0299A Ready archive output volume. Enter the CUU.
00:36:34  tape 3F21 ATTACHED TO SQLDBA  0181
00:36:36 181
01:20:22  DMSTLM428I TAP1(181) EOV1 label written on VM1203
01:20:23  tape 0181 DETACHED
01:21:57  tape 3F21 ATTACHED TO SQLDBA  0181
01:26:48  ARI0292I Archive is completed.
01:26:54  ARI0032I The Database manager has terminated.
01:26:54  ARI0043I Database manager return code is '0'.
```

Figure 109. Example ARCHLOG File Showing DB2 Archive

View Tapes (VT)

The View Tapes options displays the database TAPES file for the specified database. Refer to “Overview” on page 185.

Tape Maintenance (TM)

The Tape Maintenance option provides you with a method of updating the database TAPES file. Refer to “Overview” on page 185.

Scratch Tape Acquisition (ST)

The Scratch Tape Acquisition option acquires a scratch tape and automatically adds it to the database TAPES. It is applicable only when using VMTAPE or DYNAMT. If you have DYNAMT installed and have not specified DYNOPEN as the DYNAMT method for handling tapes, or DYNMOUNT is the chosen method for tape command processing, then DYNAMT will support scratch tape acquisition. Otherwise, DYNOPEN will not support scratch tape acquisition.

Database Recovery Tool

Introduction

The panel shown in Figure 110 is the interface for the Database Recovery tool. Provided are various recovery options, including the ability to schedule a recovery or initiate one immediately. This panel is reached by selecting Option **R** (Database Recovery) on the Control Center Main Menu panel.

If you are using Data Restore as your vehicle for recovery, refer to the appropriate chapters in this manual for more detailed information.

```
 mm/dd/yyyy                    CONTROL CENTER                      hh:mm:ss
*-------------------------------- Recovery ---------------------------------*
| Option ===>                                              CTRLID: MSTRSRV1 |
|    Database ===> SQLDBA                                  NODE:   VMSYSTM1 |
|                                                                           |
|  I  INITIATE RECOVERY          Immediate recovery initiation              |
|  S  SCHEDULE RECOVERY          Schedule a recovery                        |
|  C  CANCEL RECOVERY            Immediate cancel of active recovery        |
|  R  INITIATE FAILURE RESTART   Restart the log archive recovery process   |
|  DR RESTART DATA RESTORE       Restart the Data Restore part of recovery  |
|  VJ VIEW JOB SCHEDULE          View database job schedule                 |
|  H  HISTORY                    Display history of archives                |
|  L  LOG                        View log of previous recovery              |
|  VT VIEW TAPES                 View tape catalog                          |
|  TM TAPE MAINTENANCE           Make changes to tape catalog               |
|  ST SCRATCH TAPE ACQUISITION   Acquire a new SCRATCH tape and             |
|                                   add volid to TAPES file                 |
|                                                                           |
|           Enter OPTION, select DATABASE, press ENTER to process           |
|                                                                           |
*-------------------------------------------------------------SQMRECOV----*
 PF:   1 Help    3 End
```

Figure 110. Database Recovery Option Panel

The database recovery feature supports the automatic restore of a database that has been running with either logmode **A, L,** or **Y** and archiving using the database archive facility to tapes or disk, and additionally archiving using 'user archive'. Thus, the product supports automatic recoveries of databases with a STARTUP parm of **R** or **U**.

The recovery process obtains information about database archive activity from the ARCHHIST file and not from the database history records. If you have not been using the product to manage a database's archives, COLDLOGs, logmode switches, and recoveries, then recovery information available to it will be incomplete. In other words, the recovery process only knows as much as the product has been told.

Usage Consideration: Before you invoke any kind of database recovery, be certain you are completely familiar with the database recovery topics and considerations outlined in the *DB2 Server for VM System Administration* manual.

How Automated Recovery Works

The automatic recovery features were created to eliminate the need for manual support during a database restore. The process of running a manual database restore requires stopping the database, pre-recovery file and label definitions, startup parameter modifications, tape mounts, tape confirmations, starting the database, responding to console messages, interpreting console messages, reacting to recovery errors, keeping a recovery log, post-recovery database environment changes, and so on. The automated recovery feature will perform all these activities from start to completion of a database recovery.

Who Can Run a Recovery

Only those persons identified as administrators for a given database or Control Center administrators can initiate database recoveries. Persons authorized as administrators to other databases and persons with Control Center user or operator-level authority cannot initiate recoveries.

How Automated Recovery Relates to Automated Archive

The recovery process of the product uses information created during database archiving. If there is no archive history data for a database, automatic recovery is still possible but starting the recovery will require that you perform some manual setup work and is therefore not suggested. A recovery control file is created by the product as a result of reviewing a database's ARCHHIST file data. The control file is sent to you for review. You will then be able to review exactly how it will control the recovery and decide to proceed or cancel the initiation process.

Database After the Recovery

After completion of the recovery, the database machine will be stopped, its startup parameter changed to W, and then restarted.

Next Archive Series after the Recovery: The next database archive series to be used in the event of a database archive or log archive should be reviewed after the recovery. The recovery process does not reset the archive tape rotation series.

Logmode after the Recovery: The database logmode will be set to the logmode used for the recovery. You must use the logmode switching feature of the product to change this logmode if it is not the one required.

Recovery from a User Archive: For information regarding the product and database recoveries involving user archives, refer to Appendix F, “User Archiving” on page 515.

If you are using Data Restore to RESTORE your database or objects within the database, see Chapter 22, “Data Restore RESTORE” on page 277 for more detailed information.

Initiate/Schedule Recovery (I/S)

Although command mode execution is possible for database recovery, the preferred method is to use the panel interface to initiate the recovery process. Several panels will prompt you for critical recovery information that will tell the product how to answer the many questions that will be asked by the database during the actual restore.

The following sequence of panels will provide an overview of the recovery initiation process using Control Center.

The panel shown in Figure 111 allows you to select the logmode that you wish to use for the recovery (either L or A), and allows you to specify ALL if you want to see all available restore sets (all previous archives). Leave blank (the default) to display only the most recent archive. If your database has been running with logmode L, then you should specify L here. If your database runs with logmode A or Y, then specify A for this parameter.

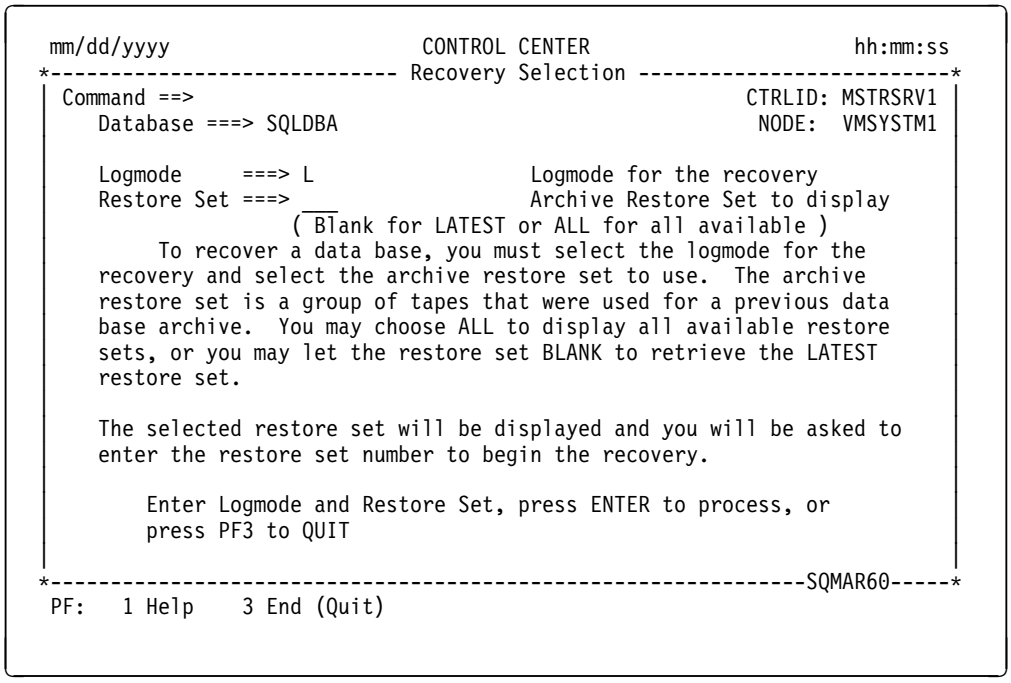


Figure 111. Control Center Recovery Selection Panel

Recovery Logmode

Automatic database recoveries can be specified to use either logmode **A** or **L**. The database STARTUP parameters will be set to **R** when the database is to be restored from an archive that was created by database archiving facilities. It is set to U when the database is to be restored from an archive that was created by user archive facilities. The database uses information in its history area to associate log archives with database archives. When a DB2 Server for VM-based recovery is started, the database will look at identification information on the database archive tape or disk that is mounted for the recovery. The database then reviews its history information about this archive to determine all log archives associated with this database archive, if any. Like the database, the product reviews its own history records and determines what tapes or disks or user restore sets are to be used during the recovery process.

Selecting a Logmode for the Recovery

You can specify a recovery logmode that is different from the logmode used when the database archive was created. Note, however, that after the recovery process completes the database will be running under the recovery logmode specified. To return to the original logmode, you must initiate a Control Center logmode switch process.

Database Recovery Restore Sets

The files identified by the product that drive a database recovery are referred to as a 'restore set'. Restore sets determined by the product will vary depending upon the logmode that you choose to recover with as well as the type of medium used (tape or disk) and the type of archive (user archive or database archive). In most cases you will choose a recovery logmode that matches the logmode associated with the database archive taken. Why you might recover a database using a different logmode than what is normally used is a topic that will not be discussed in this manual. Consult the *DB2 Server for VM System Administration* manual for more information about database recovery logmodes.

About Logmode L Recoveries

Recovering a database using logmode **L** tells the product to create a restore set containing all database archive files (or user archive set) and all associated log archive tapes. It will scan its history information for this database to determine what log archives are associated with the specified database archive. If the product had been used to perform database activities which cause log tape continuity to be broken, then the product will create a restore set considering such events. Consult the *DB2 Server for VM System Administration* manual for a list of activities which cause log continuity to be broken. If you do not use the product to process these types of activities, then your restore set created might contain too many log tapes. It will, in this case, restore as many tapes as possible before the database ends the recovery process successfully. Be sure to review the recovery log that the product sends to you upon completion of the recovery process to be sure that the recovery was processed as expected.

Logmode L Back-Level Recoveries: You can recover using logmode **L** from a back-level database archive. A back-level database archive is not the most current database archive known to DB2 Server for VM. In such a case, it may be possible to recover to the back-level database archive and use log archive tapes to recover forward to the most current database, provided that log continuity had been maintained from the back-level archive. In all cases, the database will permit the recovery process to apply all log archives that history records have associated with a specific database archive.

Logmode L and Specifying an END RESTORE Volume: An END RESTORE volume can be specified to the recovery process. The volume specified can only be a log archive volume that you do not want to restore. The product will recover all log archives up to but not including the END RESTORE volume that you specify. Any log archive volumes that would have been recovered after this volume are also not recovered. In other words, the recovery ends after restore of any log archives up to but not including the specified volume. Great care should be taken to specify the correct volume that you wish to end the database recovery at, because after you have recovered it may not be possible to recover the end restore log and any subsequent log archives. Consult the *DB2 Server for VM System Administration* manual for information on specifying END RESTORE volumes.

If you do not want the currently active log to be restored during the database recovery, then specify ACTIVE for the END RESTORE volid. The product will recover all log archives prior to the currently active and yet-to-be-archived log.

Operational Note: Even though the active log will not be recovered, it will still be log archived.

About Logmode A Recoveries

Recovering a database using logmode **A** tells the product to create a restore set containing only database archive files. You will also specify what is to be done with the active yet-to-be-archived log. The active log can be emptied (COLDLOGed) before the recovery process begins, or left intact to be restored as a part of the recovery.

Logmode A Back-Level Recoveries: You can recover using logmode A from a back-level database archive. Specify that a COLDLOG be performed; otherwise, changes contained within the currently active log will be restored to the database. This could cause many data consistency problems.

Logmode A Recovery when there are Log Archives Associated: A database can be recovered using logmode **A** using a database created using logmode **L**. The database will detect that there are log archives available and will ask if the recovery logmode should be switched to **L** so that they can be applied. The product will tell the recovery process not to switch to logmode **L** and not to apply the log archives. If you want the log archives applied, then logmode **L** should be specified to the product for the recovery logmode. Figure 112 displays the logmode **L** panel. This panel will differ slightly if logmode **A** is chosen.

```
 mm/dd/yyyy                    Control Center                        hh:mm:ss
*-------------------------- Restore Set Selection --------------------------*
| Command ==>                                                CTRLID: MSTRSRV1 |
|    Database ===> SQLDBA                                    NODE:   VMSYSTM1 |
|                                                                             |
|  Logmode           ===> L           Logmode for the recovery                |
|  Restore Set       ===> 1           Archive Restore Set number to Use       |
|                                     (Select from the list below)            |
|  Partial Restore   ===> N           (Y,N) Perform PARTIAL Restore           |
|  Use Data Restore  ===> Y           (Y,N) Restore from a Data Restore       |
|                                     Backup or Translated archive.           |
|        ------------------ Valid Restore Sets -------------------            |
|  1 (07/03/98), 2 (07-02-98), 3 (06/25/98), 4 (06/25/98)                     |
|                                                                             |
|                                                                             |
|   NOTE:   If you do NOT want to perform a COMPLETE restore, but instead     |
|           only want to apply selected LOGS, then enter Y for Partial        |
|           Restore and you will be prompted for additional information.      |
|                                                                             |
|    Enter Restore Set and press ENTER to process, or press PF3 to QUIT       |
|                                                                             |
*-----------------------------------------------------------------SQMAR70-----*
 PF:   1 Help    3 End (Quit)
```

Figure 112. Control Center Restore Set Selection Panel

If Data Restore is enabled and you have been using Data Restore BACKUPs and TRANSLATEs, refer to section Chapter 22, “Data Restore RESTORE” on

page 277. If you are not using Data Restore, the value in the *Use Data Restore* field in the menu shown in Figure 112 should be *N*.

Parameter	Description
Selecting a Restore Set	Figure 112 on page 248 allows you to select one of the valid restore sets that were displayed in the earlier report. The restore sets will always be numbered from 1 up, from the most recent archive to the oldest.
Partial Restore Option	One other option is available for the recovery process from this screen: Specifying that you wish to perform a partial restore, rather than a complete restore. A partial restore is defined as applying fewer log archive files than are available for bringing the database back to the current time. This option should only be used for special situations as defined in the *DB2 Server for VM System Administration* manual.
Use Data Restore	(Y,N.) 'Y' (Yes) indicates that you wish to recovery from a Data Restore BACKUP (either normal, incremental or full) or a translated archive. If you select 'Y', you will be taken to the 'Data Restore Recovery Options' menu. • If the restore set chosen includes a normal BACKUP, an Incremental Backup, or a Full Backup, then this value will automatically be set to 'Y'. If the restore set chosen includes an archive that has been translated by Data Restore, then you have the option of choosing 'Y' to recover using the translate or 'N' to recover from the archive. If a partial log recover has been selected, you will be taken to the 'Partial Log Recovery' menu after completing the 'Data Restore Recovery Options' menu. • Selecting 'N' (No) indicates that you wish to recovery from a DB2 Server for VM archive. If the restore set chosen does not include a Data Restore BACKUP (either normal, incremental or full) or a translated archive, then this value will automatically be set to 'N'.

Figure 113 on page 250 shows the Data Restore Recovery Options menu.

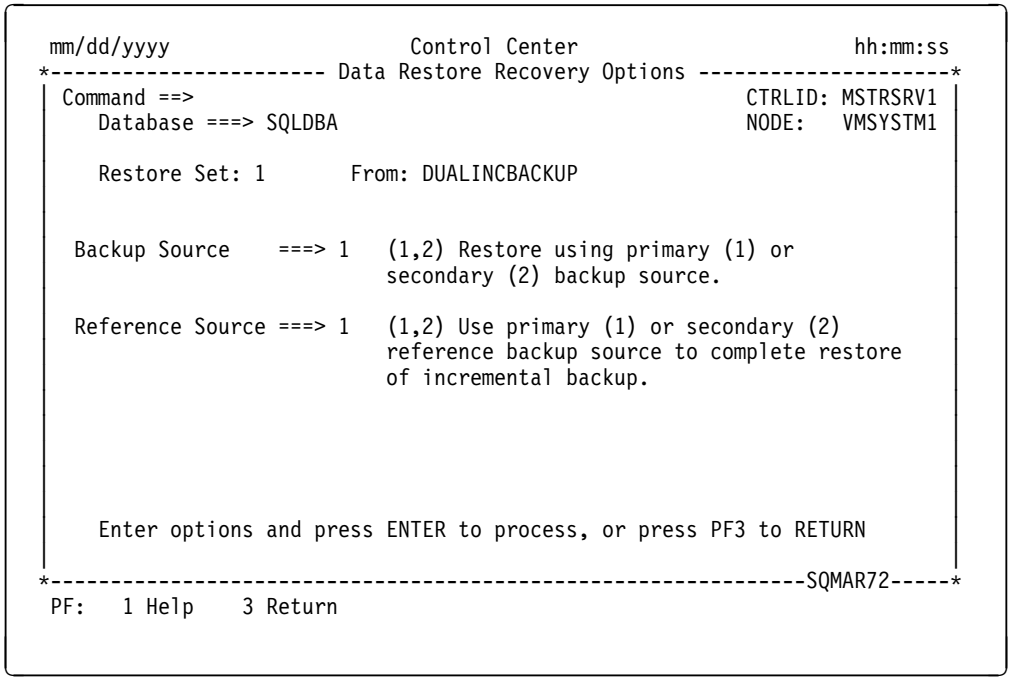


Figure 113. Data Restore Recovery Options Menu

Parameter **Description**

Backup Source

(1,2.) If recovering from a dual BACKUP you can select (1) to recover using the primary backup source or (2) to recover using the Secondary backup source. A translated archive only generates a primary backup. The default is the primary (1) backup source.

Reference Source

(1,2, or leave blank) If you are recovering from a backup then leave the value blank. If recovering from an Incremental Backup and the incremental reference is a dual Full Backup you can select (1) to recover using the primary reference backup source or (2) to recover using the Secondary reference backup source. A translated archive full only generates a primary reference backup. The default is the primary (1) reference backup source.

Final Confirmation Before Starting the Recovery

After making your entries, press the ENTER key; a report will be displayed containing historical archiving information which pertains to your selections. View this information for accuracy and decide on the restore set that you wish to use. After pressing PF3 to exit the report display, a panel will be displayed that will ask you to specify a restore set number. This panel reflects a selected logmode of L and will differ slightly if logmode **A** is chosen.

Recovery Scheduling (S)

Finally, if you initially selected Option **S** from the main recovery panel, you will be presented with a screen that will allow you to specify when your recovery is to be started.

Cancel Recovery (C)

A request can be made by authorized persons to stop a currently active database recovery. The product will then update recovery status information indicating that the recovery is to be interrupted. It will wait until there is an opportunity to respond to the database recovery prompt messages in order to gracefully terminate the recovery processing. This is done so that, if possible, the recovery can be restarted from the point of termination, to avoid having to restart the database's recovery from the beginning. In the case of a recovery from a user archive, the product invokes the routine specified in the Cancel-routine field of the database PARMS file. This routine executes the command to the database to initiate the cancel of the user recovery if so coded by the user. The database remains down. If there is no cancel-routine specified, the product issues a warning message to the administrator in a note file to start the database *after* recovery from a user archive has completed. The database remains down.

The cancel recovery option is invoked by using the recovery menu system. If a recovery has been scheduled and has not started yet, then simply display the recovery schedule and specify that it is to be deleted.

Initiate Failure Restart (R)

If the log archive portion of a recovery fails, it may be possible to resolve the problem that caused the error and restart the recovery process from the point of error, avoiding restarting from the beginning. If the product detects an error during the recovery or the recovery is stopped by an administrator, then the database STARTUP parameter is set to W. After resolution of the problem, the recovery can be restarted using the the product recovery restart option. Please note, however, that if the recovery process had not started the log archive restore portion of the recovery process, then the recovery must be restarted from the beginning. The start of the log archive restore is indicated by the recovery information message ARI0260I. If the message ARI0260I was displayed by database, then the recovery can be restarted from the point of failure. If it had not been displayed, then the recovery must be restarted from the beginning.

Restart Data Restore (DR)

If you are recovering from a Data Restore Backup and the recovery of the directory and data disks fails, it may be possible to resolve the problem that caused the error and restart the restore process on the Data Restore machine. If a Data Restore restore fails the database will be left down and the STARTUP mode parameter will be "U".

View Job Schedule (VJ)

The View Job Schedule option displays all scheduled jobs that relate to the specified database. From this display, you will also be able to view, modify, or delete any listed jobs. Refer to “Job Schedule List Tool” on page 176 for additional information.

History (H)

The database's ARCHHIST file will be updated as if a database archive had occurred. If any log tapes were recovered, then the ARCHHIST file will be updated to indicate that log archives were done. This is done because recovery history information is recorded by the database in its history log, changing future restore set information. The recovered database archive and any recovered log archives will be distinguished from regular archive activity in the ARCHHIST file by a status of RECOVERED to the right of the archive record.

Log (L)

During recovery of an archived database, status messages about key events during the recovery process will be sent out to all the database's administrators. The Control Center database status (refer to Chapter 17, “Database Status Tool” on page 203) will be continually updated and will indicate exactly what the recovery process is doing. You can also browse the database's RECOVLOG kept by the product, which is a log of all database console messages updated during the recovery process. During the database recovery portion of a user-archived database, there is no reflection of the activity going on, since this is being done outside of the product. When log recovery starts, the status activity described above resumes.

A copy of this log is sent to all the database's administrators for review.

Figure 114 on page 253 provides an example of the RECOVLOG file for a database recovery.

```
02:44:31 SQLDBA Recovery Starting!
02:44:32  ARI0025I Database SQLDBA is STARTING
02:45:30  ....... START SQLSTART EXEC: 02:45:28 EST
02:45:30  ARI0663I FILEDEFS IN EFFECT ARE:
02:45:31  Z        disk     DMSNAM   LOADLIB  *
02:45:31  ARIARCH  TAP1  SL  00001  VOLID VM3101
02:45:31  ARILARC  TAP3  SL  00001
02:45:31  ARITRAC  disk     TRACE    DATA     A1
02:45:31  ARISQLLD disk     ARISQLLD LOADLIB  Q1
02:45:32  Bdisk    disk     200
02:45:32  LOGDSK1  disk     201
02:45:32  LOGDSK2  disk     2F1
02:45:32  DDSK1    disk     202
02:45:33  DDSK2    disk     203
02:45:33  DDSK3    disk     204
02:45:45  ARIUSRDD disk     USERLIB  LOADLIB  *
02:45:46  ARI0025I THE PROGRAM ARISQLDS IS LOADED AT 88D000
02:45:46  ARI0025I THE PROGRAM ARIXRDS IS LOADED AT 7B7000
02:45:47  ARI0025I THE PROGRAM ARIXSXR IS LOADED AT 982000
02:45:47  ARI0025I THE PROGRAM ARICMOD IS LOADED AT 980000
02:45:47  ARI0015I ACCOUNT PARAMETER VALUE IS D
02:45:48  ARI0015I DUMPTYPE PARAMETER VALUE IS P
02:45:48  ARI0015I LOGMODE PARAMETER VALUE IS L
02:45:48  ARI0015I STARTUP PARAMETER VALUE IS R
02:45:49  ARI0015I SYSMODE PARAMETER VALUE IS M
02:45:49  ARI0015I EXTEND PARAMETER VALUE IS N
02:45:50  ARI0015I CHARNAME PARAMETER VALUE IS ENGLISH
02:45:50  ARI0015I DBNAME PARAMETER VALUE IS SQLDBA
02:45:50  ARI0015I PARMID PARAMETER VALUE IS SQLDBA
02:45:51  ARI0015I TRACDBSS PARAMETER VALUE IS 00000000000
02:45:51  ARI0015I TRACRDS PARAMETER VALUE IS 000000
02:45:52  ARI0016I ARCHPCT PARAMETER VALUE IS 80
02:45:52  ARI0016I CHKINTVL PARAMETER VALUE IS 30
02:45:53  ARI0016I NCSCANS PARAMETER VALUE IS 30
02:45:53  ARI0016I NCUSERS PARAMETER VALUE IS 15
02:45:53  ARI0016I NDIRBUF PARAMETER VALUE IS 125
02:45:54  ARI0016I NLRBS PARAMETER VALUE IS 11290
02:45:54  ARI0016I NLRBU PARAMETER VALUE IS 1500
02:45:55  ARI0016I NPAGBUF PARAMETER VALUE IS 200
02:45:55  ARI0016I SLOGCUSH PARAMETER VALUE IS 90
02:45:56  ARI0016I SOSLEVEL PARAMETER VALUE IS 10
02:45:56  ARI0016I DISPBIAS PARAMETER VALUE IS 7
02:45:56  ARI0204D RESTORE FROM ARCHIVE INVOKED
02:45:57 RESTORE
02:45:57  ARI0204D ... CURRENT DATA BASE WILL BE DESTROYED
02:45:57  ARI0204D REPLY RESTORE TO CONTINUE, OR CANCEL TO CANCEL
02:45:58  ARI0295A READY ARCHIVE INPUT VOLUME. REPLY CUU OR CANCEL
02:45:58 Tape mount issued for: VM3101   181
02:48:07  tape 5323 ATTACHED TO SQLDBA  0181
```

Figure 114 (Part 1 of 2). Example Database RECOVLOG File

```
02:48:09 0181
02:48:12  ARI0289I RESTORING DIRECTORY disk
03:07:57  ARI0290I RESTORING DATA disk
03:21:09  DMSTLM427I TAP1(181) EOV1 label read
03:21:10  tape 0181 DETACHED
03:22:19  tape 5323 ATTACHED TO SQLDBA  0181
06:31:38  ARI0291I SYSTEM RESTORE FROM DIRECTORY disk AND DATA disk
06:31:39 SQLDBA DIRECTORY & DATA Disks RECOVERED!
06:31:40  ARI0291I OF DATA BASE ARCHIVE COMPLETED
06:31:58  ARI0255I THE DATABASE MANAGER IS INITIATING A LOG ARCHIVE.
WHEN IT COMPLETES,
06:31:58  ARI0255I THE RESTORE PROCESS WILL CONTINUE.
06:32:01  ARI0293I ARCHIVE STARTING
06:32:02 SQLDBA Log Archive started.
06:32:03  ARI0239I EXTERNAL LABELING OF THIS ARCHIVE IS:
06:32:03  ARI0239I      TYPE:      LOG ARCHIVE
06:32:04  ARI0239I      TIMESTAMP: 02-16-97  06:31:58
06:32:05  ARI0299A READY ARCHIVE OUTPUT VOLUME. REPLY CUU
06:33:19  tape 1320 ATTACHED TO SQLDBA  0183
06:33:29  ARI0292I ARCHIVE COMPLETED
06:33:31 LOG ARCHIVE completed, series 300 volid VM3201
06:33:32  ARI0260I THE RESTORE SET FOR THIS DATA BASE ARCHIVE FOLLOWS:
06:33:32  ARI0238I      DATA BASE ARCHIVE        21:50:57
06:33:32  ARI0261I      LOG ARCHIVE              15:30:08
06:33:32  ARI0261I      LOG ARCHIVE              19:33:39
06:33:33  ARI0239I EXTERNAL LABELING OF THIS ARCHIVE IS:
06:33:33  ARI0239I      TYPE:      LOG ARCHIVE
06:33:33  ARI0239I      TIMESTAMP: 02-10-97  15:30:08
06:33:34  ARI0250D THE ABOVE INFORMATION DESCRIBES THE NEXT LOG ARCHIVE
06:34:01  ARI0250D TO BE USED IN THE RESTORE PROCESS.
06:34:01  ARI0250D REPLY 'CONTINUE' TO RESTORE THIS LOG ARCHIVE, OR
06:34:01  ARI0250D 'STOP SYSTEM' TO INTERRUPT THIS RESTORE PROCESS, OR
06:34:02  ARI0250D 'END RESTORE' TO END THIS RESTORE PROCESS.
06:34:00 CONTINUE
06:34:02  tape 0183 DETACHED
06:34:03  ARI0295A READY ARCHIVE INPUT VOLUME. REPLY CUU OR CANCEL
06:35:05  tape 1321 ATTACHED TO SQLDBA  0183
06:35:07 0183
06:35:10  ARI0240I RESTORING LOG disk
06:41:19  ARI0283I LOG ANALYSIS COMPLETE
06:41:19  ARI0282I LUW UNDO COMPLETE
07:12:04  ARI0281I LUW REDO COMPLETE
07:12:08  ARI0239I EXTERNAL LABELING OF THIS ARCHIVE IS:
07:12:08  ARI0239I      TYPE:      LOG ARCHIVE
07:12:08  ARI0239I      TIMESTAMP: 02-11-97  19:33:39
07:12:08  ARI0250D THE ABOVE INFORMATION DESCRIBES THE NEXT LOG ARCHIVE
07:12:13  ARI0250D TO BE USED IN THE RESTORE PROCESS.
07:12:13  ARI0250D REPLY 'CONTINUE' TO RESTORE THIS LOG ARCHIVE, OR
07:12:14  ARI0250D 'STOP SYSTEM' TO INTERRUPT THIS RESTORE PROCESS, OR
07:12:14  ARI0250D 'END RESTORE' TO END THIS RESTORE PROCESS.
07:12:13 END RESTORE
07:12:21  ARI0060I DATABASE MANAGER INITIALIZATION COMPLETE.
07:12:22  ARI0045I READY FOR OPERATOR COMMUNICATIONS
07:12:23  tape 0183 DETACHED
07:12:24  ARI028I The database manager is terminating
07:12:24  ARI0065I Operator Command processing is complete
07:12:26  ARI0032I THE DATABASE MANAGER HAS TERMINATED.
07:12:26  ARI0043I DATABASE MANAGER RETURN CODE IS '0'.
07:12:26 SQLDBA RECOVERY SUCCESSFUL! Database is being brought back up.
07:12:28 SQMA043I=> Database being started with logmode =  L
```

Figure 114 (Part 2 of 2). Example Database RECOVLOG File

View Tapes (VT)

The View Tapes options displays the database TAPES file for the specified database. Refer to Chapter 15, “Tape Management Tool” on page 185.

Tape Maintenance (TM)

The Tape Maintenance option provides you with a method of updating the database TAPES file. Refer to Chapter 15, “Tape Management Tool” on page 185.

Additional Recovery Considerations

The following section covers some additional recovery topics and how the product can be used to help facilitate these processes.

Considerations when Running a Manual Recovery

If you decide to run your recovery manually, then consider manually updating the ARCHHIST file for the database, indicating in order of recovery each of the archive recovery events. The history records should look just like any other full database archive records, followed by any log archives that were restored. By doing this you will keep the product synchronized with the database's history information.

Considerations when Reconfiguring Logs

If you reconfigure your database log minidisks, you should update the database's ARCHHIST file manually to indicate that a COLDLOG has been performed. This will indicate that log continuity has been broken and will help the product in accurately determining database restore set information. Simply go to the bottom of the database's archive history file and add one line containing the string COLDLOG. This will indicate to the product that log continuity has been broken.

How to Run Filtered Log Recoveries

Filtered log processing can be run under product control, but requires some up-front setup work before the database recovery or restart can be invoked. This feature was not automated because filtered log processing is not encouraged.

Usage Consideration: If you want to specify filtering during a database recovery and you want to specify a different EXTEND input file before each log archive restore, then you must run the recovery manually. Specifying different EXTEND input files requires that you stop the database before recovering the log tape and perform a CMS FILEDEF to identify a new EXTEND input file. You would then restart the database with a STARTUP=W to resume the recovery.

Filtered Log Processing during Database Restart

The following is a list of steps to be followed when filtering during a database restart:

1. Create an EXTEND input file.
2. A Control Center administrator places an EXTEND file on the database's 191 A-disk.
3. Set the database's STARTUP parameter to W and the EXTEND parameter to Y.
4. Use the product to send a CMS FILEDEF for ddname ARIEXTND to the database machine (FILEDEF ARIEXTND DISK database EXTEND A).
5. Restart the database.
6. Verify the results.

7. Set the database's EXTEND parameter back to N.

Filtered Log Processing during Database Recovery

The following is a list of steps to be followed when filtering during a database recovery:

1. Create an EXTEND input file.
2. A Control Center administrator places an EXTEND file on the database's 191 A-disk.
3. Use the product to stop the database.
4. Send a CMS FILEDEF for ddname ARIEXTND to the database machine (FILEDEF ARIEXTND DISK database EXTEND A).
5. Set the database's EXTEND startup parameter to Y.
6. Restart the database machine.
7. Invoke a database recovery using the product recovery menus.
8. Verify the results.
9. Set the database's EXTEND startup parameter back to N.

Chapter 20. Data Restore Support

Overview

When the Data Restore feature has been installed and enabled, Control Center gives users the ability to automate, manage, track and schedule Data Restore functions on one or more DB2 Server for VM databases.

See Chapter 9, “Data Restore Setup with Control Center” on page 93 for installation and setup instructions.

Data Restore UNLOADS or retrieves data and system information about a database by directly reading the database data and directory disks. In this way the normal database routines are bypassed and commands can be issued against the database whether it is running ONLINE or OFFLINE. The Data Restore machine must have read/write access to all of the database's data, directory and log disks.

Description

Control Center support of the Data Restore will:

- Automate control of Data Restore functions
- Provide a menu interface to setup, control, schedule and execute Data Restore functions
- Allow for remote or local control of Data Restore functions
- Provide job scheduling capability of Data Restore functions
- Track and maintain historic information concerning Data Restore functions

Data Restore Functions

With Control Center, a user can manage and execute the following Data Restore functions:

Function	Definition
APPLYLOG	Performs log forward recovery. Executes all of the SQL statements for the table(s) generated by the RELOAD function. A user can indicate where to stop a log forward table recovery by indicating the time/date stamp of the LUW listed in the LISTLOG output. The RELOAD with RECOVERY=YES and LISTLOG functions must be executed first. LISTLOG is only needed if the timestamp is specified (which is most of the time).
BACKUP	A BACKUP is a a User Archive using Data Restore BACKUP that backs up the whole database. An Incremental Backup is a USER ARCHIVE using Data Restore BACKUP with the INCREMENTAL option that backs up only those data pages which have changed since the last Full Backup or ARCHIVE FULL.
LISTLOG	Lists all the database LUW statements from the active log and log archives for the table(s) selected. The log output is written to working files on disk. This function can only be used after executing RELOAD with option RECOVERY=YES.

RELOAD	Loads from 1 to 90 tables from either a Data Restore translated database archive, BACKUP, or UNLOAD. If option RECOVERY=YES is chosen, RELOAD collects LUW log activity for the indicated table(s). Log forward recovery of table LUWs can later be executed using the LISTLOG and APPLYLOG functions.
RESTORE	Restores a complete database from a Data Restore BACKUP.
TRANSLATE	Converts a database archive to a Data Restore BACKUP format. Several working files are created. These files must be available in order to process RESTORE and UNLOAD commands from the translated archive.
UNLOAD	Selectively unloads one or more DBSPACES in order to retrieve table data, definitions and authorizations.
SHOWDBS	The Data Restore SHOWDBS command will retrieve information about a database's HEADER, DATA and INDEX pages for ALL acquired dbspaces. The SHOWDBS command can be executed when the database is online or offline. SHOWDBS can be executed on the CMS command line by: SQM dbmach SHOWDBS.

Important Considerations

There are some important differences between the way Control Center manages Data Restore and database server functions.

1. A password file, discussed in more detail in “Password File Setup for the Data Restore machine” on page 504, is required.
2. Control Center will perform all the minidisk links and accesses necessary to manage Data Restore functions. This lets Control Center manage disk access for multiple databases. Any disks Control Center accessed during Data Restore functions will be detached when the functions are completed.
3. Disk file definitions for Data Restore functions in the UNLTAPES and TAPES files must include the actual disk address (as defined in the VM directory) and not the access cuu of the file. Control Center will link to the actual address using the first free cuu starting with cuu 120.

   Filemodes for Data Restore disk files are used as placeholders only. Control Center will access required disks using the first free filemode.
4. A separate SDRESTOR DIRECTRY file is required on the Control Center code disk. The SDRESTOR DIRECTRY will be used in menu operations to provide a map between a database and the Data Restore machine that will perform the Data Restore functions. See “Setup Step 8: Create the SDRESTOR DIRECTRY” on page 98 for information about creating this file.
5. You should set your PREMOUNT_DRTAPE option in the database PARMS file to "N", unless your tape management system does not include the DMSTVI tape exit. The PREMOUNT_DRTAPE option, is described in more detail in “Data Restore Parameters in the Database PARMS File” on page 223. The DMSTVI tape exit will automatically perform all tape mounts based on VOLID information contained in the FILEDEF and LABELDEF performed prior to executing the Data Restore functions.

With typical database archives, mounting the tape before the archive begins, improves availability because the database can take a checkpoint as soon as the archive starts, instead of waiting for completion of the tape mount.

In both the server and Data Restore functions, if the tape manager is set to NONE or blank, or if the DMSTVI MODULE is not present, the tapes will be premounted automatically.

6. All Data Restore functions will be executed on the Data Restore machine. This means that the DASD and tapes will be owned by the Data Restore machine. The reasons for this are to:
   a. Segregate Data Restore from the database manager.
   b. Allow ownership of tapes to be held by the Data Restore machine. This will facilitate tape mounting for RESTORE operations.
   c. Allow table RELOADs from BACKUP tapes. RELOAD tables requires that the database server be up and running, so the command cannot be executed on the database manager. This will allow RELOAD to be performed from either BACKUPs or UNLOADs without the need for additional tape authority. Executing a RELOAD RECOVERY will require tape authority for the database log archive tapes or read access to the log archive disk file.
7. Any changes made to Data Restore parameters in a given database PARMS file WILL be used during the next Data Restore function, regardless of the status of the database. This is because Data Restore parameters in the database PARMS file are re-initialized on the Data Restore machine prior to each Data Restore function.
8. A secondary status called "**DBSTAT2**" will be used to track Data Restore machine status changes when two tapes must be managed concurrently, such as during a DUAL BACKUP. Both the DBSTATUS and DBSTAT2 GLOBALV values will be updated as required.
9. Required block size and RECFM FILEDEF options for Data Restore work files are pre-determined and cannot be modified by the user.

Disk and Tape Passwords

A password file (LINKPWDS) is required to support to provide the database server password for 'SQLDBA' to Data Restore and, if necessary, to support the Data Restore machine's links to the database's directory, log, data, and archive disks. See “Password File Setup for the Data Restore machine” on page 504 for information about creating this file.

Architecture

Command and control of the Data Restore machine activities will be accomplished through a combination of the database and support machine methodologies.

For automated operations, the Data Restore machine runs as a disconnected machine and uses the Single Console Image Facility (SCIF) to give control of its console to Control Center. This will enable unattended operation of Data Restore functions under the supervision of a Control Center service machine.

Data Restore functions, such as UNLOAD, that are initiated by users, or initiated by the database manager (SQLEND UARCHIVE using BACKUP), are first processed

by Control Center. It checks the authorization of the user to execute the function, changes the status of the Data Restore machine to indicate the database involved and the Data Restore function that is to be executed. It will then use SCIF to execute the appropriate program on the Data Restore machine.

All Data Restore functions will execute on the Data Restore machine. Only one database can be processed at a time on a particular Data Restore machine.

Unlike a database, the Data Restore code is not running all of the time. With the SCIF interface, Control Center knows when Data Restore code is executed. The program executing on the Data Restore machine will perform the necessary FILEDEFs and LABELDEFs, SYSIN file creation, and any other setup required. The Data Restore function executes while Control Center monitors the Data Restore machine's console.

If necessary, Control Center responds to any Data Restore messages. When the Data Restore function is complete, the result will be indicated to Control Center, and appropriate notifications made. After the Data Restore function is finished, its status will return to **NORMAL**, and the Data Restore machine will be free to process the next function.

As with databases, all control files and logging activity associated with the Data Restore machine will be updated and maintained on the Control Center service machine.

Message and Command Processing

The SCIF interface allows Control Center to receive console messages, and place commands and responses directly on Data Restore's console as if it were logged on to the Data Restore machine.

The use of SCIF lets Control Center respond to messages that are produced on the Data Restore machine console while the Data Restore modules are executing.

Much like a database, messages produced by the Data Restore machine, such as TAPE ATTACHED, error and action messages, and other information will be processed asynchronously by Control Center. Each message will be processed in turn. If the incoming message matches one of the messages in the DATABASE MESSAGES file, the appropriate action will be taken.

If a response is required by the Data Restore machine, the SQMCPSND EXEC (using CP SEND) is used to put commands directly on the Data Restore machine console. SQMCPSND exits as soon as the command or response is successfully sent and does not wait until the issued command is completed. Control Center can move on to the next message or action to be processed, thereby allowing quick asynchronous operation.

SCIF messages contain the userid of the sender. This userid will indicate which Data Restore machine is involved. To track which database the Data Restore function is being executed against, the GLOBALV group, DBSTATUS for that Data Restore machine, will be updated to include the database machine name involved and information about the functions being processed, each time a database job is executed on the Data Restore machine.

The DBSTATUS for the Data Restore machine, described in more detail in “Status Tracking” on page 261, is separate from the DBSTATUS of the database manager. By using this design, Control Center can monitor activities of both the database and Data Restore machines.

Communicating with Data Restore

During menu operations, the SDRESTOR DIRECTRY file will be used to associate one or more Data Restore machines with the proper Control Center service machines and databases. This lets the user issue commands against local or remote Data Restore machines. Control Center uses SDRESTOR DIRECTRY to determine which Data Restore machine is associated with the database, and which service machine is managing that Data Restore machine. The command will then be routed to the proper Data Restore machine's console.

While the Data Restore functions are executing, parameters in the database PARMS file, and the Data Restore machine DRPARMS file, will be used to associate a Data Restore machine with a database and the service machine.

The following association between the Data Restore machine, database machine and Control Center must be setup in order for the proper automated control and management of Data Restore functions to occur:

- A Data Restore machine can only be SCIFed to one Control Center service machine.
- A database can only be associated with one Data Restore machine.
- A Data Restore machine must be managed by the same Control Center service machine as the database.
- You can have more than one database manager associated with a Data Restore machine. However, that Data Restore machine can only perform Data Restore functions for one database at a time.
- You can have more than one Data Restore machine associated with a server. In this way, you can have Data Restore functions executing against different databases at the same time.

Interfaces

If the Data Restore machine is not running disconnected, the SCIF connection will be broken. Control Center will not be able to issue commands on the Data Restore machine's console or respond to appropriate Data Restore functions.

If logged on to the Data Restore machine, a user can execute Data Restore functions directly without the intervention of the Control Center service machine.

Status Tracking

In order to track which database and Data Restore function is currently executing, a set of 'DBSTATUS' and 'SQMODE' GLOBALV values for each Data Restore machine will be used to control the flow of activity and anticipate responses. This is similar to the current database DBSTATUS and SQMODE strategy. Refer to Chapter 17, “Database Status Tool” on page 203.

For the Data Restore machine's DBSTATUS, the first word will indicate what database manager is involved. If the Data Restore machine is not processing any Data Restore functions, the DBSTATUS and the SQMODE will be 'NORMAL'.

When the DBSTATUS for a Data Restore machine is not 'NORMAL', users will not be allowed to initiate new Data Restore functions.

Secondary status called '**DBSTAT2**' will be used to track Data Restore machine status changes when two tapes must be managed concurrently, such as during a DUAL BACKUP. Both the DBSTATUS and DBSTAT2 GLOBALV values will be updated as required.

Installation Considerations

Using Data Restore functions with Control Center depends upon:

1. Installing the Data Restore Feature as described in the *DB2 Server for VSE & VM Data Restore*.
2. Updating the required control files, and
3. Executing of Control Center's installation procedure for Data Restore support (Chapter 9, “Data Restore Setup with Control Center” on page 93).

Control File Setup

Implementation of Data Restore requires modification or editing of these control files:

- SDRESTOR DIRECTRY
- DRPARMS File (Data Restore machine's Parms, similar to database Parms file)
- DATABASE PARMS
- SQLMSTR PROFILE
- DATABASE TAPES
- DATABASE UNLTAPES (if Data Restore unloads are expected)

Resources

Data Restore functions either require, or have the option of, output to DASD. Control Center will use the A disk of the Data Restore machine for SYSIN, SYSPRINT, and control files. Additional minidisks will be required to store the output from Data Restore functions.

Considerations that affect the size and number of minidisks needed include:

- Whether BACKUPS will be performed to DASD,
- The size of the database or DBSPACES,
- The frequency of BACKUPs to DASD (You can have data from more than one BACKUP on a minidisk, however, you may decide to have each BACKUP go to a different minidisk on a different DASD pack.), and
- The quantity of UNLOAD data output you wish to keep.

Invocation

You can invoke Data Restore functions with Control Center through:

- Panel interface or
- Automatic invocation: The command for the BACKUP function is automatically issued when an explicit SQLEND UARCHIVE is executed, and the "Drmstr_enabled" parameter in the database PARMS file is set to 'Y'. All BACKUP options are specified in the database PARMS file.

```
 mm/dd/yyyy                 CONTROL CENTER                       hh:mm:ss
*------------------------------ Data Restore Menu -------------------------*
| Option ===>                                                 CTRLID: MSTRSRV1 |
|    Database ===> SQLDBA                                     NODE:   VMSYSTM1 |
|                                                             DRMACH: DREST41  |
|                                                                              |
|     T  TRANSLATE ARCHIVE          TRANSLATE archive into BACKUP format       |
|     U  UNLOAD DBSPACES            UNLOAD one or more dbspaces                |
|     R  RELOAD TABLES              RELOAD one or more tables                  |
|     LL LISTLOG                    LISTLOG selection panel                    |
|     AL APPLYLOG                   APPLYLOG selection panel                   |
|     VJ VIEW JOB SCHEDULE          VIEW database job schedule                 |
|     S  VIEW DRMACH STATUS         VIEW Data Restore Machine status           |
|     SR RESET DRMACH STATUS        RESET Data Restore Machine status          |
|     D  SHOWDBS                    Generate report about all dbspaces         |
|          View Tapes   Edit Tapes    View History    View Log                 |
|  BACKUP      BT          BM             BH             BL                    |
|  UNLOAD      UT          UM             UH             UL                    |
|  TRANSLATE   TT          TM             TH             TL                    |
|                                                                              |
|           Enter OPTION, select DATABASE, press ENTER to process              |
| The requested command was issued                                             |
*------------------------------------------------------------------SDRMAIN-----*
   PF:   1 Help    3 End
```

Figure 115. Data Restore Primary Menu

Authorization

A user must be authorized in the database PARMS file or the SQLMSTR Profile to execute Data Restore functions. The database PARMS file includes authorizations that affect the database. The SQLMSTR PROFILE controls authorities for ADMINISTRATIVE commands that will be executed on the Data Restore machine.

To issue Data Restore functions, the user must, at a minimum, be listed as a database administrator in the database PARMS file. This file is also be used by the Data Restore machine.

If the user is listed as a Control Center Administrator (Level 5), in the Control Center's *SQLMSTR Profile*, all commands (including Data Restore functions) can be issued by that user against all databases controlled by that Control Center service machine.

For more details, see “Maintaining the SQLMSTR PROFILE File” on page 118.

Performance

Performance improvements from using Data Restore will be dependent upon the size of the database objects being processed and your overall archive/recovery strategy.

Chapter 21. Data Restore BACKUP

Overview

Data Restore BACKUP is used to perform a user archive of the database. It accesses, reads and unloads the database directory and data to either disk, tape, or both. The BACKUP output produced can later be used to either recover the whole database or selected tables.

When enabled, the Data Restore BACKUP will be performed automatically in conjunction with an explicit SQLEND user archive of the database.

If only a small part of data is modified daily, you can reduce the time required for a user archive by using the Incremental Backup function. Using the Incremental Backup function, only datapages which have changed since the last Full Backup are recorded.

Having Data Restore enabled to Control Center ("Drmstr-enabled" = "Y" in the database PARMS file), does NOT require that you perform Data Restore backups. For explicit archives, you can always select to execute a normal DB/2 archive. If the "Uarchive-enabled" database parameter value is "N", then Data Restore BACKUP will not be executed.

To use BACKUP with Control Center, it will be invoked as a SQLEND UARCHIVE. This will ensure that the database log history is updated, and the ARCHHIST files are properly updated for recovery or RESTORE operations.

Terms and Definitions

Term	Definition
BACKUP	A USER ARCHIVE using Data Restore BACKUP that backs up the whole database. A BACKUP can not be used as a reference for a subsequent Incremental Backup.
Incremental Backup	A USER ARCHIVE using Data Restore BACKUP that backs up only those data pages which have changed since the last Full Backup or ARCHIVE FULL. If no Full Backup exists, Data Restore will not produce an Incremental Backup. To RELOAD or RESTORE from an Incremental Backup, it is necessary to provide the associated Full Backup.
Full Backup	A USER ARCHIVE using Data Restore BACKUP that backs up the whole database and can be used as a reference for subsequent Incremental Backup.
Reference Backup	The last Full Backup taken prior the Incremental Backup. When processing an Incremental Backup for RELOAD or RESTORE, the required pages are contained in the Incremental Backup and its reference backup. A reference backup for an

	Incremental Backup is the last Full Backup taken before the Incremental Backup.
Incremental Reference	The last Full Backup, translated ARCHIVE FULL or ARCHIVE FULL taken prior to the Incremental Backup.
ARCHIVE FULL	A DB2 Server for VM archive of the whole database that can be be translated into a reference backup for an Incremental Backup.
ARCHIVE	A DB2 Server for VM archive of the whole database that can not be used as a reference BACKUP.

Implicit Archives

Implicit archives will be initiated automatically by the database when the ARCHPCT of the log file is reached. Logmode A will cause an implicit DB/2 archive, and logmode L will cause an implicit log archive. Since this is initiated by the database, if logmode A is used, Control Center will not be able to request a SQLEND UARCHIVE using BACKUP.

There are several ways that you can prevent an implicit archive from occurring. One way would be to schedule a log archive (or full archive for logmode A) at intervals that will normally prevent the ARCHPCT from being reached. Another way would be to schedule a database monitor routine (using SHOW LOG) that would periodically check the log percent, and explicitly initiate an archive if the log was near the ARCHPCT value.

DASD and TAPES

BACKUP is executed on the Data Restore machine This means that the DASD and TAPES will be owned by the Data Restore and not the database manager. The reason for this is to:

- Segregate the Data Restore machine from the database manager.
- Allow ownership of tapes to be held by the Data Restore machine. This will facilitate tape mounting for RESTORE operations.
- Allow table RELOADs from BACKUP tapes. RELOAD requires that the database server be up and running, so the command cannot be executed on the database manager. This will allow RELOAD to be performed from either BACKUPs or UNLOADs without the need for additional tape authority. Executing a RELOAD RECOVERY will require tape authority for the database log archive tapes or read access to the log archive disk file.

Tape Series and History Files

The same tape series, as recorded by the "Archive_series" value in the database PARMS file, will be used for both archive and BACKUP. This is done in order to keep the BACKUP and archive events synchronized.

The ARCHHIST file records database archive and user archive events. It also holds all backup and archive information. Recovery sets are determined by information in the ARCHHIST file. BACKUP and TRANSLATE timestamps are appended to the corresponding ARCHHIST record so they can be mapped to the related ARCHHIST and TRANSHIST events. Keeping the series synchronized ensures that a BACKUP

or TRANSLATE event will not be purged from the log files while there still is a corresponding archive event.

Unlike a log archive to disk, the filetype of the archive file is not changed by Data Restore during the BACKUP process. You must be sure to use unique FILETYPES for each BACKUP minidisk file that is listed in the TAPES file.

BACKUP Process

After the log archive (if logmode=L) completes, the BACKUP function will begin. Based upon input to the database PARMS file, and the database TAPES file, the proper SYSIN file will be created, media FILEDEFed and LABELDEFed, and tapes mounted.

After the BACKUP completes, Control Center will be notified and the database ARCHHIST file will be updated with the proper status.

If the BACKUP completes successfully, the tapes used during this process, along with the timestamp of the BACKUP, will be added to the ARCHHIST file. The archive complete record in the database ARCHHIST file will indicate the type of backup (FULL, Incremental, Dual) that was performed.

If the BACKUP fails, a failure record will be placed in the ARCHHIST file. Any log archives performed prior to BACKUP will still be valid.

After the BACKUP is finished, the database is restarted.

When using BACKUP, the entries in the database TAPES file will track the BACKUP tapes to be used, and the database ARCHHIST file will record the tapes or files that are used.

Incremental Backup Process

Execution of an Incremental Backup is treated like a normal user archive by DB2 Server for VM and the process used by Control Center to manage the Incremental Backup will look very similar to a normal BACKUP. However, the Incremental Backup will only backup those database data pages that have changes since the last Full Backup or ARCHIVE FULL.

Consider the following situation: a Full Backup is executed each weekend and Incremental Backup every day. The Incremental Backup produced on Monday will only contain pages modified on Monday. The Incremental Backup produced on Tuesday will contain pages modified on Monday and Tuesday (all pages modified since the last Full Backup), and so on.

Before an Incremental Backup for a database can be executed through Control Center the user must enable Incremental Backup (Enable-incbackup = Y) in the database PARMS file for that database and add Incremental Backup tapes or files to the database TAPES file. After updating the TAPES and PARMS files, the user must also perform a Full Backup or ARCHIVE FULL before Data Restore will allow an Incremental Backup.

An Incremental Backup is a USER ARCHIVE and is recorded in the database history area as a USER ARCHIVE. Databases using logmode = L will perform a log archive prior to the Incremental Backup.

When "Incbackup-enabled" = "Y" in the Control Center database PARMS (parameters) file, Control Center will only execute a BACKUP with the FULL or INCREMENTAL option. Full Backup is the default backup type. This will allow Control Center to keep a Full Backup and its Incremental Backups together in the same tape series. When "Incbackup-enabled" = "N", Control Center will not execute an Incremental Backup.

An ARCHIVE or BACKUP using the FULL option will backup **all** the pages of the database and will be used as the reference backup for subsequent Incremental Backups. The reference backup is not processed when executing Incremental Backup, but is required for the RELOAD and RESTORE functions.

A Full Backup or ARCHIVE FULL will place a timestamp in the database directory and also resets the modified_pages indicators. Only the last Full Backup or ARCHIVE FULL timestamp is kept in the database directory.

When a Full Backup or ARCHIVE FULL completes successfully, Control Center will record the timestamp in the database ARCHHIST file. When an Incremental Backup completes successfully, its timestamp and the timestamp of its reference backup will be included in the database ARCHHIST entry. This will allow Control Center to maintain and manage the Incremental Backup and its associated reference backup.

An Incremental Backup and its reference backup are equal to one usable BACKUP. If a reference backup has been overwritten, then the Incremental Backup associated with that reference backup can **not** be used for a RESTORE or RELOAD. To RESTORE or RELOAD from an Incremental Backup, Data Restore will first process all the changed pages recorded in the Incremental Backup and then process unchanged pages from the reference backup. Control Center will automatically manage all necessary FILEDEFs and tape mounts.

For implicit ARCHIVES, logmode **A**, the database server will perform an ARCHIVE FULL if there has been any Full Backup or ARCHIVE FULL performed previously. During archives, the database will indicate if an ARCHIVE FULL has executed.

An ARCHIVE FULL will have to be translated by Data Restore before it can be used to RELOAD or RESTORE an Incremental Backup. Data Restore will indicate if an ARCHIVE or ARCHIVE FULL is being translated.

Tape Rotation with Incremental Backup

Incremental Backup has its own "Incbackup-series" in the database PARMS file to control which set of tapes will be used for the next Incremental Backup.

The Incbackup-series in the database PARMS file indicates the series and subseries in the database TAPES file that will be used to supply tapes for the next Incremental Backup. The Incbackup-series is presented in the form "**series.subseries**" where:

series The series indicates the archive-series of the last archive or BACKUP. The series portion of the Incbackup-series will be the same as the Logarch-series and is incremented after any archive or BACKUP completes. The series is incremented and the subseries reset to "01" each time any normal BACKUP, Full Backup, normal archive or ARCHIVE FULL completes.

subseries The subseries indicates which set of Incremental Backup tapes to use within the series. The subseries will be a number between "01" and "99". The first subseries after an archive or backup will always be "01". For example, Incremental Backup series 100.01 indicates that the first set of Incremental Backup tapes (01INCBK) in series 100 are to be used.

The subseries is incremented **only** after an Incremental Backup successfully completes.

The Archive-series, Logarch-series and Trace-series are **not** incremented when an Incremental Backup completes.

The Incbackup-series will be maintained in the database PARMS file even when "Incbackup-enabled" = "N".

After an Incremental Backup completes, the subseries will be incremented even though there may not be an "INCBK" entry in the database tapes file for that Incbackup-series.

The user must setup the database TAPES file so that there will be enough entries of each tape type to accommodate the expected backup/archive workload. However, there will be times when a user has performed more than the usual number of Incremental Backups, and there may not be anymore "INCBK" entries in the database TAPES file to satisfy a regularly scheduled Incremental Backup. Control Center provides several options to ensure that archiving can continue.

If another Incremental Backup is requested when there is no "INCBK" entry for the current Incbackup-series, then one of these actions will occur:

1. If you are using SCRATCH tapes for your Incremental Backup, then Control Center will automatically add a new "INCBK" SCRATCH entry in the database TAPES file for the current Incbackup-series. The Incremental Backup process will continue.
2. If the "Auto-full" option in the database PARMS file is set to "Y" (Yes), then Control Center will automatically switch the Incremental Backup to a Full Backup. The "Auto-full" parameter will be ignored if the "AUTOFULL" or "NOAUTOFULL" command option is used.
3. If you have indicated the "AUTOFULL" command option when setting up the Incremental Backup through the menu or on the command line, then Control Center will switch the Incremental Backup to a Full Backup regardless of the "Auto-full" value in the database PARMS file.

   If you have indicated the "NOAUTOFULL" command option then Control Center will **not** switch the Incremental Backup to a Full Backup regardless of the "Auto-full" value in the database PARMS file. The Incremental Backup will be cancelled.
4. If none of the above options are chosen, the Incremental Backup will be cancelled.

If the Incremental Backup is cancelled, you can use the Tape Maintenance menu to add the new "INCBK" entry and then request another Incremental Backup.

The user must understand the implications of switching an Incremental Backup to a Full Backup. A Full Backup may take considerably longer than an Incremental

Backup and can affect the execution of other scheduled database maintenance activity.

There may be times when switching or not switching makes more sense. The AUTOFULL/NOAUTOFULL command option offers the user the flexibility to control the Incremental Backup without changing the overall Incremental Backup strategy. In order to keep the Incremental Backup tapes together with its reference backup, Control Center will increment the Incbackup-series whenever a BACKUP or archive completes.

An incremental backup is a User Archive and when the database is using logmode *L*, log archives will be initiated by the database.

Number of LOG ARCHIVE entries in a series

There must be enough log archive entries in each Logarch-series to handle all log archives between Full Backup or ARCHIVE FULL in the Incbackup-series. If the database is logmode *L*, then a log archive is performed before each USER ARCHIVE (BACKUP or Incremental Backup).

Before You Begin

Before you begin using BACKUP, consider or perform the following:

- When and if to use SQLEND UARCHIVE with BACKUP
- Whether to use Incremental Backup for user archives
- Review Chapter 15, “Tape Management Tool” on page 185
- The logmode to be used by each of your databases
- Single or Dual BACKUP
- Medium type (tape or disk) to be used for single or dual BACKUP
- Whether SCRATCH tapes should be used
- The number of tapes required for a BACKUP
- Use of specific pre-allocated tapes or scratch tapes and tape management product used (VMTAPE, EPIC, DYNAM/T or other)

Files Used to Manage BACKUPS

The four files below provide necessary information to Control Center in managing BACKUP activity.

- SQLMSTR CONTROL
- SQMSTAPE EXEC
- Database PARMS
- Database TAPES

SQLMSTR CONTROL

This file contains information that describes the interfaces to tape management systems like VMTAPE, EPIC and DYNAM/T . It is created during the service machine's installation and is located on Control Center's A-disk. Refer to Chapter 4, “Installing the Service Machine” on page 37 and Chapter 11, “Managing the Environment” on page 113 for specific information regarding this file.

SQMSTAPE EXEC

This exec is used by the Data Restore machine to issue tape mount requests, if required. It exists on the service machine's code disk and can be modified to accommodate unsupported tape management systems. The SQMSTAPE $EXEC contains the original product code and should not be modified.

Database PARMS File

Based on certain parameters chosen in the database PARMS file, BACKUP will automatically be initiated each time a SQLEND UARCHIVE is performed.

The database PARMS file contains Data Restore information that will be used during the BACKUP process. For specific details regarding this file, and the Data Restore parameters, refer to “About the Database Parameters Tool” on page 213.

Changing the Database PARMS File

It is important to note that any changes made to Data Restore parameters in a given database PARMS file WILL be used during the next Data Restore function regardless of the status of the database. This is because Data Restore parameters in the database PARMS file are re-initialized on the Data Restore machine prior to each Data Restore function.

Database TAPES File

The database TAPES file contains the tapes or disks to be used during BACKUP. It must be defined before performing a BACKUP. Similar in structure to the database TAPES file, it is used prior to executing BACKUP to correctly establish required CMS FILEDEF and LABELDEF information that will be used during the operation. The current archive series (recorded in the database PARMS file), will be used to determine the BACKUP series.

Entries for the primary backup (ARCHIV) are indicated in the TAPES file by using a type of BACKUP. Entries for the dual backup (ARCHIV2) are indicated by using a type of BACKUP2. If the Dual_backup value in the database PARMS file is "N", then tape/file information for BACKUP2 will be ignored. Even if you do not anticipate using dual BACKUPS, it might be helpful to still include an entry for BACKUP2. If you only perform user archives, you still must have an ARCHIVE entry in the database TAPES file for each series.

Remember, if you use logmode A, and the log fills up, the archive will be automatically handled by Control Center based on the tape file information listed for that series. You do NOT need to always perform a SQLEND UARCHIVE because you have enabled Data Restore. For explicit (user initiated) archives, you can always select which kind of archive to execute (user or otherwise).

If the BACKUP or BACKUP2 (dual backup) media is disk, then the TAPES file entry must include the filename, filetype and link address of the disk where the file is to

be written. The filemode value is required as a placeholder only. The filemode and access cuu (virtual address) will be determined prior to executing the FILEDEFS for the function.

```
SERIES TYPE    DATE   TIME     STATUS   FILENAME  FILETYPE  FM   CUU
 100   BACKUP  97021  19:00:56 FILLED   VB3025
 100   BACKUP2 97021  19:00:56 FILLED   BACK100   DB2VM510  *    400
```

Figure 116. Example Database TAPES file for DUAL backup to TAPE and DISK

Figure 116 above shows BACKUP series 100 using dual backups where both tape and disk are specified. In this case, BACKUP, the primary backup, will use tapes VB3024 and VB3025, as needed. If an additional tape is needed, a SCRATCH tape will be acquired. The VOLID of the SCRATCH tape will be added to the TAPES file. BACKUP2 will use CMS file BACK100 DB2VM510, located on minidisk address 400, for the dual backup.

Changing BACKUP Output information

Changes to BACKUP output media types (disk to tape or tape to disk) or volid/file information will take effect **immediately** and **should not** be changed during execution of the BACKUP. Changes to database TAPES file while the function is running could cause the process to *fail*.

BACKUP Media Types (Tape and Disk)

BACKUP to tape and disk are fully supported. This information is specified in the database PARMS file. You can designate that your primary backup or dual backup be performed to tape and disk. See the example database PARMS file below in Figure 117 which shows dual BACKUP using both tape and disk. The only restriction is that all primary backups are of the same media type, and all your DUAL backups are of the same media type (tape or disk).

```
:Archive_media.TAPE
:Uarchive_enabled.Y
:Drmstr_enabled.Y
:Dual_backup.Y
:Backup2_media.DISK
:Scratch_tape_option.N
```

Figure 117. Example DUAL BACKUP to TAPE and DISK Database Parameters

About BACKUPS to Disk

To BACKUP using disk, the database TAPES file and the database PARMS file must be defined properly. The PARMS file will have disk for **Archive_media** and **Backup2_media** (if dual is used), rather than tape.

Note: You should not specify a virtual address of less than 200 since addresses between 180 and 190 are reserved for tape usage.

All media entries in the TAPES file must match the media format in the PARMS file. The volid for disk archiving should be expressed as a filename, filetype, filemode, and the link address (as defined in the VM directory).

BACKUP Disks Cannot Span Minidisks: A single BACKUP to disk cannot span multiple minidisks. Therefore, a full BACKUP must fit on a single minidisk. BACKUPS to tape may span multiple tapes.

BACKUP Disks Should not be Linked and Accessed: Another requirement of BACKUP disks is that each output minidisk should not be linked and accessed by the Data Restore machine. Control Center will handle all required link and accessing. The link address listed in the TAPES file will be linked to using the next available virtual cuu starting with 120 and the next free filemode. This is to prevent the Data Restore machine from running out of filemodes or cuus when it must manage Data Restore functions for several databases. Control Center will check the contents of the database SQLFDEF file to ensure that the cuu selected does not match an address used by the database. Data Restore will detach any cuus that match a database"s minidisk addresses as part of its operation. Control Center will also detach any duplicate links that the Data Restore machine has made to the same address.

BACKUPS and Tape File Rotation: In order to support the rotation from one tape series to the next, BACKUPS can only be executed by issuing a SQLEND UARCHIVE. The BACKUPS will always use the same series as the current archive series. This ensures correct mapping of a user ARCHIVE event to a BACKUP event.

BACKUP Disk Cleanup: Control Center will use the unique filename provided by the user in the TAPES file for each backup to disk. Therefore, each BACKUP file will be overwritten when the series is used again.

If you change the name of the BACKUP files, you should manually perform disk cleanup of any old BACKUP files.

At the completion of a full SQLEND UARCHIVE, Control Center will automatically switch to the next series of tapes within the TAPES file for the next archive or BACKUP, to prevent the next archive from writing over the previous archive. Control Center will use the same series as the archive to process the BACKUP.

User Archiving Without Data Restore

User archiving without Data Restore can still be processed by changing the **Drmstr_enabled** parameter to "N" in the database PARMS file and keeping the **Uarchive_enabled** parameter set to "Y". However, if you do not use Data Restore, then you should have a user archive process in place as defined in Appendix F, “User Archiving” on page 515.

How to Invoke BACKUP

Once all the control files have been defined to Control Center and the Data Restore machine, BACKUP can be invoked either through command mode execution, or through Control Center's menu interface.

Since BACKUP is a USER ARCHIVE function, select Option **A** from the main menu of Control Center to reach the Archiving Tool. Then select Option **I**. Figure 118 on page 274 is now displayed. Options on this menu are available to the BACKUP administrator for selection.

```
 mm/dd/yyyy                    CONTROL CENTER                        hh:mm:ss
*---------------------------- Initiate Archive -----------------------------*
| Option ===>                                                CTRLID: MSTRSRV1 |
|    Database ===> SQLDBA                                    NODE:   VMSYSTM1 |
|                                                                             |
|    ********************** ARCHIVE COMMANDS **********************         |
|    SA SQLEND ARCHIVE parms         Full archive, data base down             |
|    SL SQLEND LARCHIVE parms        Log archive, data base down              |
|    SU SQLEND UARCHIVE parms        User archive, data base down             |
|    A  ARCHIVE parms                Full archive, data base up               |
|    L  LARCHIVE                     Log archive, data base up                |
|                                                                             |
|    ***************** Data Restore BACKUP COMMANDS *****************         |
|    BU BACKUP parms                 Data Restore Backup                      |
|    BI BACKUP INCREMENTAL parms     Data Restore Incremental Backup          |
|                                                                             |
|    Valid SQLEND Parms:    DVERIFY, TRCPURGE                                 |
|    Valid Incremental Backup Parms: AUTOfull NOAUTOfull                      |
|                                                                             |
|             Enter OPTION and PARMS, press ENTER to Process                  |
|                                                                             |
*-----------------------------------------------------------------SQMAR10------*
 1 Help    3 End (Cancel)
```

Figure 118. Database Archiving Initiation Panel

Parameter Definition

BU Will perform a User Archive using Data Restore BACKUP. When Incbackup-enabled = "Y" (Yes) in the database PARMS file, "BU" will **always** execute a Full Backup. A Full Backup can be used as the reference backup for an Incremental Backup. When Incbackup-enabled = "N" (No) in the database PARMS file, "BU" will **only** execute a BACKUP. A normal BACKUP can not be used as a reference backup for an Incremental Backup.

BI Will perform a User Archive using Data Restore Incremental Backup.

The AUTOFULL and NOAUTOFULL options can be used with Incremental Backup to control how Control Center will respond if there are no INCBK tapes for the current incbackup-series in the database TAPES file.

Note: If you are using SCRATCH tapes for your Incremental Backup, the AUTOFULL/NOAUTOFULL option is not necessary. Control Center will automatically add a new INCBK SCRATCH entry in the database TAPES file for the current Incbackup-series and The Incremental Backup process will continue.

If you have indicated the AUTOFULL command option then Control Center will switch an Incremental Backup to a Full Backup if there are no INCBK tapes for the current incbackup-series in the database TAPES file. With AUTOFULL, Control Center will switch the Incremental Backup to a Full Backup regardless of the Auto-full value in the database PARMS file.

If you have indicated the NOAUTOFULL command option then Control Center will **not** switch the Incremental Backup to an Full Backup

regardless of the Auto-full value in the database PARMS file. The Incremental Backup will be cancelled.

The user must understand the implications of switching an Incremental Backup to a Full Backup. A Full Backup may take considerably longer than an Incremental Backup and can affect the execution of other scheduled database maintenance activity.

There may be times when switching or not switching makes more sense. The AUTOFULL/NOAUTOFULL command option offers the user the flexibility to control the Incremental Backup without changing the overall Incremental Backup strategy.

Note: Executing the BACKUP command is the same as executing the User archive command when Data Restore is enabled to Control Center.

After BACKUP is initiated using Control Center, either through command mode or panel interface, no further action will be required by the user.

Authorization

You must have database or Control Center administrator authority to execute BACKUP. Those persons identified as Control Center administrators on other databases, or persons with Control Center user or operator authority cannot invoke BACKUP.

Cancel a Data Restore BACKUP

To cancel a Data Restore BACKUP, select Option **C** from the Archiving Menu, or use the command mode interface to terminate the Data Restore process. For BACKUPS, the cancel routine process will end with a user archive failed message sent to the database. Control Center will restart the database indicating the BACKUP has failed.

Chapter 22. Data Restore RESTORE

Overview

RESTORE is used to recover an entire database from either a Data Restore BACKUP, Incremental Backup, or a translated database archive. Database archives MUST be translated prior to executing the RESTORE. If DUAL BACKUPs are used, the user has the choice of restoring from either the primary backup or dual backup. RESTORE is executed as part of a normal USER RECOVERY.

To RESTORE from an Incremental Backup, Data Restore first recovers the Incremental Backup pages and then Full Backup pages.

Considerations When Using Data RESTORE

These steps must be taken to prepare for Data RESTORE recovery:

- The Uarchive_enabled and the Drmstr_enabled parameters in the database PARMS file must be set to **Yes**.
- The restore sets displayed in the Restore Set Report will include information about any BACKUP or TRANSLATE that occurred.
- The "Use Data Restore" option from the Restore Set Selection menu must be set to "Y" (YES).
- The "Backup Type" option from the Restore Set Selection menu must either indicate "BACKUP" for restore from a primary BACKUP or translated archive, or "BACKUP2" if restore from the dual BACKUP is desired.
- Recovery using Data Restore is a USER ARCHIVE recovery.
- After you select what restore set you wish to recover, you will be presented with a temporary file which will include the control files used for the database recovery (database RECOVCTL) and the Data Restore RESTORE (database DRESTCTL). The database DRESTCTL file includes entries for both the Incremental Backup and the Full Backup. Figure 119 is an example of a DRESTCTL file for Incremental Backup.

```
RECOVERY LOGMODE = Y DRBKUPI FROM BACKUP @ 11/13/97 11:52:13
#1 DR  BACKUP 1997-11-13 11.56.07 100 N VP0135
#2 DR  REFBK  1997-11-13 11.56.07 100 N VP0134
END RESTORE SET
```

Figure 119. RESTORE Control File (DRESTCTL) for Incremental Backup

```
--------------------------- DRESTCTL FILE -----------------------------

RECOVERY LOGMODE = L FROM: DUALINCBACKUP  @06-16-98 11:09:32 INCREF 06-15-98 13:35:20
#1 DR BACKUP      1998-06-16 11.15.39 300.01 N PM1160
#2 DR REFBACKUP   1998-06-15 13.43.06 300    N VM8451
END RESTORE SET

--------------------------- RECOVCTL FILE -----------------------------

RECOVERY LOGMODE = L FROM: DUALINCBACKUP  @06-16-98 11:09:32 INCREF 06-15-98 13:35:20
#1 USER ARCHIVE 1998-06-16 11.15.39 300.01 N USER BACKUP DR_BACKUP 06-16-98 11:09:32
#2 LOG Archive ACTIVE LOG - TO BE ARCHIVED
END RESTORE SET
```

Figure 120. RESTORE Control File (DRESTCTL) for Incremental Backup When Backup is a Full Backup

How to Invoke RESTORE

You will use the Control Center panel interface to initiate a RESTORE. Various recovery options are shown in the figure which follows, including the ability to schedule a recovery, or initiate one immediately. This panel is reached by selecting option **R** (Database Recovery) from the Control Center Main Menu.

```
mm/dd/yyyy                    CONTROL CENTER                        hh:mm:ss
*-------------------------------- Recovery ---------------------------------*
| Option ===>                                                CTRLID: MSTRSRV1 |
|    Database ===> SQLDBA                                     NODE:  VMSYSTM1 |
|                                                                             |
|     I  INITIATE RECOVERY          Immediate recovery initiation             |
|     S  SCHEDULE RECOVERY          Schedule a recovery                       |
|     C  CANCEL RECOVERY            Immediate cancel of active recovery       |
|     R  INITIATE FAILURE RESTART   Restart after Recovery Failure            |
|     VJ VIEW JOB SCHEDULE          View database job schedule                |
|     H  HISTORY                    Display history of archives               |
|     L  LOG                        View log of previous recovery             |
|     VT VIEW TAPES                 View tape catalog                         |
|     TM TAPE MAINTENANCE           Make changes to tape catalog              |
|     ST SCRATCH TAPE ACQUISITION   Acquire a new SCRATCH tape and            |
|                                   add volid to TAPES file                   |
|                                                                             |
|           Enter OPTION, select DATABASE, press ENTER to process             |
|                                                                             |
*-----------------------------------------------------------------SQMRECOV----*
 PF:   1 Help    3 End
```

Figure 121. Database Recovery Menu

Usage Consideration: Before you invoke any kind of database recovery, be certain you are completely familiar with the database recovery topics and considerations outlined in the *DB2 Server for VM System Administration* manual.

RESTORE Process

The recovery process obtains information about database archive activity from the ARCHHIST file and not from the database history records. If you have not been using Control Center to manage database archives, COLDLOGs, logmode switches, and recoveries, then recovery information available to Control Center will not be complete.

Automatic recovery features were created to eliminate the need for manual support during a database restore. The process of running a manual database restore requires stopping the database, pre-recovery file and label definitions, startup parameter modifications, tape mounts, tape confirmations, starting the database, responding to console messages, interpreting console messages, reacting to recovery errors, keeping a recovery log, post-recovery database environment changes, and so on. The automated recovery features provided by the Control Center Data Restore interface will perform all these activities from start to completion.

Authorization

You must have database or Control Center administrator authority to execute RESTORE. Those persons identified as Control Center administrators on other databases or persons with Control Center user or operator authority cannot invoke RESTORE.

How Automated RESTORE Relates to Automated BACKUP

RESTORE uses information created during database archiving, and Data Restore backups and translates. A recovery control file and restore control file is created by Control Center as a result of reviewing a database's ARCHHIST and TRANHIST file data. A temporary file containing both the recovery and restore control files is sent to you. You will then be able to review exactly how the recovery will be controlled and decide to proceed or cancel the initiation process.

Several panels will prompt you for critical recovery information that will be used during the actual RESTORE.

Database Recovery Restore Sets using Data Restore

The files that drive a database recovery are referred to as a "restore set". Restore sets will vary depending upon the logmode, use of tape or disk, and the type of archive (user or database). Figure 122 on page 280 is an example of a Restore Set Report used for the recovery process that shows the recovery sets from Incremental Backup, Full Backup, archive and translate, an archive, and a non-Data Restore user archive.

```
RESTORE SET REPORT for Data Base TESTDB2      Date: 1998-07-10  Time: 07.15.59

RESTORE SET(s) generated using LOGMODE = L

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


 ....... Restore Set #1  From Data Base Archive Created 06-16-98 11:09:32     ....
 .......                                    Option: DUALINCBACKUP         ....
          Archive         Archive      Date      Time
        Type (DB or Log)  Sequence   Archived  Archived Series Volid
        ________________  ________  _________ ________ ______ ______
        DB USER ARCHIVE   Uarc #1   1998-06-16 11.15.39 300.01 USER BACKUP DR_BACKUP 06-16-98 11:09:32
        DataRestore Backup BAC1 @1   1998-06-16 11.15.39 300.01 PM1160
                           BAC2 @1   1998-06-16 11.16.39 300.01 PM1161
                           BAC1 @2   1998-06-16 11.17.39 300.01 PM1162
                           BAC2 @2   1998-06-16 11.17.39 300.01 PM1163

        Reference Backup   in Restore Set 2 series 300 created 06-15-98 13:35:20 >  DUALFULLBACKUP

                                .....................................
        ACTIVE Log         >>>>    Archived During The Recovery    <<<<

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


 ....... Restore Set #2  From Data Base Archive Created 06-15-98 13:35:20     ....
 .......                                    Option: DUALFULLBACKUP        ....
          Archive         Archive      Date      Time
        Type (DB or Log)  Sequence   Archived  Archived Series Volid
        ________________  ________  _________ ________ ______ ______
        DB USER ARCHIVE   Uarc #1   1998-06-15 13.43.06 300    USER BACKUP DR_BACKUP 06-15-98 13:35:20
        DataRestore Backup BAC1 @1   1998-06-15 13.43.06 300    VM8451
                           BAC2 @1   1998-06-15 13.43.06 300    BK2001

        Log Archive(s):    Log  #2   1998-06-16 11.09.44 300    PM1159
                                .....................................
        ACTIVE Log         >>>>    Archived During The Recovery    <<<<

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


 ....... Restore Set #3  From Data Base Archive Created 06-15-98 13:20:00     ....
 .......                                    Option: INCBACKUP             ....
          Archive         Archive       Date     Time
        Type (DB or Log)  Sequence   Archived  Archived Series Volid
        ________________  ________  _________ ________ ______ ______
        DB USER ARCHIVE   Uarc #1   1998-06-15 14.42.35 200.01 USER BACKUP DR_BACKUP 06-15-98 13:20:00
        DataRestore Backup BAC1 @1   1998-06-15 14.42.35 200.01 GP1111

        Reference Backup   in Restore Set 4 series 200 created 06-15-98 13:07:55 > ARCHIVEFULL/TRANSLATE

        Log Archive(s):    Log  #2   1998-06-15 13.35.27 200.01 VB4603
                                .....................................
                           #3   1998-06-16 11.09.44 300    PM1159
                                .....................................
        ACTIVE Log         >>>>    Archived During The Recovery    <<<<

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

Figure 122 (Part 1 of 2). Database Recovery Set Report Showing an Incremental Backup Recovery Set and other Archive Activity

```
....... Restore Set #4  From Data Base Archive Created 06-15-98 13:07:55     ....
.......                                    Option: ARCHIVEFULL/TRANSLATE ....
         Archive         Archive      Date      Time
       Type (DB or Log)  Sequence   Archived  Archived Series Volid
       ________________  ________  __________ ________ ______ ______
       Data Base Archive  Tape #1   1998-06-15 13.16.13 200    VB4556
                               #2   1998-06-15 13.16.16 200    AR0010

                          TDSK @1   1997-11-14 10.45.00 200    SQL35300 TRANSDSK * 401
                          BAC1 @2   1997-11-14 10.49.21 200    TR0001
                          BAC1 @3   1997-11-14 10.58.43 200    TR0002
       Log Archive(s):    Log  #3   1998-06-15 14.33.33 200    VP0066
                               ..................................
                               #4   1998-06-15 13.35.27 200.01 VB4603
                               ..................................
                               #5   1998-06-16 11.09.44 300    PM1159
                               ..................................
       ACTIVE Log         >>>>    Archived During The Recovery    <<<<

....... Restore Set #4 END ..................................................

....... Restore Set #5  From Data Base Archive Created 06-15-98 08:02:23     ....
.......                                    Option: UARCHIVE              ....
         Archive         Archive      Date     Time
       Type (DB or Log)  Sequence   Archived  Archived Series Volid
       ________________  ________  __________ ________ ______ ______
       DB USER ARCHIVE    Uarc #1   1998-06-15 08.58.47 400    USER BACKUP

       Log Archive(s):    Log  #2   1998-06-15 13.08.01 100    VB4554
                               ..................................
                               #3   1998-06-15 14.33.33 200    VP0066
                               ..................................
                               #4   1998-06-15 13.35.27 200.01 VB4603
                               ..................................
                               #5   1998-06-16 11.09.44 300    PM1159
                               ..................................
       ACTIVE Log         >>>>    Archived During The Recovery    <<<<

....... Restore Set #5 END ..................................................

....... Restore Set #6  From Data Base Archive Created 06-12-98 15:45:38     ....
.......                                    Option: ARCHIVE               ....
         Archive         Archive       Date     Time
       Type (DB or Log)  Sequence   Archived  Archived Series Volid
       ________________  ________  __________ ________ ______ ______
       Data Base Archive  Tape #1   1998-06-12 15.45.03 300    NA0001
                               #2   1998-06-12 15.48.03 300    NA0002
                               #3   1998-06-12 15.49.46 300    NA0003

       Log Archive(s):    Log  #4   1998-06-15 08.02.28 300    VP0093
                               ..................................
                               #5   1998-06-15 13.08.01 100    VB4554
                               ..................................
                               #6   1998-06-15 14.33.33 200    VP0066
                               ..................................
                               #7   1998-06-15 13.35.27 200.01 VB4603
                               ..................................
                               #8   1998-06-16 11.09.44 300    PM1159
                               ..................................
       ACTIVE Log         >>>>    Archived During The Recovery    <<<<

....... Restore Set #6 END ..................................................
```

Figure 122 (Part 2 of 2). Database Recovery Set Report Showing an Incremental Backup Recovery Set and other Archive Activity

Figure 123 on page 282 displays a Restore Set Selection panel specifying logmode **L** and Restore Set **1**.

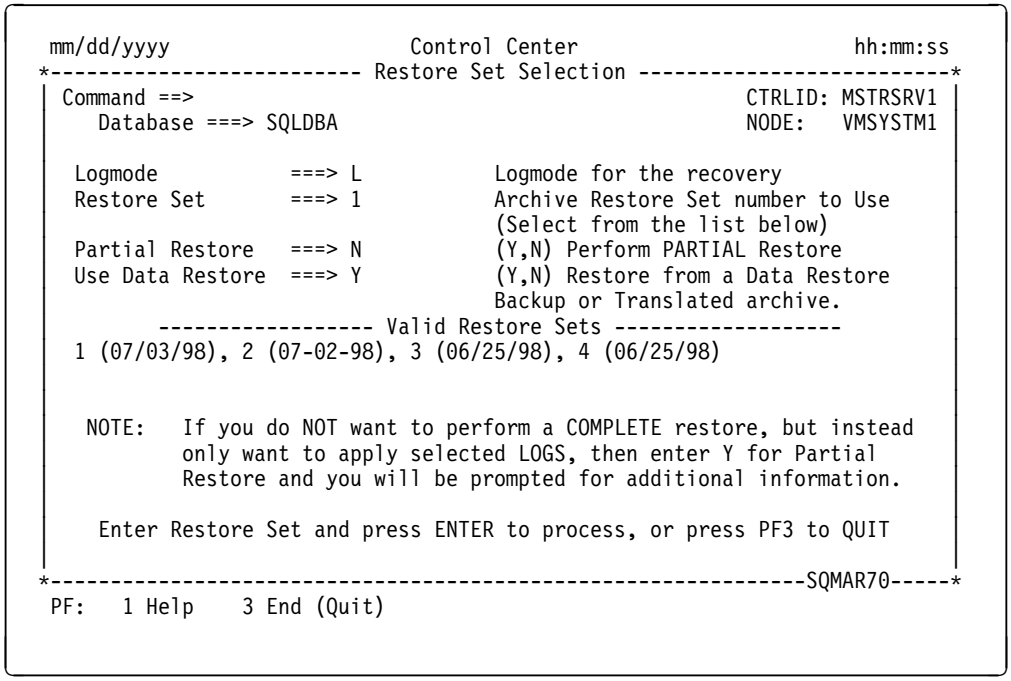


Figure 123. Control Center Restore Set Selection Panel using Data Restore

Selecting a RESTORE Set

After you have entered options on the Restore Set Selection Menu (Figure 123), Control Center lets you select one of the valid RESTORE sets that was also displayed in the earlier report. The RESTORE sets will always be numbered sequentially starting with 1, from the most recent archive to the oldest.

If you have been performing USER ARCHIVES using Data Restore BACKUP or have translated database archives, information regarding the tapes and/or files used will be presented for each restore set listed in the report. If a backup was performed, then all the tapes and files used in the primary backup will be identified with "BAC1" and all the DUAL backup output identified with "BAC2". If a TRANSLATE was performed on a database archive, all output information concerning the archive will be presented along with output information for the TRANSLATE. TRANSLATEs are treated as primary backups and are identified by "BAC1". The disk where the "SYS001", "DIRWORK" and "HEADER" files were written is identified by "TDSK".

Initiate Failure Restart

If the Data Restore RESTORE portion of the recovery fails, it may be possible to resolve the problem that caused the error and restart the restore process from the point of error, avoiding restarting from the beginning. Please note, however, that if the recovery process had not started the log archive restore portion of the recovery process, then the recovery must be restarted from the beginning.

History Files

The database ARCHHIST or TRANSHIST files will be updated during database recovery with pertinent archive information. If any log tapes were recovered, then the ARCHHIST file will be updated to indicate that log archives were done. This is done because recovery history information is recorded by the database in its history log, changing future restore set information. The recovered database archive and any recovered log archives will be distinguished from regular archive activity in the ARCHHIST file by a status of RECOVERED to the right of the archive record.

Recovery Log

During recovery of an archived database, status messages about key events during the recovery process will be sent to all the database administrators. The Control Center database status (refer to Chapter 17, “Database Status Tool” on page 203) will be continually updated and will indicate exactly what the recovery process is doing. You can also browse the database RECOVLOG, which is a log of all database console messages issued during the recovery process.

Figure 124 on page 284 provides an example of the RECOVLOG file for a database recovery.

```
 00.57.19  SQMCPSND (SQLEND QUICK) ENTERED
 00.57.19  ARI0032I The database manager has terminated.
 00.57.20  ARI0043I Database manager return code is 0.
 00.57.20 User Archive Recovery Starting. Database brought down.
 00:57:20  Recovery from USER ARCHIVE of DB2VM610 beginning.
 00:57:21  RESTORE Beginning Series 400 @  1997-01-03 16:14:35
 00:57:21  Submit routine SDRUSTRT executed.
 00.57.21  DASD 0207 DETACHED
 00.57.21  DASD 0200 DETACHED
 00.57.21  DASD 0202 DETACHED
 00.57.21  DASD 0300 DETACHED
 00.57.21  SQMCPSND (DRESTORE DB2VM610 MEDIA= TAPE TIMESTAMP= 1997-01-03 16:14:3
ENTERED
 00.57.21  DASD 0301 DETACHED
 00.57.21  DASD 0400 DETACHED
 00.57.21  DASD 0401 DETACHED
 00.57.21  DASD 0600 DETACHED
 00.57.21  DASD 0601 DETACHED
 00.57.21  DASD 0602 DETACHED
 00.57.21  DASD 0700 DETACHED
 00.57.21  DASD 0701 DETACHED
 00.57.22  DASD 0702 DETACHED
 00.57.22  ARI0796I End SQLSTART EXEC: 01/30/97 00:57:21 EST
 00.57.22  Ready; T=4.10/6.98 00:57:21
 00.57.22  DRESTORE DB2VM610 MEDIA= TAPE TIMESTAMP= 1997-01-03 16:14:35
 00.57.24  DRXSCIF1 SQLMSTRID SCIF SYNCPOINT
 00.57.25  SQMCPSND (SYNCPOINT SQMSTDV1) ENTERED
 00.57.29  Communications established with SQMSTDV1
 00.57.34  SQMCPSND (SYNCPOINT SQMSTDV1) ENTERED
 00.57.38  DMSLBD441R Enter VOLID information:
 00.57.38  SQMCPSND (VB3201) ENTERED
 00.57.38  DMSLBD441R Enter VOLID information:
 00.57.38  SQMCPSND (VB3202) ENTERED
 00.57.38  DMSLBD441R Enter VOLID information:
 00.57.38  SQMCPSND () ENTERED
 00.57.40  XTS9-100 Data Restore feature 6.1.0
 00.57.40  DMSXSU587I XEDIT:
 00.57.40  _|If var1 = 'DEF' Then CP LINK DB2VM610 207 207 W
 00.57.40  _|If var1 = 'DEF' Then CP LINK DB2VM610 200 200 W
 00.57.40  _|If var1 = 'DEF' Then CP LINK DB2VM610 202 202 W
 00.57.40  _|If var1 = 'DEF' Then CP LINK DB2VM610 300 300 W
 00.57.40  _|If var1 = 'DEF' Then CP LINK DB2VM610 301 301 W
 00.57.40  _|If var1 = 'DEF' Then CP LINK DB2VM610 400 400 W
 00.57.41  _|If var1 = 'DEF' Then CP LINK DB2VM610 401 401 W
 00.57.41  _|If var1 = 'DEF' Then CP LINK DB2VM610 600 600 W
 00.57.41  _|If var1 = 'DEF' Then CP LINK DB2VM610 601 601 W
 00.57.41  _|If var1 = 'DEF' Then CP LINK DB2VM610 602 602 W
 00.57.41  _|If var1 = 'DEF' Then CP LINK DB2VM610 700 700 W
 00.57.41  _|If var1 = 'DEF' Then CP LINK DB2VM610 701 701 W
 00.57.41  _|If var1 = 'DEF' Then CP LINK DB2VM610 702 702 W
 00.57.41  DMSXSU583I EOF:
 00.57.41  HCPDTV040E Device 0207 does not exist
 00.57.41  HCPDTV040E Device 0200 does not exist
 00.57.41  HCPDTV040E Device 0202 does not exist
 00.57.41  HCPDTV040E Device 0300 does not exist
 00.57.41  HCPDTV040E Device 0301 does not exist
 00.57.41  HCPDTV040E Device 0400 does not exist
 00.57.41  HCPDTV040E Device 0401 does not exist
 00.57.41  HCPDTV040E Device 0600 does not exist
 00.57.41  HCPDTV040E Device 0601 does not exist
 00.57.41  HCPDTV040E Device 0602 does not exist
 00.57.42  HCPDTV040E Device 0700 does not exist
 00.57.42  HCPDTV040E Device 0701 does not exist
 00.57.42  HCPDTV040E Device 0702 does not exist
 00.57.42  EZI08037I Message acknowledged. Awaiting processing
 00.57.42  EZI15043I Request for Mount of VB3201 has been queued for processing.
 00.57.42  WAITING FOR MOUNT. TYPE '1' TO CANCEL
```

Figure 124 (Part 1 of 5). Example database RECOVLOG file for Backup

```
 00.59.03  EZI15008I SL check verified. Label matches request.
 00.59.04  EZI15036I Tape positioned before label. Issue CMS command TAPE DVOL1
to get past label..
 00.59.04  Tape 181 attached R/O
 00.59.04  XTS9-304 Restore from user archive invoked
 00.59.04  XTS9-305 Current database will be destroyed
 00.59.04  XTS9-406 Enter 0(CANCEL) or 1(CONTINUE)
 01.17.58  SQMCPSND (1) ENTERED
 01.17.58  XTS9-008 Restoring directory
 01.19.33  XTS9-009     30604 directory blocks restored
 01.19.33  DMSACC724I 202 replaces B (207)
 01.19.33  XTS9-011 Restoring DDSK1
 01.19.34  XTS9-010         8 blocks restored
 01.19.34  DMSACC724I 400 replaces B (202)
 01.19.34  XTS9-011 Restoring DDSK4
 01.19.34  XTS9-010         4 blocks restored
 01.19.34  DMSACC724I 600 replaces B (400)
 01.19.34  XTS9-011 Restoring DDSK6
 01.19.34  XTS9-010         3 blocks restored
 01.19.34  DMSACC724I 601 replaces B (600)
 01.19.34  XTS9-011 Restoring DDSK7
 01.19.34  XTS9-010         1 blocks restored
 01.19.34  DMSACC724I 700 replaces B (601)
 01.19.34  XTS9-011 Restoring DDSK9
 01.19.34  XTS9-010         1 blocks restored
 01.19.34  DMSACC724I 701 replaces B (700)
 01.19.34  XTS9-011 Restoring DDSK10
 01.19.34  XTS9-010         1 blocks restored
 01.19.34  DMSACC724I 202 replaces B (701)
 01.19.35  XTS9-011 Restoring DDSK1
 01.19.46  XTS9-010      1682 blocks restored
 01.19.47  DMSACC724I 400 replaces B (202)
 01.19.47  XTS9-011 Restoring DDSK4
 01.19.48  XTS9-010       196 blocks restored
 01.19.48  DMSACC724I 600 replaces B (400)
 01.19.48  XTS9-011 Restoring DDSK6
 01.21.41  XTS9-010     17949 blocks restored
 01.21.41  DMSACC724I 601 replaces B (600)
 01.21.41  XTS9-011 Restoring DDSK7
 01.22.28  XTS9-010      7476 blocks restored
 01.22.28  DMSACC724I 700 replaces B (601)
 01.22.28  XTS9-011 Restoring DDSK9
 01.24.22  XTS9-010     17954 blocks restored
 01.24.22  DMSACC724I 701 replaces B (700)
 01.24.22  XTS9-011 Restoring DDSK10
 01.24.30  DMSTLM427I TAP1(181) EOV1 label read
 01.24.31  TAPE 0181 DETACHED
 01.24.31  EZI08037I Message acknowledged. Awaiting processing
 01.24.31  EZI15043I Request for Mount of VB3202 has been queued for processing.
 01.24.31  WAITING FOR MOUNT. TYPE '1' TO CANCEL
 01.24.31  WAITING FOR MOUNT. TYPE '1' TO CANCEL
 01.26.34  EZI15008I SL check verified. Label matches request.
 01.26.34  EZI15036I Tape positioned before label. Issue CMS command TAPE DVOL1
to get past label..
 01.26.34  Tape 181 attached R/O
 01.28.21 Recovery from BACKUP has completed.
 01.28.24 Recovery from User Archive has completed.
 01.28.25  SQMCPSND (PROFILE) ENTERED
 01.28.26  ARI0025I This is an INITIATION test for SCIF operation
 01.28.26  SQMCPSND (SQMID SQMSTDV1) ENTERED
 01.28.27  01/30/97 01:28:27 @SQMREQFL SQMSFILE SQLMSTR CONTROL (WAIT=300 FILE
SQLMSTR NAME
 01.28.29  01/30/97 01:28:29 @SQMREQFL SQMSFILE DB2VM610 LINKPWDS (WAIT=300 FILE
```

Figure 124 (Part 2 of 5). Example database RECOVLOG file for Backup

```
DB2VM610 L
 01.28.29  INKPWDS A0
 01.28.30  ARI0025I TVITEST: DMSTVI MODULE available
 01.28.30  01/30/97 01:28:30 @SQMREQFL SQMSFILE DB2VM610 PARMS (WAIT=300 FILE
DB2VM610 PARM
 01.28.33  ARI0025I DB2/VM VERSION 6 RELEASE 1 MOD 0
 01.28.34  File DB2VM610 DBEXTMAP A1 sent to SQMSTDV1 at WMAVM1 on 01/30/97
 01:28:34
 01.28.34  TAPEFUPD DB2VM610 Clear_filedef
 01.28.37  ARI0811I Archive filedef SCIF SYNCPOINT
 01.28.37   SQMRDR: DB2VM610 DBEXTMAP A received by SQMSTDV1 at WMAVM1
 01.28.37  SQMCPSND (SYNCPOINT SQMSTDV1) ENTERED
 01.28.38  ARI0811I Larchive filedef SCIF SYNCPOINT
 01.28.38  SQMCPSND (SYNCPOINT SQMSTDV1) ENTERED
 01.28.41  DMSLBD441R Enter VOLID information:
 01.28.41  SQMCPSND (DB35L402 LOG402 L 204) ENTERED
 01.28.42  ARI0811I Trace filedef SCIF SYNCPOINT
 01.28.42  SQMCPSND (SYNCPOINT SQMSTDV1) ENTERED
 01.28.45  ARI0717I Start SQLSTART EXEC: 01/30/97 01:28:44 EST.
 01.28.45  ARI0663I FILEDEFS in effect are:
 01.28.45  Z        DISK     DMSNAM   LOADLIB  *
 01.28.45  ARILARC  DISK     DB35L402 LOG402   L1
 01.28.45  ARITRAC  TAP2  SL  00001  VOLID SCRATCH
 01.28.45  ARISQLLD DISK     ARISQLLD LOADLIB  Q1
 01.28.45  BDISK    DISK     207
 01.28.45  LOGDSK1  DISK     200
 01.28.45  DDSK1    DISK     202
 01.28.45  DDSK2    DISK     300
 01.28.46  DDSK3    DISK     301
 01.28.46  DDSK4    DISK     400
 01.28.46  DDSK5    DISK     401
 01.28.46  DDSK6    DISK     600
 01.28.46  DDSK7    DISK     601
 01.28.46  DDSK8    DISK     602
 01.28.46  DDSK9    DISK     700
 01.28.46  DDSK10   DISK     701
 01.28.46  DDSK11   DISK     702
 01.28.46  ARIUSRDD DISK     USERLIB  LOADLIB  *
 01.28.46  ARIARCH  TAP1  SL  00001
 01.28.46  ARISSCR  DISK     ARISSCR  MACRO    *
 01.28.46  ARISSTR  DISK     ARISSTR  MACRO    *
 01.28.46  ARISCCS  DISK     ARISCCS  MACRO    *
 01.28.46  NETID    DISK     SNA      NETID    *
 01.28.46  ARISPOOL DISK     DB2VM610 ARISPOOL *
 01.28.47  ARI0020I Virtual machine addressing mode = 31
 01.28.47           Virtual machine storage size = 32768KB
 01.28.48  ARI0025I The program ARISQLDS is loaded at 7B7A00.
 01.28.48  ARI0025I The program ARIXRDS is loaded at 6246B8.
 01.28.48  ARI0025I The program ARIXSXR is loaded at D510D8.
 01.28.48  ARI0025I The program ARICMOD is loaded at DE25A8.
 01.28.48  ARI0015I PROTOCOL parameter value is SQLDS.
 01.28.48  ARI0015I ACCOUNT parameter value is N.
 01.28.48  ARI0015I DUMPTYPE parameter value is F.
 01.28.48  ARI0015I LOGMODE parameter value is L.
 01.28.48  ARI0015I STARTUP parameter value is U.
 01.28.49  ARI0015I SYSMODE parameter value is M.
 01.28.49  ARI0015I EXTEND parameter value is N.
 01.28.49  ARI0015I DBNAME parameter value is DB2VM610.
 01.28.49  ARI0015I RESID parameter value is DB2VM610.
 01.28.49  ARI0015I PARMID parameter value is DB2VM610.
 01.28.49  ARI0015I TRACDBSS parameter value is 00000000000.
 01.28.49  ARI0015I TRACDSC parameter value is 00.
 01.28.49  ARI0015I TRACRDS parameter value is 0000000.
 01.28.49  ARI0015I TRACCONV parameter value is 0.
```

Figure 124 (Part 3 of 5). Example database RECOVLOG file for Backup

```
01.28.50  ARI0015I TRACSTG parameter value is 0.
01.28.50  ARI0016I TRACEBUF parameter value is 0.
01.28.50  ARI0016I ARCHPCT parameter value is 90.
01.28.50  ARI0016I CHKINTVL parameter value is 10.
01.28.50  ARI0016I NCSCANS parameter value is 30.
01.28.50  ARI0016I NCUSERS parameter value is 5.
01.28.50  ARI0016I NDIRBUF parameter value is 30.
01.28.50  ARI0016I NLRBS parameter value is 2520.
01.28.50  ARI0016I NLRBU parameter value is 1000.
01.28.50  ARI0016I NPACKAGE parameter value is 10.
01.28.50  ARI0016I NPACKPCT parameter value is 30.
01.28.51  ARI0016I NPAGBUF parameter value is 30.
01.28.51  ARI0016I SLOGCUSH parameter value is 95.
01.28.51  ARI0016I SOSLEVEL parameter value is 10.
01.28.51  ARI0016I DISPBIAS parameter value is 7.
01.28.51  ARI0208D Did the user-restore of the directory and
01:28:51 "1" entered in response to ARI0208D.
01.28.51 DBEXTENT(S) execute successfully?
01.28.51           Enter 0(No) or 1(Yes)
01.28.51  SQMCPSND (1) ENTERED
01.28.52  ARI0255I The database manager is initiating a log archive. When it
01.28.52           is completed, the restore process will continue.
01.28.52  ARI0293I Archive is starting.
01.28.52 DB2VM610 Log Archive started.
01.29.00 LOG ARCHIVE completed, series 400 volid DB2VM610 01309701 L 204
01.29.00  ARI0260I The restore set for this database archive is
01.29.00           as follows:
01.29.00  ARI0238I     database  archive                    01-03-97 16:14:35
01.29.00  ARI0261I     log archive - disk  DB2VM610 01039704 01-03-97 17:05:51
01.29.01  ARI0261I     log archive - disk  DB2VM610 01309701 01-30-97 01:28:52
01.29.01  ARI0239I External labeling of this archive is:
01.29.01               Type:      log  archive
01.29.01               Timestamp: 01-03-97  17:05:51
01.29.01  ARI0252I     Medium:    disk  DB2VM610 01039704 *
01.29.01  ARI0250D The above information describes the next log archive
01.29.05 CONTINUE
01.29.05           to be used in the restore process. Enter either:
01.29.05             CONTINUE     to restore this log archive,
01.29.05             STOP SYSTEM  to interrupt this restore process, or
01.29.06             END RESTORE  to end this restore process.
01.29.06  SQMCPSND (CONTINUE) ENTERED
01.29.06  ARI0240I Restoring log disk.
01.29.46  ARI0283I Log analysis is complete.
01.29.46  ARI0282I LUW UNDO is completed.
01.29.46  ARI0281I LUW REDO is completed.
01.29.46  ARI0239I External labeling of this archive is:
01.29.46               Type:      log  archive
01.29.46               Timestamp: 01-30-97  01:28:52
01.29.46  ARI0252I     Medium:    disk  DB2VM610 01309701 *
01.29.46  ARI0250D The above information describes the next log archive
01.29.53 CONTINUE
01.29.53           to be used in the restore process. Enter either:
01.29.53             CONTINUE     to restore this log archive,
01.29.53             STOP SYSTEM  to interrupt this restore process, or
01.29.53             END RESTORE  to end this restore process.
01.29.53  SQMCPSND (DETACH 183) ENTERED
01.29.53  HCPDTV040E Device 0183 does not exist
01.29.53  SQMCPSND (CONTINUE) ENTERED
01.29.53  ARI0240I Restoring log disk.
01.30.19  ARI0283I Log analysis is complete.
01.30.19  ARI0282I LUW UNDO is completed.
01.30.21  ARI0281I LUW REDO is completed.
01.30.22  ARI0143I The application server has been initialized
01.30.22           with the following values:
01.30.22           CHARNAME = INTERNATIONAL, DBCS = NO, CHARSUB = SBCS,
01.30.22           CCSIDSBCS = 500, CCSIDMIXED = 0, CCSIDGRAPHIC = 0.
01.30.22  ARI0134I Application server DB2VM610 has been
01.30.22           identified as a local resource.
01.30.22  ARI0060I Database manager initialization complete.
01.30.23  ARI0045I Ready for operator communications.
```

Figure 124 (Part 4 of 5). Example database RECOVLOG file for Backup

```
01.30.23  SQMCPSND (DETACH 183) ENTERED
01.30.23  HCPDTV040E Device 0183 does not exist
01.30.23  SQMCPSND (SQLEND) ENTERED
01.30.23  ARI0028I The database manager is terminating.
01.30.23  ARI0065I Operator command processing is complete.
01.30.23  ARI0032I The database manager has terminated.
01.30.23  ARI0043I Database manager return code is 0.
01.30.23 DB2VM610 RECOVERY SUCCESSFUL! Database is being brought back up.
01.30.24 SQMA043I=> Database being started with logmode =  L
```

Figure 124 (Part 5 of 5). Example database RECOVLOG file for Backup

Chapter 23. Data Restore UNLOAD

Overview

The Data Restore UNLOAD function allows the user to unload one or many DBSPACES with the database up or down. It is different from the DBSU UNLOAD utility which is used in Control Center for DBSPACE reorganizations. The Data Restore UNLOAD does not unload DBSPACE table records in key order.

When the tables are reloaded using Data Restore, the tables are not reorganized. The DBSU and Data Restore unload formats are not compatible with each other.

However, because Data Restore unloads the active pages in the DBSPACE, it provides a faster unload than DBSU and can be used to provide a backup of critical DBSPACES. Data Restore UNLOAD unloads DBSPACES and Data Restore RELOAD reloads tables. Tables can later be recovered from either a Data Restore BACKUP, TRANSLATE, or UNLOAD.

Control Center will give the user a way to unload and store DBSPACES for later use.

How the UNLOAD Tool Works

An UNLOAD may be desired to provide a temporary backup for a DBSPACE before changing its tables. Or, a DBSPACE may be unloaded as part of a regular backup strategy for vital system tables.

To manage the different size and frequency requirements for unloading DBSPACES, Control Center will use an UNLOAD ID (UNLOAD identification name) to identity each unload event. The UNLOAD ID is an eight (8) character name that will be used in the database UNLTAPES file to identify which files or tapes are to be used with a particular unload event. The UNLOAD ID will be used to identify the file (*unloadid* UNLDDBSP) which will contain the list of DBSPACES to unload. The UNLOAD ID will also be used in the UNLDHIST file to refer to the unload event that has completed.

To set up an UNLOAD event, the DBA will use the Control Center menu interface to select which DBSPACES to unload, the file or tapes to use, a unique UNLOAD ID that will be used to identify which output media to use, and the Data Restore values to use in the SYSIN file.

Data Restore allows you to UNLOAD a maximum of 90 DBSPACES. All the DBSPACES selected must fit on a maximum ten (10) lines of the SYSIN file. This is a Data Restore restriction. In practice, it will not be possible to fit 90 DBSPACES on 10 lines unless all the DBSPACE owners are "PUBLIC" and the DBSPACEnames are very short.

At execution time, the Data Restore machine will perform necessary FILEDEFS and LABELDEFS, create the SYSIN file, and execute the Data Restore UNLOAD routine (XTS91001 MODULE). The file or tape output will be recorded in the

database UNLDHIST file, and the SYSPRINT file appended to the database UNLDDESC (unload description) file.

The unloaded DBSPACE tables can later be reloaded using the RELOAD tool.

UNLOAD Control and OUTPUT files

The basic files created will be:

Filename	Description
unloadid UNLDDBSP	If more than one DBSPACE is selected, this file will contain a formatted list of the DBSPACES that will be used in the SYSIN file. The file is kept on the Control Center service machine and will be used each time the UNLOAD ID job is executed.
dbmach UNLDSYIN	The SYSIN file created on the Data Restore machine to execute the UNLOAD. This file contains the Data Restore commands and options to be executed, and additional information provided by Control Center.
dbmach UNLDLIST	The output listing of the UNLOAD created by Data Restore This file will list all the tables that were unloaded, and shows the same table list as you would see in the SYSPRINT for a DESCRIBE function. All SYSIN data is reproduced in the listing file.
dbmach UNLDLOG	A log of the current UNLOAD. The file is maintained on the Control Center service machine. A backup will be made of the previous UNLDLOG file.
dbmach UNLDDESC	A copy of the UNLDLIST (SYSPRINT) file for every successful UNLOAD will be appended to the UNLDDESC file. This file will maintain every UNLOAD performed on the database. Periodically, the file will have to be edited to reduce its size. This can be done by using the filelist option on the Control Center General Utilities menu. (**SQM =g.f** *dbmach* UNLDDESC).
UNLOAD ERROR	If errors occur before, during, or after execution of the UNLOAD on the Data Restore machine, the ERROR file will indicate the type of error. If an error condition was detected during processing, then the name of the module and the approximate line number of the error will be presented. The ERROR file is sent to all database administrators listed in the specific database PARMS file.

Input and Output Media Processing

The output media (unloaded DBSPACE tables) is determined by the selection made on the set up panels. The media selected (DISK/TAPE/SCRATCH) must match the media type represented in the UNLTAPES file. If SCRATCH is selected, then the tapes listed in the UNLTAPES file will not be re-used. New SCRATCH tapes will be requested.

The UNLOAD (ARCHIV) tape will use cuu 181.

Database UNLTAPES File

The UNLTAPES file is different from other Control Center tapes files. The differences are discussed in the following sections.

UNLOAD ID

Instead of "TYPE", referring to the type of operation the media will be used for (for example, "ARCHIVE", "LOG", "BACKUP"), this entry will identify the unique name assigned by the user for the UNLOAD event. It is critical that this UNLOAD ID be the exact name as indicated in the set up panels. Control Center will use this name when requesting tapes to be used in FILEDEF, LABELDEF, and tape mount operations.

UNLOAD Series

In the UNLTAPES file, it is not necessary to have more than one series of output media listed for the UNLOAD ID. If you only wish to maintain one level of DBSPACE backup, then your UNLOAD ID should be listed in series 100 only.

However, if you wish to keep multiple levels of backups for a DBSPACE, the UNLOAD ID may have multiple series (100, 200, 300, and so on), with each series using different VOLID or file names. If more than one series is defined for an UNLOAD ID, Control Center will cycle through each series until it starts back at series 100 again.

If output is to disk, then the previous file will be overwritten. If you change the file name of your output, be sure that there is enough space on the disk to hold the new file.

For disk output, the cuu indicated MUST be the link address on the Data Restore machine where the output will be written. The disk should not be linked and accessed prior to execution. Control Center will link to this address using the first available access cuu starting at "120". The first free filemode will be used. If the operation ends normally, the disk will be detached.

UNLOAD Tapes

Sometimes you will want to unload a DBSPACE to temporary storage until another operation is complete. You can define an UNLOAD ID that can be used many times to hold different DBSPACES. Each UNLOAD using this UNLOAD ID will overwrite the previous data.

Figure 125 on page 292 is an example of the database UNLTAPES file entries required to process different unloads.

```
100  UNLOAD1  97003  11:41:49 FILLED UNLD1100 DB2VM510 * 400
100  UNLOAD2  97003  10:57:08 FILLED VB3073
100  UNLOAD3  96364  13:51:42 FILLED VB3075
100  WORKDISK 97008  12:43:44 FILLED TEMP100 DB2VM510 * 403
200  UNLOAD1  97006  12:31:21 FILLED UNLD1200 DB2VM510 * 400
200  UNLOAD2  97010  14:29:09 FILLED VB3074
300  UNLOAD2  97017  14:29:09 FILLED VB3078
```

Figure 125. Example of UNLTAPES file entries for UNLOAD

Passwords

UNLOAD requires the database password for user SQLDBA in the "DBAPW=" SYSIN file control statement.

If database disks are password protected, UNLOAD requires the "READ" password for all of the database disks in the "READPW=" SYSIN file control statement.

A password file (drmach LINKPWDS) file is required. Refer to Appendix C, “Password Support” on page 501 for instructions of how to implement Data Restore password support.

How to Invoke UNLOAD

To invoke UNLOAD, select Option **U** from the Data Restore Menu. The first entry panel of the UNLOAD tool is shown in Figure 126.

```
 mm/dd/yyyy                      CONTROL CENTER                       hh:mm:ss
*------------------------------ UNLOAD Utility ------------------------------*
|                                                          CTRLID: MSTRV1    |
| Database ==> SQLDBA                                      NODE:   VMSYSTM1  |
|                                                          DRMACH: DREST1    |
| Select DBSPACES ==> 2 (1= *, 2= Search List)                               |
|                       (3= Input file, 4= Type in one dbspace)              |
|                                                                            |
|    DBSPACE OWNER ==> ________  (blank for ALL, PUBLIC, PRIVATE, ownername) |
|    DBSPACE NAME  ==> ________________ (blank for ALL, use % for wildcard)  |
|    DBSPACE PAGES:  MINIMUM ==> 1       MAXIMUM    ==> 9999999             |
|    STORPOOL      ==> ____            (blank for ALL Pools or specific pool |
|                                                                            |
|                                                                            |
|    DBSPACE INPUT FILE ==> ____________________ Filename, filetype, filemode|
|                                                                            |
|                                                                            |
|       The % SQL wildcard character can be used when specifying the         |
|       Dbspace Owner or Dbspace Name values for selection.                  |
|                                                                            |
*-----------------------------------------------------------------DRUNLD2------*
 PF1 HELP    PF3 QUIT    PF4 EXIT    PF5 Main Menu    Enter Retrieve List
```

Figure 126. Data Restore UNLOAD Panel

On this panel, you will tell Control Center what DBSPACES you wish to unload. If you choose to type the name of one DBSPACE, or to unload all DBSPACES('*'), you will be required to select the media, mode, and condition options.

Data Restore UNLOAD Options

Parameter	Description
Select DBSPACES	Used to specify how you wish to enter the name or names of DBSPACES to be unloaded. • 1 - Use "*" to unload all DBSPACES. The number of DBSPACES must not bt greater than 90. • 2 - Indicates that you wish to use an SQL SELECT query to generate a list of DBSPACES based on owner, DBSPACEname, pages, and pool parameters. Wild cards can be used for the owner and DBSPACEname. The database must be available, and on the same NODE as the user who is executing this option. • 3 - Indicates that you wish to use an input list file that contains a list of DBSPACEowners and DBSPACEnames to unload. Each record must be in the format: DBSPACEowner DBSPACEname. All other information in the record will be ignored. If the DBSPACEowner or DBSPACE name contains blanks, then it must be enclosed with double quotes ("). The list cannot contain wildcards. The validity of the DBSPACES will not be checked. This option can be used when the database is down. • 4 - Indicates that you wish to enter the name of single DBSPACEowner and DBSPACEname. The name cannot contain any wildcards. The validity of the DBSPACE will not be checked. This option **can** be used when the database is down.
Dbspace Owner	A valid DBSPACE owner. If Option **2** is used, then the owner can contain wildcards, or be "PRIVATE" for all private DBSPACES, or be left blank for "ALL" owners.
Dbspace Name	A valid DBSPACE name. If Option **2** is used, then the name can contain wildcards, or be left blank for "ALL" DBSPACEnames.
Pages	A valid minimum and maximum NPAGES to consider.
STORPOOL	A valid storage pool number. Include a minus sign (-) for non-recoverable pools.

If you choose DBSPACE selection Option **2** from the UNLOAD Dbspace Utility Menu, you will be presented with a list of the DBSPACES that match your selection criteria. You may UNLOAD a maximum of 90 DBSPACES; all the DBSPACES selected must fit on a maximum ten (10) lines of the SYSIN file. This is a Data Restore restriction.

You may eliminate DBSPACES from your list by indicating "O" on the left hand input line of the list panel. Control Center will build the SYSIN DBSPACE list for you. If the DBSPACES you selected will not fit on ten lines, or if you have more than 90 DBSPACES, you will be returned to the DBSPACE list panel.

The number of DBSPACES selected is indicated in the **Selected DBSPACES** field in the upper right hand corner of the panel. The Select UNLOAD DBSPACES panel is shown in Figure 127, and the OMIT DBSPACE Option **O** has been specified.

```
 mm/dd/yyyy                      CONTROL CENTER                       hh:mm:ss
*--------------------------- Select Unload Dbspaces -------------------------*
|  Database => SQLDBA              Selected DBSPACES => 7                    |
|                                 PCTFREE  PCT                               |
| SEL OWNER     NAME              ACQ ALT  INDX  POOL  NPAGES   NACTIVE      |
| --- -------- ------------------ -------  ----  ---- --------  -------      |
|   _  D128980  DUKE_BIGTEST        10   0    10    -8    40960    16660      |
|   O  PUBLIC   BDGT_MSTRCA_DBS     10   0     5     6    20480       -1      |
|   _  PUBLIC   DATARFTR            10   0    33     4     6400       32      |
|   _  PUBLIC   DUKE_BIGTEST        10   0    30     6    40960    18124      |
|   _  PUBLIC   HELPTEXT            10   0    33     1     8192     1190      |
|   _  PUBLIC   SAMPLE              10   0    33     6    40960        4      |
|   _  PUBLIC   SORTA_BIG           10   0     5     6    20480       -1      |
|   _  SQMDEMO3 DUKE_BIGTEST        10   0    30    -8    40960    18369      |
|                                                                            |
|                                                                            |
|  Select:  O = Omit Dbspace                                                 |
|                                                                            |
|                               Page 1     of 1                              |
*----------------------------------------------------------------DRUNLD3------*
1 Help         3 QUIT         5 Continue
9 Sort/Name   10 Sort/Nact   11 Sort/Pages   12 Sort/Pool
```

Figure 127. Select Unload Dbspaces Panel

The menu which follows (Figure 128) is where the UNLTAPES (UNLOAD TAPES) file can be selected or updated. You must either verify the UNLOAD ID, series, media (file/tapes) to be used for the UNLOAD, or create a new entry for your unload.

```
 mm/dd/yyyy                      CONTROL CENTER                      hh:mm:ss
*---------------------- Database UNLTAPES File Update ---------------------*
| Command ===>                                                CTRLID: MSTSRV1 |
|  Database => SQLDBA                                           NODE: VMSYSTM1 |
|                                          VOLID/                           |
| SERIES  TYPE     DATE   TIME     STATUS   FILENAME  FILETYPE  FM  ADDRESS  |
|  100    UNLOAD1  97003  11:41:49 FILLED   UNLD1100  DB2VM510   *   400     |
|  100    UNLOAD2  97003  10:57:08 FILLED   VB3073                          |
|  100    UNLOAD3  96364  13:51:42 FILLED   VB3075                          |
|  100    WORKDISK 97008  12:43:44 FILLED   TEMP100   DB2VM510   *   403     |
|  200    UNLOAD1  97006  12:31:21 FILLED   UNLD1200  DB2VM510   *   400     |
|  200    UNLOAD2  97010  14:29:09 FILLED   VB3074                          |
|  200    UNLOAD3  96334  17:25:17 FILLED   VB3076                          |
|  300    UNLOAD2  97017  14:29:09 FILLED   VB3078                          |
|                                                                            |
|                                                                            |
|                                                                            |
|   Make changes, place D in SERIES to DELETE , press PF10 to process        |
|                           Page 1     of 1                                  |
|                                                                            |
*----------------------------------------------------------------SQMTP20-----*
 PF:  1 Help   3 End    4 Add Tape   7 Bkwd     8 Fwd     10 Process updates
```

Figure 128. Database UNLTAPES File Update Panel

From the panel below, you will decide whether to execute or to schedule the UNLOAD. You must also specify UNLOAD ID for the UNLOAD, who to notify, and make any necessary changes in the SYSIN options.

```
 mm/dd/yyyy                     Control Center                       hh:mm:ss
*-------------------------  DRF UNLOAD EXECUTION  --------------------------*
|  DATABASE ==> SQLDBA                                    CTRLID: MSTRSRV1 |
|                                                         NODE:   VMSYSTM1 |
|                                                                          |
|  Execution       ==> 1            ( 1= Schedule, 2 = Execute Immediately |
|                                                                          |
|  UNLOAD ID       ==> ________     (Unique name for UNLOAD job )          |
|   The Unload Id will indicate which disk or tape set to be used.         |
|   The Unload ID MUST be the same as the ID used in the UNLTAPES file.    |
|                                                                          |
|                                                                          |
|  NOTIFY ==> ________  (Userid)   ________   (Node )                      |
|                                                                          |
|  MEDIA  ==> 1        ( 1= DISK,    2= Predefined TAPE, 3= SCRATCH Tape)  |
|  MODE   ==> 1        ( 1= Online,  2= Offline )                         |
|  COND   ==> 1        ( 1= Include, 2= Exclude )                         |
|  WRKSIZE => 2048     ( Npages * 5 / 1024, default = 2048)               |
|                                                                          |
|                                                                          |
*-------------------------------------------------------------DRUNLD4-----*
  PF1 HELP   PF3 QUIT     PF4 Modify/View UNLTAPES file     ENTER  Execute
```

Figure 129. UNLOAD Execution Panel

Parameter	Description
Execution	Used to specify whether to (2) execute the unload immediately or (1) schedule for later execution. If schedule is selected, you will be presented with the scheduling panel.
UNLOAD ID	Unique name that will identify the unload. The name can be a maximum of 8 characters.
NOTIFY	Userid and NODE of person to notify when the UNLOAD is successful.
Media Type	Choose disk, pre-defined tapes, or SCRATCH tapes. If disk or pre-defined tapes are chosen, the entries in the database UNLTAPES must match the media type selected. If SCRATCH tape is selected, the tapes defined in the database UNLTAPES file will be replaced with the new SCRATCH tapes used. You will have the option of changing media on the last panel.
MODE	The MODE is the Data Restore UNLOAD SYSIN file command option "MODE=" which indicates whether the database will be running when the UNLOAD is executed. Select ONLINE (1) if the database will be running when the UNLOAD is performed. Select OFFLINE (2) if the database will not be running when the UNLOAD is performed. The default is ONLINE.
COND	The condition is the Data Restore UNLOAD SYSIN file command option "COND=", which indicates if the list of DBSPACES in the SYSIN file will be unloaded or excluded. INCLUDE (1) indicates that the list of DBSPACES will be unloaded. EXCLUDE (2) indicates that all DBSPACES except for those listed will be unloaded. EXCLUDE is not valid with the "*" DBSPACES selection option. The default is INCLUDE.
WRKSIZE	The WRKSIZE if the Data Restore UNLOAD SYSIN file command option "WRKSIZE=", which indicates the work size value to use for the UNLOAD. The default is 2048.

After the UNLOAD

The database UNLDHIST file will be updated with the file or tapes that were produced for the UNLOAD and the date and time stamps of the UNLOAD.

If the UNLOAD was successful, the UNLDLIST (SYSPRINT) file will be sent to the Control Center service machine. This file will be appended to the database UNLDDESC (unload describe) file to provide a running record of all successful unloads. When this becomes too large, it can be manually edited to remove old entries by using the General Utilities Filelist tool (SQM =G.F * UNLDDESC).

Chapter 24. Data Restore RELOAD

Overview

RELOAD allows the user to reload and recreate up to 90 tables from a Data Restore BACKUP, translated database archive, or a Data Restore UNLOAD. Control Center requires that a database archive first be translated before it is used as an input source for RELOAD.

Additionally, if the RECOVERY option is set to 'YES', then Data Restore will read log archives and the active log to record additional table LUWs executed since the BACKUP. This source can later be examined with the LISTLOG function and re-executed using the APPLYLOG function.

Data Restore RELOAD requires that the database server be active. It is different from a DBSU RELOAD utility which is used in Control Center for DBSPACE reorganizations, since Data Restore RELOAD does not reload table records in key order. When the tables are reloaded using Data Restore RELOAD, the tables are not reorganized. The DBSU and Data Restore RELOAD formats are not compatible with each other.

To RELOAD from an Incremental Backup, Data Restore first recovers the Incremental Backup pages and then Full Backup pages.

The Control Center RELOAD tool will set up, execute, and manage the Data Restore RELOAD process.

How the RELOAD Tool Works

RELOAD set up consists of several parts.

1. Selection of the input source for the RELOAD (BACKUP, UNLOAD, or TRANSLATE)
2. Examination of the recovery sets available for that input source
3. Building the RELOAD command for one or more tables. For every table listed, the user can choose the appropriate PURGE, NEW, ADD, REPLACE option and enter the required TNAME, CREATOR, NEWTNAME, NEWCREATOR, DBSPACE, OWNER values.
4. A RELOAD ID (RELOAD identification name) will be chosen which will be used to track the RELOAD event.
5. The RELOAD control file (RELOAD ID RELDCTL) will be created. This file will include the source file or tapes to be used for the RELOAD, and the formatted list of table RELOAD commands. This file will be sent to the Control Center service machine. The database RELDCTL file includes entries for both the Incremental Backup and the Full Backup. Figure 130 on page 298 is an example of a RELDCTL file for Incremental Backup.
6. At execution time, required FILEDEFS and LABELDEFS will be performed, the SYSIN file created, tapes mounted (if required), and the RELOAD routine (XTS91001 MODULE) executed.
7. If RECOVERY=YES, and the Control Center TAPEPWD=YES option is selected, the Control Center service machine will attempt to change the READ

passwords for the database LOG archive tapes that are in the RELDCTL file to 'ALL'. This will allow the tapes to be read by the Data Restore machine. These passwords will be removed when the RELOAD is finished.

8. If the RELOAD is successful, and RECOVERY=YES, then a LISTLOG will automatically be performed.
9. The SYSPRINT file (RELOAD ID RELDLIST) will be sent to the Control Center service machine and to the database administrators.

```
:DBMACH=TESTDB2 RELOAD RECOVERY=YES from BACKUPREC @06-16-98 11:09:32
#1 DR BACKUP      1998-06-16 11.15.39 300.01 N PM1160
#2 DR REFBACKUP   1998-06-15 13.43.06 300    N VM8451
#3 LOG Archive ACTIVE LOG
END RESTORE SET
:CONTROL.BEGIN :DRSTRT 19980210 :DBMACH=SQL61DB1
RELOAD CREATOR=TESTDBA TNAME=ACTIVITY
      FUNCT=PURGE
:CONTROL.END
```

Figure 130. RELOAD Control File (RELDCTL) for Incremental Backup

RELOAD Control and OUTPUT files

The basic files created will be:

Filename	Description
reload id RELDCTL	The file will contain information about the file or tapes that will be used as the input media for the RELOAD. This file will also contain the formatted table RELOAD commands that will be used in the SYSIN file. The file is kept on the Control Center service machine and will be used each time the RELOAD ID job is executed.
dbmach RELDSYIN	The SYSIN file created on the Data Restore machine to execute the RELOAD. This file contains the Data Restore commands and options to be executed and additional information used by Control Center
dbmach RELDLIST	The output listing of the RELOAD created by Data Restore. The file will be sent to the Control Center service machine and DBAs when the RELOAD is complete.
dbmach LMBRLGx	If RECOVERY=YES, then the LMBRLG1, LMBRLG2 and LMBRLG3 files will be created on the 191 disk of the Data Restore machine. to extract all changes referenced in the log for the reloaded tables.
dbmach LMBRWRK	Work file created by Data Restore if the tables contain LONG columns.
dbmach LISTLOG	If RECOVERY=YES, a LISTLOG will be executed and the file will contain the LISTLOG SYSPRINT created by Data Restore This file will list all the LUWS recorded in the work files for the current

	RELOAD RECOVERY= YES. This file will be sent to the Control Center service machine.
dbmach RELDLOG	A log of the current RELOAD. The file is maintained on the Control Center service machine. A backup will be made of the previous RELDLOG file.
RELOAD ERROR	If errors occur before, during, or after execution of the function on the Data Restore machine, the RELOAD ERROR file will indicate the error type. If an error condition was detected during Control Center processing, then the name of the module and the approximate line number of the error will be presented. The ERROR file is sent to all database administrators listed in the specific database PARMS file.

Input and Output Media Processing

The output media (LMBRWRK and LMBRLGx work files) will be created on the 191 disk of the Data Restore machine.

The RELOAD input tape (ARCHIV) will use CUU 182, and the LOG input tape (LARCHIV) will use CUU 181.

No tapes file processing is required.

Passwords

RELOAD requires that the database password for user SQLDBA be included in the 'DBAPW=' SYSIN file control statement.

If the database disks are password protected, RELOAD requires the 'READ' password for all of the database disks be included in the 'READPW=' SYSIN file control statement.

A password file (drmach LINKPWDS) file is required. Refer to Appendix C, “Password Support” on page 501, for instructions on how to implement Data Restore password support.

If RECOVERY=YES, then the Data Restore machine must have the authority to READ the database's LOG archive files or tapes. This can be handled manually or attempted through the TAPEPWD option.

How to Invoke RELOAD

To invoke RELOAD, select Option **R** from the Data Restore Menu (Figure 115 on page 263).

The first entry panel for the RELOAD tool is shown in Figure 131 on page 300.

The menus you will use to set up your RELOAD will depend upon whether you wish to perform a normal RELOAD from a BACKUP, TRANSLATE, or UNLOAD, or perform a RELOAD with a LOG recovery, or a restart of a previous RELOAD.

1. RESTART: If you select to restart a previous reload, you will reuse the input media and table list in a RELOAD ID RELDCTL file that already exists. You will be taken directly to the RELOAD execution menu.
2. NORMAL RELOAD: If Log Recovery = NO, you will be presented with the RELOAD Selection menu. You will view the HISTORY file for the BACKUP, TRANSLATE, or UNLOAD input source you selected, determine the DATE and TIME stamp for the input you wish to RELOAD, and proceed to create your table list.
3. LOG RECOVERY: If Log Recovery = YES, then you will be presented with a set of RESTORE Set Selection menus, similar to what is used for the database recovery and TRANSLATE tools. Here you determine your restore set, view the input media control file, and then proceed to create your table list.

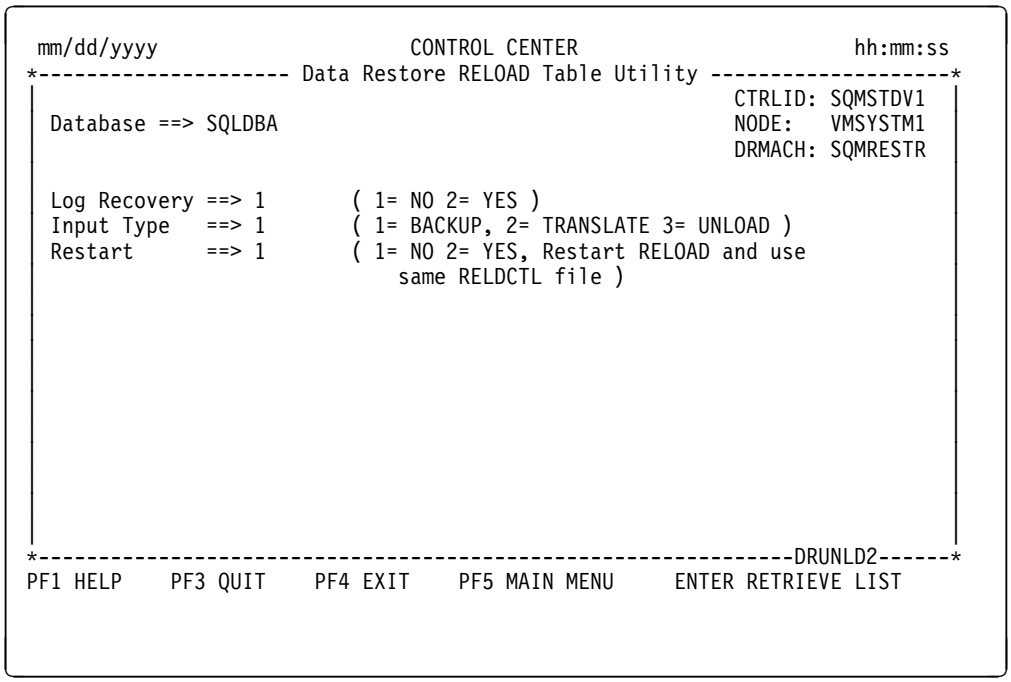


Figure 131. Data Restore RELOAD Dbspace Utility Menu

RELOAD Parameters

Parameter	Description
Database	This is the machine name of the database where the tables will be reloaded.
Log Recovery	If Log Recovery is 1 (NO), then no log recovery files (LMBRLGx) will be created, and a LISTLOG and APPLYLOG cannot be performed. If Log Recovery is 2 (YES), the input source must be a BACKUP or TRANSLATE. Th Restartcount option is not valid with RECOVERY=YES.
Input Type	Indicates that you wish to recover tables from either a BACKUP, TRANSLATE, or UNLOAD.
RESTART	Indicates that you wish to restart a failed RELOAD using the same input media and RELOAD table commands. RESTART will skip the input selection,

and RELOAD table commands creation panels, and pass directly to the RELOAD execution menu. You must indicate the same RELOAD ID that was used previously. You may also use the RESTARTCOUNT option to skip over records already reloaded. The RESTART option is not valid with RECOVERY=YES. The RESTARTCOUNT option is not valid unless RESTART is indicated.

RELOAD from an UNLOAD

If you selected to RELOAD and UNLOAD, and RESTART=NO, then you will be presented with the Unload Selection menu (Figure 133 on page 302). From this panel you will view the unload description file (UNLDDESC) or unload history file (UNLDHIST) to determine the timestamp that Data Restore generated for the UNLOAD that you wish to RELOAD.

From this panel, you will press **PF5** to review the appropriate history file to determine the DATE and TIME stamps of the UNLOAD that you want to use for RELOAD. For each UNLOAD that you perform, Control Center updates the UNLDHIST file with the media used and the timestamp generated by Data Restore for the UNLOAD. The timestamp for an UNLOAD is identified by an "@" symbol immediately preceding the timestamp. When you return to the menu, you will enter the DATE and TIME stamps, and proceed to the table list panel.

Press **PF6** to review the Unload UNLDDESC file which contains all the UNLOAD listing for the database. This file contains the same information that would be generated by the DESCRIBE function. The timestamp for an UNLOAD is identified by the XTS9-142 message. For example, `XTS9-142 Base SQLDB2 Date 09/08/98 Time 14:39:43`.

```
XTS9-100 Data Restore feature VERSION 6.1.0
XTS9-309 Processing DB2 for VSE and VM version 6
XTS9-160 External labeling of this unload is
XTS9-142 Base SQLDB2   Date 09/08/98 Time 14:39:43
XTS9-013 Table SQLDBA  .ACTIVITY           may be reloaded
```

Figure 132. Example of UNLOAD UNLDDESC (Describe) file entries

```
 mm/dd/yyyy                    CONTROL CENTER                        hh:mm:ss
*-------------------------  Unload Selection  ------------------------------*
|                                                          CTRLID:  SQLMDEMO |
|  Database ==> SQLDBA                                     NODE:    VMSYSTM1 |
|                                                          DRMACH:  DREST41  |
|                                                                            |
|                                                                            |
|  1)  Press PF5 to review the UNLOAD HISTORY file (UNLDHIST)                |
|  2)  Determine the DATE and TIME stamps for the UNLOAD you wish to         |
|      RELOAD.  The DATE and TIME stamps follow the VOLID/file entry         |
|      in the COMPLETION record line.                                        |
|  3)  Input the DATE and TIME stamps below.                                 |
|                      EXAMPLE UNLOAD HISTORY ENTRY                          |
|      UNLOAD.TEMP Beginning Series 200                                      |
|      UNLOAD filled, series 200 Volid VP001 400  - 08/02/97 16:00:00        |
|      UNLOAD.TEMP COMPLETED SERIES 200 - 08/02/97 16:00:00                  |
|                                                                            |
|      Timestamp   ==> __________  ( Date )  _________    ( Time )          |
|                                                                            |
|                                                                            |
*----------------------------------------------------------------DRRELD4------*
  PF1 HELP    PF3 QUIT    PF5 UNLOAD History     PF6 UNLOAD DESCRIBE
                                                 ENTER  Continue
```

Figure 133. Unload Selection Menu

Figure 134 is an example of an UNLOAD history file and its corresponding DATE and TIME stamp that must be entered on UNLOAD Selection Panel.

```
01/03/97 11:41:20 UNLOAD.TEMP Beginning Series 100
01/03/97 11:41:49 UNLOAD filled, series 100 Volid 35UNLO1 DB2VM510 * 400  @ 03/01/97 11:41:42
01/03/97 11:41:50 UNLOAD.TEMP COMPLETED SERIES 100 @ 03/01/97 11:41:42
```

Figure 134. Example UNLOAD History file with UNLOAD DATE and TIME stamps

Restart Selection

If you select RESTART = Yes, the next menu invoked is the Restart Selection menu. Here you will enter the same time and date stamps that were used for the original reload. Since the same RELDCTL file will be used again for the reload, you will not be presented with the Reload Table Command generation menu. Instead, you will be taken directly to the Reload Execution menu.

```
 mm/dd/yyyy                     CONTROL CENTER                        hh:mm:ss
*----------------------  Reload Restart Selection  --------------------------*
|                                                         CTRLID:  SQLMDEMO |
|  Database ==> SQLDBA                                    NODE:    VMSYSTM1 |
|                                                         DRMACH:  DREST41  |
|                                                                           |
|  1. Input the  SAME  DATE and TIME stamp that was used in                 |
|     the original reload control file (RELDCTL).  On the next panel        |
|     use the same RELOAD ID (filename) as before. CTRLCTR will use         |
|     the same file for input media and RELOAD commands.                    |
|                                                                           |
|  2. Input the DATE and TIME stamp below. Press Enter to continue.         |
|                       EXAMPLE UNLOAD HISTORY ENTRY                        |
|      UNLOAD.TEMP Beginning Series 200                                     |
|      UNLOAD filled, series 200 Volid VP001 400  - 08/02/97 16:00:00       |
|      UNLOAD.TEMP COMPLETED SERIES 200 - 08/02/97 16:00:00                 |
|                                                                           |
|      Timestamp   ==> __________  ( Date )  _________    ( Time )          |
|                                                                           |
|                                                                           |
*---------------------------------------------------------------DRRELD6------*
 PF1 HELP    PF3 QUIT    PF5 UNLOAD History     PF6 UNLOAD DESCRIBE
                                                ENTER  Continue
```

Figure 135. Reload Restart Selection Menu

RELOAD from a BACKUP, TRANSLATE and Log Recovery

If you select to RELOAD from a BACKUP or TRANSLATE, you will be lead through a process to perform a reload from either a BACKUP, INCREMENTAL BACKUP, or translated archive. When "Log Recovery" = "2" (yes), then logical units of work (LUWs) will be extracted from the selected log archives. After the RELOAD is complete, Control Center will automatically perform a LISTLOG. You can review the LISTLOG output from the LISTLOG function menu. The selected LUWs can later be re-executed on the database by the APPLYLOG function.

If you are reloading from an incremental backup, Data Restore will first reload the table records from the incremental backup and then reload the unchanged records from the reference backup.

The RELOAD setup process will be:

1. Select whether to view recovery sets for the latest or all the available recovery sets.
2. Review the database RESTORE report which lists the available recovery sets. An example of this report can be found in Figure 122 on page 280. You must remember the number of the restore set you wish to RELOAD. This restore set number will inputted in the next step. You should also note the type of backup and whether it was single or dual backup. If are performing a Log Recovery, you can select "Partial Log" = "Y" (YES) to limit the number of logs that will be processed.
3. On the select menu you will input the restore set number to RELOAD and whether to RELOAD from the primary or secondary backup.
4. You will then view the reload control file (RELDCTL). Verify that the file contains the tapes that should be used.
5. You will then be asked to press the 'PF3' key to cancel the RELOAD or press the ENTER key to continue the RELOAD.

6. You will then proceed to the "Table Reload Command Generation" menu.

Table RELOAD Command Generation

The next step is to create the list of RELOAD commands for each table that is to be reloaded. See Figure 136. Up to 90 table RELOAD statements can be listed in one SYSIN file. Each RELOAD command must include either the PURGE, NEW, ADD, or REPLACE option that is to be applied, and the appropriate TNAME, CREATOR, NEWTNAME, NEWCREATOR, DBSPACE OWNER and NAME parameters. RELOAD performance will be affected by the number and type of tables listed.

Note: If the selected tables are using Referential Integrity, then you must understand how it will affect reloading the tables. Data Restore does not reload tables in the order presented in the SYSIN file.

When you are finished entering the RELOAD table commands, press **PF5** to continue to the RELOAD Execution panel.

```
 mm/dd/yyyy                      CONTROL CENTER                        hh:mm:ss
*---------------------------- Reload Table List -----------------------------*
|  Database => SQLDBA                              Selected Tables => 0      |
|     P,N,A,R   Table                           Dbspace                      |
| Num  Funct    Creator   Table Name            Owner      Dbspace Name      |
| ---  -----    --------  ------------------    --------   ----------------- |
|       P       CREATOR_  TNAME_______________                               |
|          NEW  NEWCREAT  NEWTNAME__________  OWNER___   DBSPACE___________ |
|                                                                            |
|       _        ________  ____________________                             |
|          NEW  ________  ____________________  ________   ________________  |
|                                                                            |
|       _        ________  ____________________                             |
|          NEW  ________  ____________________  ________   ________________  |
|                                                                            |
|       _        ________  ____________________                             |
|          NEW  ________  ____________________  ________   ________________  |
|                                                                            |
|                                Page 1     of 1                             |
*----------------------------------------------------------------DRRELD3------*
PF1 Help         PF3 QUIT     PF5 Continue
PF7 Back         PF8 Forw
```

Figure 136. RELOAD Table List Panel

In the next panel (Figure 137 on page 305), you will decide whether to execute or schedule the RELOAD, and specify the RELOAD ID for the RELOAD, some additional SYSIN file options, and who to notify.

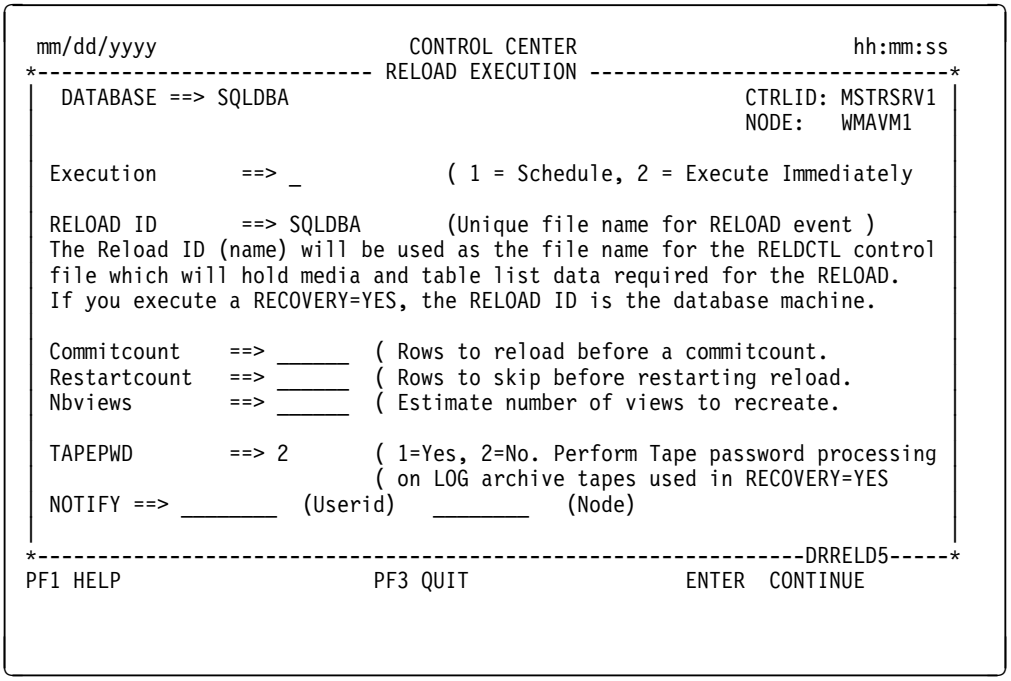


Figure 137. RELOAD Execution Panel

Reload Execution Parameters

Parameter	Description
Execution	Used to specify whether to (1) schedule for later execution, or (2) execute the unload immediately. If schedule is selected, you will be presented with the scheduling panel.
RELOAD ID	Unique name that will identify the reload. The name can be a maximum of 8 characters. If RECOVERY=YES, the RELOAD ID will be the database machine name.
Commitcount	Used to specify how many table records to reload before Data Restore will issue a commitcount.
Restartcount	Used to specify how many table rows will be skipped before reloading.
Nbviews	Data Restore assumes a maximum of 200 views may be created with the reload. If the number of views is greater than 200, indicate the number here.
TAPEPWD	Enter '1' (YES) to have Control Center automatically handle changing tape password authority on the LOG archive tapes before and after execution when RECOVERY=YES and the log archive media is tape. If you enter 'N' (NO), it is assumed that you have given the Data Restore machine the required authority to READ the

	database's log archive tapes, or the LOG archive media is DISK.
NOTIFY	Userid and NODE of person to notify when the RELOAD is successful.

Tape Passwords

While RELOAD does not use the database, it does use archive tapes that belong to the database. The Data Restore machine must have authority to read these archive tapes in order to perform the RELOAD.

If you use a tape management system, then your tape manager will have a password or tape authority process in place. If you use native CMS to manage your tapes, then the tape OPERATOR will control the tape security. During a database or log archive, passwords are not used when the database's tapes are first mounted. However, many tape managers have catalog commands that can be used to later add, change or delete passwords on a tape. Additionally, tape managers can have authority levels, such as the VMTAPE ANYTAPE, that allow a user to mount other user's tapes.

For VMTAPE users, you have several options that you can use. The database RELDCTL file that is created prior to execution of RELOAD lists the files or tapes to be used as input media for the RELOAD. If you choose 'TAPEPWD=Y' from the menu, then the SDRVTPWD EXEC will execute VMTAPE catalog commands on the database against the tapes listed in the database RELDCTL file to add a read password of 'ALL' to the tapes. This will allow anybody to read that tape without the need for a password. After the RELOAD, SDRVTPWD will be used to remove the password from the tapes.

If the archive in the control file was written to disk, or if you have no tape management system (CMS), then no automatic password processing will take place.

There are some restrictions with using the automatic password processing. SDRVTPWD only works with VMTAPE, but it can be modified so that catalog commands for other tape management systems can be used.

- The database does not have to be running, but the database manager must be logged on and disconnected in order for the CP SEND command to process.
- Because the catalog command is executed on the database as a 'CP' command, the database can be running when the catalog command is executed. However, if the catalog command fails, the RELOAD will be cancelled.

Chapter 25. Data Restore TRANSLATE

Overview

The Data Restore TRANSLATE function allows the user to translate a database archive into a format that can be used by Data Restore. The original archive is not altered. After the archive media is read twice, a Data Restore BACKUP is produced along with three work files, SYS001, DIRWORK, and HEADER; these files are required for the RELOAD function.

How the Translate Tool Works

Execution of TRANSLATE by Control Center is performed in two steps.

The first step is for the user to select an archive to be translated from a list of archive sets. Based on the archive chosen, Control Center creates a control file (database TRANSCTL) that will list the input archive file or tapes to be translated. Only the database archive will be translated; the log archives are not affected. This file will reside on Control Center's 191 disk.

You can only execute one TRANSLATE operation at a time for each database. One TRANSCTL file is created specifically for each TRANSLATE process. A new TRANSCTL file is created each time the TRANSLATE setup is run.

The second step occurs at execution time. Control Center will use the media chosen for TRANSLATE in the database PARMS file as the output for the translate process. The actual file or tapes used for translation will be pulled from the database TAPES file in the same series as the archive to be translated. The Translate_media in the database PARMS file must match the media listed in the database TAPES file. The Data Restore machine must have the authority to read the archive tapes or link to the database's archive disk. A SYSIN file will be created and FILEDEFS and LABELDEFS executed. Data Restore reads the archive twice. If the archive is on tape, it will request that the first tape of the archive be mounted again.Control Center will detach the tape and manage the mount of the first tape. The TRANSLATE will then continue.

After the translate, the timestamp of the TRANSLATE will be appended to the original archive record in the database ARCHHIST file. The database TRANSHIST file will record the file or tapes used in the TRANSLATE and the location of the work files.

Translate Files

The basic files created will be:

Filename	Description
dbmach TRANSCTL	The control file that identifies the input media of the archive to be translated. This file resides on the Control Center service machine.

dbmach TRANLIST	The output listing of the TRANSLATE created by Data Restore. It is sent to the Control Center service machine's 191 disk.
dbmach TRANSYIN	The SYSIN file created on the Data Restore machine to execute the TRANSLATE.
dbmach TRANSLOG	A log of the current TRANSLATE. The file is maintained on the Control Center service machine.

Input and Output Media Processing

The input media (archive) is determined by whether the archive represented in the database TRANSCTL file is on tape or disk. The output media (translate) is determined by the translate values in the database PARMS file. The Translate_media value in the PARMS file (TAPE/DISK) must match the type of media indicated in the database TAPES file.

Tape Processing

Information about output media for TRANSLATE tapes and work files is kept in the database TAPES file in the same series as the associated archive. One TRANSLATE is allowed per database TAPES file series. The file or tapes to be used for the TRANSLATE are indicated by using a type of TRANS.

While there may be one or more tapes designated for the TRANSLATE (TRANS), only one entry (TRANSDSK) will be used to indicate where the work files will be kept.

The three work files created by Data Restore (SYS0001, DIRWORK, HEADER) can only be written to disk; these files will all be written to the same disk. Because all the work files are written to the same disk, only one entry in the database TAPES file (TRANSDSK) will be used to indicate the name and location of the work files. The TRANSDSK entry includes the FILENAME and link address of the work files. Only the file name will be significant for the TRANSDSK entry. The actual filetype used will correspond to the type work file that will be FILEDEFed and LABELDEFed. The file mode will be determined at the time of the TRANSLATE. This strategy will allow more than one TRANSLATE work file output on a disk. See Figure 139 on page 309 for an example of the filedefs specified.

TRANSLATE may require more tapes than those used in the archive. If predefined tapes are used in the database TAPES file, be sure to include more tapes to handle the TRANS entry.

```
100  TRANS    00000  00:00:00 UNUSED SCRATCH
100  TRANSDSK 00000  00:00:00 UNUSED DB2VM510 TRANSDSK *  400
```

Figure 138. Example of TAPES file entries for TRANSLATE

The preceding figure shows how the TRANSDSK entry in the database TAPES file will be used when performing a FILEDEF on the translate work files.

```
FILEDEF SYS0001 DISK  DB2VM510 SYS0001 T ( RECFM F BLOCK 4096
FILEDEF HEADER  DISK  DB2VM510 HEADER  T ( RECFM F BLOCK 4096
FILEDEF DIRWORK DISK  DB2VM510 DIRWORK T ( RECFM F BLOCK 0512
```

Figure 139. Example FILEDEFS for TRANSDSK entry

The TRANSLATE (new backup) tapes/file will use cuu 185, and the archive tapes (ARIARCH) will use cuu 181.

Because the TRANSLATE output tapes are taken from the same series as the archive being translated, the database TAPES file must have a series that matches the series in the ARCHHIST file. For example, if you have a series 100 through 600 and you delete series 600 from the tapes fie, you may still have an entry for series 600 in the ARCHHIST file. You can TRANSLATE the archive for the deleted series 600 because the input tapes still exist, but there will be no entry in the TAPES file to indicate what output tapes to use.

How to Invoke TRANSLATE

To invoke the TRANSLATE function, choose Option **T** from the Data Restore Menu (Figure 115 on page 263).

The first entry panel of the TRANSLATE tool is shown in Figure 140.

```
 mm/dd/yyyy                    CONTROL CENTER                        hh:mm:ss
*---------------------------- Translate Selection ----------------------------*
| Command ==>                                                CTRLID: SQMSTDV1  |
|    Database ===> DB2VM510                                  NODE:   WMAVM1    |
|                                                            DRMACH: SQMRESTR  |
|                                                                              |
|    Archive Set ===> ALL Archive Translate Set to display.                    |
|                              ( Blank for LATEST or ALL for all available )   |
|         To TRANSLATE a database archive you must view the archive set        |
|    and select the archive translate set to use.  The archive                 |
|    set is a group of tapes that were used for a previous database            |
|    archive.  You may choose ALL to display all available archive             |
|    sets, or you may leave the translate set BLANK to retrieve the LATEST     |
|    archive set.                                                              |
|                                                                              |
|    The selected translate set will be displayed and you will be asked to     |
|    enter the translate set number to begin the TRANSLATE.                    |
|                                                                              |
|        Enter Translate Set, press ENTER to process, or                       |
|        press PF3 to QUIT                                                     |
*------------------------------------------------------------------------------*
```

Figure 140. Translation Selection Panel

You will then be presented with a Recovery Set report that is the same type of report you would see to select an archive set for recovery. Review the report and note the archive set number you wish to translate. On the next menu you will enter the archive set you have chosen.

The Translation Set Selection panel of the TRANSLATE tool is shown in Figure 141 on page 310.

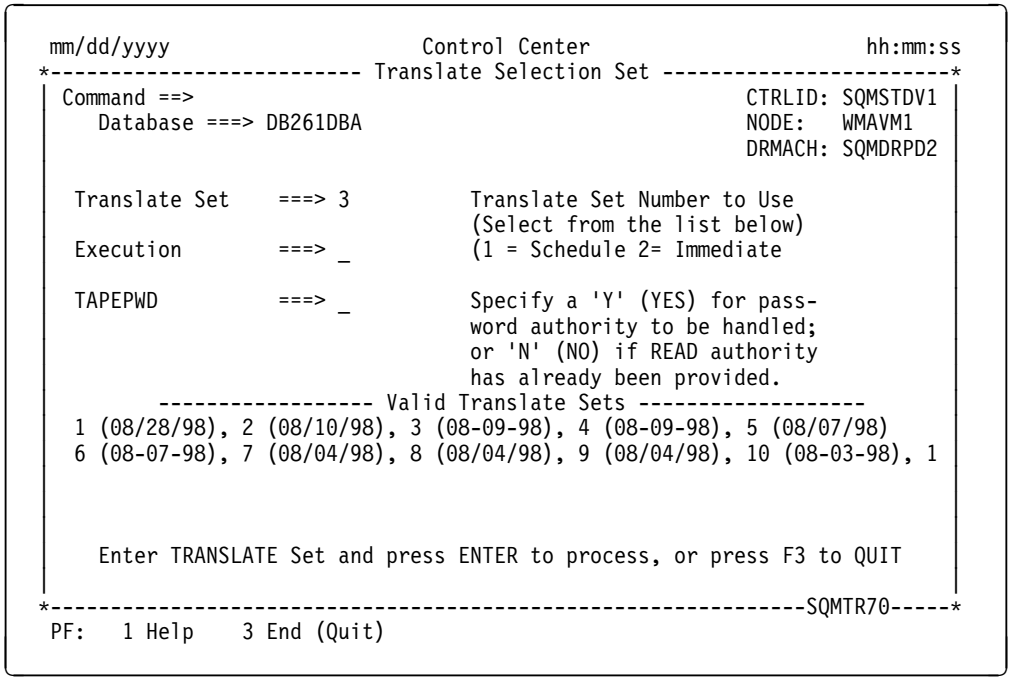


Figure 141. TRANSLATE Set Selection Panel

Parameter	Description
Translate Set	Enter the number of the archive set you wish to translate.
Execution	Enter "1" to schedule the translate, or "2" to execute it immediately.
TAPEPWD	Enter Y (YES) to have the Control Center attempt to automatically handle changing tape password authority before and after it executes the translate. If you enter N (NO), it is assumed that you have given the Data Restore machine the required authority to READ the database's archive tapes.

Tape Passwords

While TRANSLATE does not use the database, it does use archive tapes that belong to the database. The Data Restore machine must have authority to read these archive tapes in order to perform the TRANSLATE.

If you use a tape management system, then your tape manager will have a password or tape authority process in place. If you use native CMS to manage your tapes, then the tape OPERATOR will control the tape security. During database or log archives, passwords are not used when the database's tapes are first mounted. However, many tape managers have catalog commands that can be used to later add, change or delete passwords on a tape. Additionally, tape managers can have authority levels, such as the VMTAPE ANYTAPE, that allow a user to mount other user's tapes.

For VMTAPE users, you have several options that you can use. The database TRANSCTL file that is created prior to execution of TRANSLATE lists the archive tapes that will be translated. If you choose TAPEPWD=Y from the menu, then the SDRVTPWD EXEC will execute VMTAPE catalog commands on the database against the tapes listed in the database TRANSCTL file to add a read password of ALL to the tapes. This will allow anybody to read that tape without the need for a password. After the TRANSLATE, SDRVTPWD will be used to remove the password from the tapes.

If the archive in the control file was written to disk, or if you have no tape management system (CMS), then no automatic password processing will take place.

There are some restrictions with using the automatic password processing. SDRVTPWD only works with VMTAPE, but it can be modified so that catalog commands for other tape management systems can be used.

- The database does not have to be running, but the database manager must be logged on and disconnected in order for the CP SEND command to process.
- Because the catalog command is executed on the database as a CP command, the database can be running when the catalog command is executed. However, if the catalog command fails, the translate will be cancelled.

After the Translate

The FULL ARCHIVE completed entry for the original archive in the database ARCHHIST file, for the archive being translated, will be updated with the timestamp of the translate. The ARCHHIST file will be read backwards until the series, date and time are matched. This means that if a matching recovery series is found, the archive recovery entry will be marked with the TRANSLATE timestamp.

The database TRANHIST file will be updated with the file or tapes that were produced for the translate and the timestamp of the TRANSLATE. The location of the SYS0001, DIRWORK and HEADER files will be represented in the TRANHIST file by one TRANSDSK entry which will look like the TRANSDSK entry in the database TAPES file.

If you translate an archive that has already been translated, the old translate tag in the ARCHHIST file will be overwritten with the new translate tag.

The original archive in the ARCHHIST file and the new backups in the TRANHIST file are now mapped to each other by the timestamp, and the new backup can be later used as input for a RESTORE, RELOAD or RELOAD RECOVERY.

The database TRANLIST file (SYSPRINT) will be sent to the Control Center service machine and administrators listed in the PARMS file will be notified.

If automatic tape password processing was selected, then the Control Center will attempt to remove the ALL password from the archive tapes.

Chapter 26. Data Restore LISTLOG

Overview

The Data Restore LISTLOG function produces a report of the Logical Units of Work (LUWs) contained in the database log archives and active log for all the tables that were listed in the SYSIN file for a RELOAD with RECOVERY = YES. The database administrator will use the LISTLOG report to decide which LUWs will be re-applied to the database during the APPLYLOG function.

Control Center provides the user interface to Data Restore's LISTLOG where this function can be executed against the database's current RELOAD work files on the Data Restore machine's 191 disk. Normally, Control Center will execute a LISTLOG as part of the RELOAD RECOVERY=YES operation.

How LISTLOG Works

During the RELOAD RECOVERY=YES function, Data Restore produces work files to record all LUWs found for the tables listed in the RELOAD SYSIN file. These files are kept on the Data Restore machine's 191 disk. The LISTLOG function will be executed, and the LISTLOG report will be sent to the Control Center service machine and the DBA listed in the NOTIFY option.

Control Center requires that LISTLOG can only be executed against the current RELOAD work files.

The LISTLOG function will stop reporting table LUWs when it encounters a DROP statement for any of the affected tables. This includes any DROP statements that are used during DBSPACE reorganizations. NO LUWs can be recovered after a TABLE DROP statement. Also, if tables are using referential integrity, the log records will include the original SQL statement, plus any cascade response required to maintain referential integrity. During APPLYLOG, all the LUWS will be recreated by Data Restore. If referential integrity is active during the APPLYLOG, you may get unexpected results.

How to Invoke

To invoke LISTLOG, the DBA will select Option **LL** from the Data Restore Menu for immediate or scheduled execution.

At execution time, the Data Restore machine will perform necessary FILEDEFS and LABELDEFS, create the SYSIN file, and execute the LISTLOG routine (XTS91001 MODULE). The LISTLOG SYSPRINT (database LISTLOG) will be sent to the Control Center service machine and the DBA.

The LISTLOG will later be used to determine LUWS to recover during the APPLYLOG function.

LISTLOG Control and OUTPUT files

The basic files created will be:

Filename	Description
dbmach LISTSYIN	The SYSIN file created on the Data Restore machine to execute the LISTLOG. This file contains the Data Restore commands and options to be executed, and additional information provided by Control Center.
dbmach LISTLOG	The output listing of the LISTLOG created by Data Restore. This file will list all the LUWS recorded in the work files for the current RELOAD RECOVERY= YES.
LISTLOG ERROR	If errors occur before, during, or after execution of the function on the Data Restore machine, the LISTLOG ERROR file will indicate the error type. If an error condition was detected during Control Center processing, then the name of the module and the approximate line number of the error will be presented. The ERROR file is sent to all database administrators listed in the specific database's PARMS file.

Input and Output Media Processing

No tapes file processing is required. LISTLOG will use the work files for the database on the Data Restore machine's 191 disk as input.

Passwords

LISTLOG requires that the database password for user SQLDBA be included in the 'DBAPW=' SYSIN file control statement.

A password file (drmach LINKPWDS) file is required. Refer to “Password File Setup for the Data Restore machine” on page 504 for instructions on how to implement Data Restore password support.

How to Invoke LISTLOG

The first entry panel of the LISTLOG tool is shown in Figure 142 on page 315. Here you will specify immediate or scheduled LISTLOG execution.

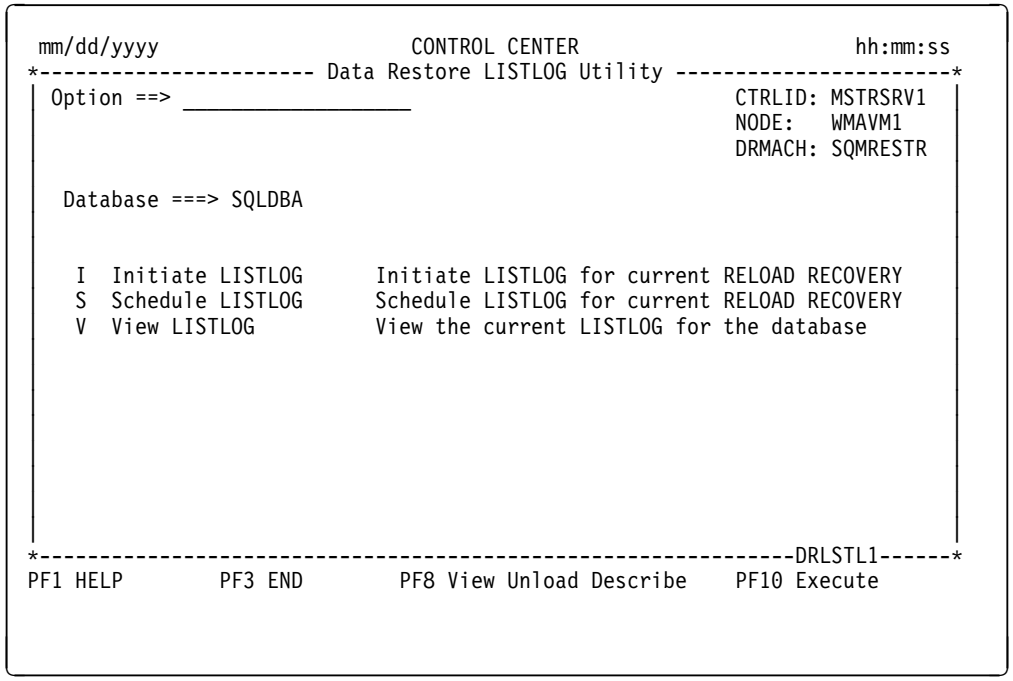


Figure 142. Data Restore LISTLOG Utility Menu

LISTLOG Job Descriptions

Parameter	Description
I	Execute the LISTLOG immediately.
S	Schedule the LISTLOG function for a later time. You will be presented with a job scheduling panel.
V	View the current LISTLOG report. If your RELOAD RECOVERY=YES completed successfully, then the current LISTLOG report can be used for the APPLYLOG function.

Chapter 27. Data Restore APPLYLOG

Overview

The Data Restore APPLYLOG function allows the user to recover the Logical Units of Work (LUWs) contained in the database LISTLOG report. A successful RELOAD RECOVERY=YES and LISTLOG must be performed prior to executing APPLYLOG. The database must be up and running when the APPLYLOG executes.

How APPLYLOG Works

During the RELOAD RECOVERY=YES function, Data Restore produced work files to record all LUWs found for the tables listed in the RELOAD SYSIN file. These files are kept on the Data Restore machine's 191 disk. If the RELOAD RECOVERY=YES was successful, the LISTLOG function will automatically be executed, and the LISTLOG report sent to the Control Center service machine and the DBA. LISTLOG can also be invoked using Control Center.

Control Center requires that LISTLOG and APPLYLOG can only be executed against the current RELOAD work files.

The LISTLOG identifies all the LUWs recorded in the database log archives and active log for the tables that were listed in the SYSIN file for a RELOAD with RECOVERY=YES.

The LISTLOG function will stop reporting table LUWs when it encounters a DROP statement for any of the affected tables. This includes DROP statements that are used during DBSPACE reorganizations. You will not be allowed to recovery any LUWs after a TABLE DROP statement. Also, if your tables are using referential integrity, the log records will include the original SQL statement plus any cascade response required to maintain referential integrity. During APPLYLOG, all the LUWS will be recreated by the Data Restore product. If referential integrity is active during the APPLYLOG, you may get unexpected results.

APPLYLOG requires that the SYSIN file include the 'END=' command statement to indicate the timestamp for the LUW where the recovery should stop. The LUW referenced in the SYSIN file will **NOT** be executed. The database administrator will use the LISTLOG report to decide which LUWs need to be re-applied during the APPLYLOG function, and indicate the 'END=' timestamp on the Control Center APPLYLOG panel.

To use APPLYLOG, the DBA will invoke the Control Center APPLYLOG menu interface.

At execution time, the Data Restore machine will perform necessary FILEDEFS and LABELDEFS, create the SYSIN file, and execute the APPLYLOG routine (XTS91001 MODULE). The APPLYLOG SYSPRINT (database APLYLIST) will be sent to the Control Center service machine and the DBA.

APPLYLOG Control and OUTPUT files

The basic files created will be:

Filename	Description
dbmach APLYSYIN	The SYSIN file created on the Data Restore machine to execute the APPLYLOG. This file contains the Data Restore commands and options to be executed, and additional information provided by Control Center.
dbmach APLYLIST	The output listing of the APPLYLOG created by Data Restore
APPLYLOG ERROR	If errors occur before, during or after execution of the function on the Data Restore machine, the APPLYLOG ERROR file will indicate the error type. If an error condition was detected during Control Center processing, then the name of the module and the approximate line number of the error will be presented. The ERROR file will be sent to all database administrators listed in the specific database's PARMS file.

Input and Output Media Processing

No tapes file processing is required. Data Restore will use the database work files found on the Data Restore machine's 191 disk as input.

Passwords

APPLYLOG requires that the database password for user SQLDBA be included in the 'DBAPW=' SYSIN file control statement.

A password file (drmach LINKPWDS) file is required. Refer to “Password File Setup for the Data Restore machine” on page 504, for instructions on how to implement Data Restore password support.

How to Invoke APPLYLOG

To invoke APPLYLOG, select Option **AL** from the Data Restore Menu (Figure 115 on page 263).

The first entry panel of the APPLYLOG tool is shown in Figure 143 on page 319.

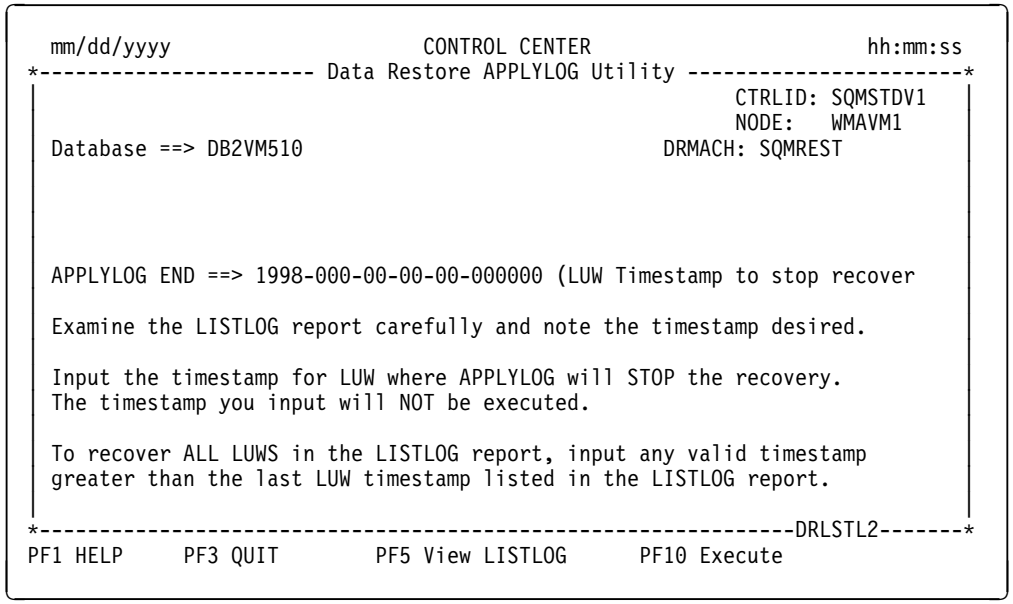


Figure 143. Data Restore Applylog Utility Menu

Parameter	Description
APPLYLOG END	Indicates the timestamp for the LUW in the LISTLOG report where the recovery should STOP. The LUW indicated by the timestamp will *NOT* be executed.

Chapter 28. Single User Mode Tools

Overview

The single user mode tools are accessible from the Database Utility Functions Panel shown in Figure 144. The single user mode tools are identified as "Single-User" types on the Database Utility Functions panel. Each of these types of tools will be covered in this chapter with the exception of the Single User Mode DBSPACE Reorganization tool (refer to Chapter 30, “DBSPACE Reorganization Tools” on page 367).

```
 mm/dd/yyyy                    CONTROL CENTER                        hh:mm:ss
*------------------------ Database Utility Functions -----------------------*
| Option ===>                                                CTRLID: MSTRSRV1 |
|    Database ===> SQMDBA                                     NODE:  VMSYSTM1 |
|  **FUNCTION**              **TYPE**        *****DESCRIPTION*****           |
|  E  ADD DBEXTENTS          Single-User    SQLADBEX utility                 |
|  C  COPY/MOVE DBEXTENTS    Single-User    SQLCDBEX utility                 |
|  D  DELETE DBEXTENTS       Single-User    SQLADBEX DELETE                  |
|  A  ADD DBSPACES           Single-User    SQLADBSP utility                 |
|  CL COLDLOG                Single-User    COLDLOG utility (SQLLOG)         |
|  RC REORG CATALOG INDEXES  Single-User    SQLCIREO utility                 |
|  RS RELOAD SINGLE USER     Single-User    SUM dbspace reload only          |
|  SU REORG SINGLE USER      Single-User    SUM reorg entirely on DBmachine  |
|  R  SQLREORG               Multi-User     DBSPACE reorganization           |
|  T  SQLTABLE               Multi-User     Table reorgs/manipulation        |
|  M  SQLMAINT               Multi-User     Automated dbspace maintenance    |
|  B  SQLRBIND               Multi-User     Rebind Packages                  |
|  RI REORG INDEX TOOL       Multi-User     DBSPACE Index Maintenance Tool   |
|  RD REORG DRIVER TOOL      Multi-User     SQLREORG Driver by DBSPACE List  |
|  SL SEARCH/LIST            Multi-User     Search and List DBSPACES/TABLES  |
|                                                                           |
*-------------------------------------------------------------SQMDUTIL----*
 PF:   1 Help    3 End    8 Forward (tools continued)
```

Figure 144. Database Utilities Panel

Before continuing, you should be familiar with the material presented in:

- Chapter 2, “Architecture” on page 13,
- Chapter 18, “Database Startup and Termination Tools” on page 213, and the
- *DB2 Server for VM System Administration* manual.

How the Single User Mode Tools Work

These tools are used to invoke and manage utilities that run on database virtual machines in single user mode (SUM). By using these tools you can fully automate your database's single user mode activities.

When the database utilities run, they ask many processing-related questions. For example, as the SQLADBEX (add DBEXTENT) utility runs, it prompts for the virtual addresses of the dbextents (minidisks) to be added.

Control Center provides this information to the SQLADBEX utility as it is requested. The information used is provided by you through its single user mode tool panels.

Each of its single user mode tools can be invoked to run immediately or scheduled for later execution. When a SUM tool is invoked, the database machine is terminated and the appropriate single user mode utility is started. During the processing of the utility the database is unavailable. Upon completion, the database will be returned to its original starting status (up or down).

Single User Mode Log File

The single user mode tools create log files containing the database console messages produced during the specific SUM activity. Log files are stored on Control Center's 191 A-disk, with filenames matching the database's name and filetypes that indicate the type of activity. For example, the database console messages produced during an add dbextent (ADBEX) job for database SQLDBA would be saved in log file SQLDBA ADBEXLOG. Each specific log file created by these tools will be outlined in this chapter.

Single User Mode History Files

The single user mode tools maintain individual history files recording the date, time, and other relevant information pertaining to these types of activities. The history files are stored on the its 191 A-disk, with filenames matching the database's name and filetypes that indicate the type of activity. For example, the history information for all add dbextent activities for database SQLDBA would be found in history file SQLDBA ADBEXHIS.

Who Can Use Single User Mode Tools

Use of the single user mode tools requires Control Center database Administration-level or greater authorization (refer to “Authorization” on page 26).

Display of log and history files requires Control Center user-level or greater authorization.

Operational Note: Control Center provides a default set of tools that are accessible by its database User, Operator, or Administration-level authorizations. These defaults can be changed so as to add or delete Control Center database User, Operator, or Administration authorizations to specific tools. For more details on how your authorization levels can be modified, refer to Appendix E, “Authorizations” on page 511.

Add DBEXTENTS Tool

The Add DBEXTENTS tool automates the execution of the SQLADBEX utility. Use this tool when you need to add new dbextents (minidisks) to a database.

Before beginning, you should review and understand all information regarding the adding of new dbextents to a database as outlined in the *DB2 Server for VM System Administration* manual.

Note: All new minidisks for dbextents to be added must be added to the database virtual machine's VM directory prior to the running of the dbextent add process.

How the Add DBEXTENTS Tool Works

To start the Add DBEXTENTS tool, use the panel interface to provide the virtual addresses and storage pool numbers (refer to the *DB2 Server for VM System Administration* manual) for each new dbextent (minidisk) to be added.

Once the information has been specified, you can initiate the process immediately or schedule it to run later. When the process runs, the database machine will be terminated and Control Center will start the SQLADBEX utility on the database virtual machine. While the SQLADBEX utility runs, Control Center provides all information that is requested.

Upon completion of the SQLADBEX utility, the database will be returned to its original status (either up or down).

Add DBEXTENTS Tool Entry Panel

Shown in Figure 145 on page 324 is the Add DBEXTENTS entry panel used to specify dbextents to be added to a database. The panel consists of two sections: New DBEXTENTS and New NON-RECOVERABLE Storage Pools.

New DBEXTENTS

Specify in this section of the panel all new dbextents to be added to the specified database. If a storage pool specified in this section is new and is *not* specified in the New NON-RECOVERABLE section, then it will be defined to the database as a "recoverable" storage pool. The entry for storage pool 7 in Figure 145 on page 324 shows an example of how a new "recoverable" storage pool is specified.

New NON-RECOVERABLE Storage Pools

If a storage pool is new and is specified in this section, then it will be defined to the database as a "non-recoverable" storage pool. Storage pool 9 in Figure 145 on page 324 shows an example of how a new "non-recoverable" storage pool is specified.

Existing Storage Pools

If a storage pool is not new, specify the pool number in the New DBEXTENTS section of the panel and *not* in the New NON-RECOVERABLE section.

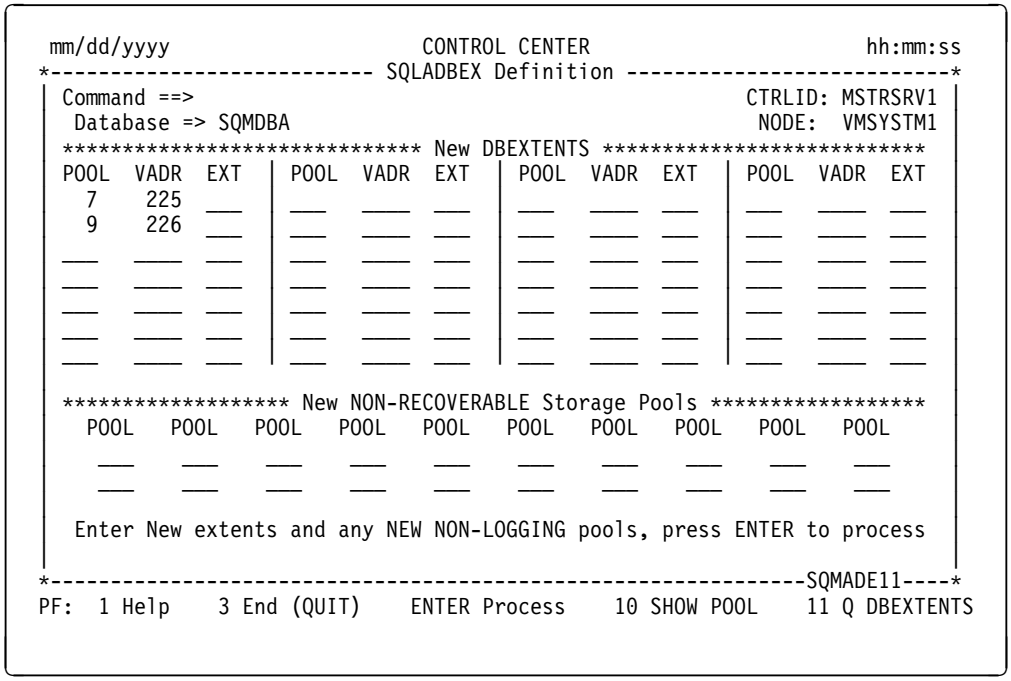


Figure 145. Control Center Add DBEXTENTS Definition Entry Panel

POOL, VADR, and EXT

The pool and virtual address (VADR) entry fields must be specified for all DBEXTENTS to be added. There are 28 POOL/VADR/EXT entry fields available within the New DBEXTENTS section of the panel. This limits to 28 as the maximum number of DBEXTENTS that may be added by Control Center in a single execution of the Add DBEXTENTS tool (see note below).

The VADR entry fields must match the virtual addresses for the new minidisks as defined in the VM directory for the database's virtual machine. The POOL entry field entered for each minidisk may be either an existing storage pool or a new storage pool. If an existing storage pool is specified, the minidisk will be added as another DBEXTENT in the storage pool.

When new storage pools are defined, you may optionally designate that they should be non-recoverable by listing them in the lower section of the panel.

Important:

Only list pool numbers in the lower section of the data entry panel if you want them to be NON-RECOVERABLE (non-logging).

Add DBEXTENTS Control Files

Two files are created based on your entries on the Add DBEXTENTS panel. These files are used by Control Center during the execution of the SQLADBEX utility on the database machine. These control files are identified by database and have filetypes of ADEXTENT and ADNLPOOL. Control Center will invoke the SQLADBEX utility to start the add dbextents process. When the SQLADBEX utility

prompts for what is to be done, Control Center uses the information contained in these two files.

Note:

The Add DBEXTENTS tool uses a single set of control files for each database machine. You cannot, therefore, schedule multiple add dbextents jobs for the same database because each new job will overlay the previously created control file set for that database.

Add DBEXTENTS Log File

A log file is created that contains all the database console messages produced while the SQLADBEX utility runs. This log file is kept for each database on Control Center's 191 A-disk. The file is identified by a filename matching the database virtual machine and a filetype of ADBEXLOG.

Add DBEXTENTS History File

In addition to the log file, Control Center maintains for each database a historical record of its Add Dbextents activity. This file is kept on its 191 A-disk, and is identified by a filetype of ADBEXHIS (the filename matches the database virtual machine name).

Note: This file is also used to record deleted dbextent information.

Example Add DBEXTENTS History File

Shown in Figure 146 is an example of an ADBEXHIS file which provides information about all dbextents that have been added to a database. This example shows that pools 7 and 8 were added on 03/16/97 at approximately half an hour past local midnight.

```
01/02/97 00:06:49 ADDED (POOL, VADR) 6 222
02/20/97 01:34:21 ADDED (POOL, VADR) 5 224
03/16/97 00:38:26 ADDED (POOL, VADR) 7 225
03/16/97 00:38:26 ADDED (POOL, VADR) 8 226
```

Figure 146. Example Add DBEXTENTS History File (ADBEXHIS)

Delete DBEXTENTS Tool

The Delete DBEXTENTS tool automates the execution of the SQLADBEX utility's delete processing. Use this tool when deleting dbextents from a database.

There are many technical considerations and associated precautions involved when deleting a dbextent. Review the sections on deleting a dbextent in the *DB2 Server for VM System Administration* manual.

Before starting, you must determine the dbextent number(s) for the dbextents that you want to delete. The numbers can be determined by reviewing the report generated by the Query DBEXTENTS/STORPOOL Mapping tool prior to using this tool or by using PF11 from this tool's panel. Only those dbextents that are empty (No. Pages In Use = 0) or those that exist within a storage pool with "enough" free

pages in other dbextents may be deleted; "enough" is defined as a number large enough to hold any data that is still in the dbextent to be deleted.

How the Delete DBEXTENTS Tool Works

To start the Delete DBEXTENTS tool, you use the panel interface to provide the dbextent numbers for the dbextents that you want deleted.

Once the information has been specified, you can initiate the process immediately or schedule it to run later. When the process runs, the database machine will be terminated and Control Center will start the SQLADBEX utility on the database virtual machine. While the SQLADBEX utility runs Control Center provides all information that is requested.

Upon completion of the SQLADBEX utility, the database is returned to its original status (up or down).

About the Delete DBEXTENTS Entry Panel

Shown in Figure 147 is the entry panel provided for deleting dbextents. The panel provides for a maximum of ten (10) dbextents to be deleted by a single Delete DBEXTENTS job.

```
 mm/dd/yyyy                    CONTROL CENTER                       hh:mm:ss
*----------------------- Delete DBEXTENT Selection ------------------------*
| Command ==>                                                CTRLID: MSTRSRV1 |
|  Database => SQLDBA                                         NODE:  VMSYSTM1 |
| ************************** Delete DBEXTENTS ************************** |
|                                                                           |
|     Dbextent Number to DELETE ==> 57                                      |
|     Dbextent Number to DELETE ==> 58                                      |
|     Dbextent Number to DELETE ==> ____                                    |
|     Dbextent Number to DELETE ==> ____                                    |
|     Dbextent Number to DELETE ==> ____                                    |
|     Dbextent Number to DELETE ==> ____                                    |
|     Dbextent Number to DELETE ==> ____                                    |
|     Dbextent Number to DELETE ==> ____                                    |
|     Dbextent Number to DELETE ==> ____                                    |
|     Dbextent Number to DELETE ==> ____                                    |
|                                                                           |
|                                                                           |
|                                                                           |
|       Enter Extent Number(s) to DELETE and press ENTER to PROCESS         |
|                                                                           |
*-------------------------------------------------------------SQMDDE10----*
PF:  1 Help    3 End (QUIT)    ENTER Process     10 SHOW POOL   11 Q DBEXTENTS
```

Figure 147. Control Center Delete DBEXTENTS Entry Panel

Entry Field	Description
Dbextent Number to DELETE	Specify dbextents to be deleted in the entry fields. Figure 147 shows two dbextents (57 and 58) that are to be deleted.

PF Key Selections

PF Key	Description
SHOW POOL (PF10)	Issues a SHOW POOL command to the database and displays the response (refer to Chapter 16, “Database Operation Interface Tools” on page 197). You can use this information to review a database's current dbextent space usage.
Q DBEXTENTS (PF11)	Query DBEXTENTS provides a detailed report of a database's dbextents, including a mapping to current virtual addresses, volids, and storage pools. Additional information includes a list of any deleted dbextents, the number of cylinders and pages allocated, and real device addresses. The report is presented in both dbextent and storage pool order. Refer to Chapter 37, “Control Center Administration Tools” on page 485.

Delete DBEXTENTS Control File

A control file is created based on inputs to the Delete DBEXTENTS panel. This file will be used by Control Center during the execution of the SQLADBEX utility on the database machine. The control file is kept on Control Center's 191 A-disk and is identified by a filetype of DBDELEXT (filename matches the database virtual machine name).

Note:

The Delete DBEXTENTS tool uses a single control file for each database machine. You cannot, therefore, schedule multiple Delete DBEXTENTS jobs for the same database because each new job (scheduled or immediate invocation) will overlay the previously created control file.

Delete DBEXTENTS Log File

Control Center creates a log file that contains all the database console messages produced while the SQLADBEX utility runs. This log file is kept for each database on its 191 A-disk. The file is identified by a filename matching the database virtual machine name and a filetype of DDBEXLOG.

Delete DBEXTENTS History File

In addition to the log file, Control Center maintains for each database a historical record of its Delete DBEXTENTS activity. This file is kept on the Control Center 191 A-disk, and is identified by a filetype of ADBEXHIS (filename matches the database virtual machine name).

Note: This file is also used to record add dbextents information.

Example Delete DBEXTENTS History File

Shown in Figure 148 is an example of an ADBEXHIS file which provides information about all dbextents that have been added and deleted from a database. This example shows that dbextents 57 and 58 were deleted on 03/16/97 at about 1:30 in the afternoon.

```
12/14/95 00:06:49 ADDED (POOL, VADR) 6 222
01/20/97 01:24:16 ADDED (POOL, VADR) 7 223
01/28/97 01:34:21 ADDED (POOL, VADR) 8 224
02/20/97 00:38:26 ADDED (POOL, VADR) 7 225
02/24/97 00:38:26 ADDED (POOL, VADR) 9 226
03/16/97 13:29:54 DELETED (EXTENT) 57
03/16/97 13:29:54 DELETED (EXTENT) 58
```

Figure 148. Example Delete DBEXTENTS History File (ADBEXHIS)

Copy/Move DBEXTENTS Tools

Use these tools when it is necessary to move any of your database's dbextents (directory, log, or data) from one minidisk to another.

Terminology used in this Section: The terms "dbextent" and "minidisk" are used synonymously throughout the “Copy/Move DBEXTENTS Tools” section.

Prior to copying any of your database's dbextents, you should review and be familiar with the SQLCDBEX utility information outlined in the *DB2 Server for VM System Administration* manual.

How the Copy/Move DBEXTENTS Tools Work

When these tools run, you are prompted by the Copy/Move DBEXTENTS panels for dbextent information. After indicating which dbextents are to be copied, you then have the option to invoke the tool immediately or to schedule a job for later execution.

When a Copy/Move tool runs, the database will be terminated (if it is not already terminated) and Control Center will invoke and manage the execution of the SQLCDBEX utility. As the utility prompts for inputs, Control Center will use the information that you provided.

Upon completion of the Copy/Move process, the database will be returned to its original status (up or down).

About the Copy/Move DBEXTENTS Options Panel

The Copy/Move DBEXTENTS options panel is shown in Figure 149 on page 329.

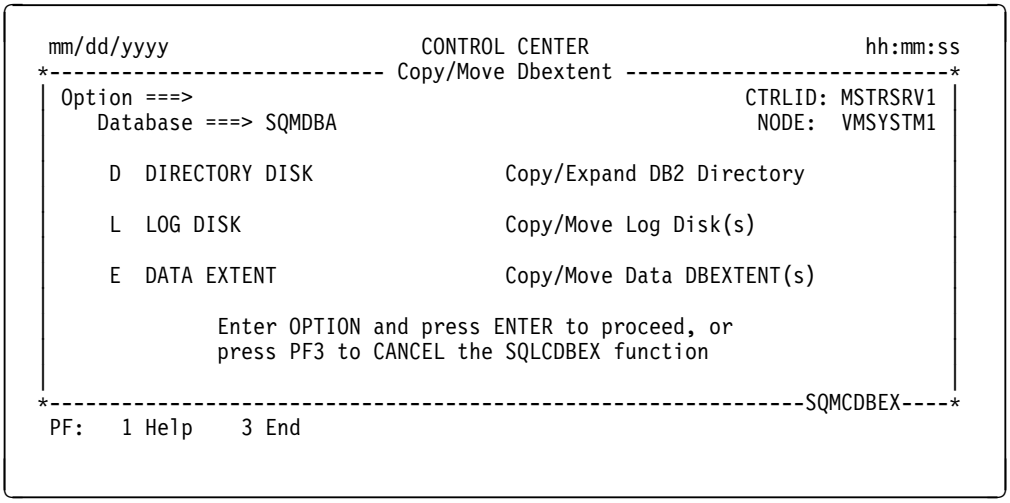


Figure 149. Copy/Move DBEXTENTS Options Panel

Options

Option	Description
DIRECTORY DISK (D)	Invokes the Copy/Expand Directory tool to copy your database's Directory (BDISK) minidisk to a minidisk of equal or larger (expanded) size.
LOG DISK (L)	Invokes the Copy/Move Log Disk tool to copy the database log (LOGDSK1 or LOGDSK2) minidisks to new minidisks of equal size.
DATA EXTENT (E)	Invokes the Copy/Move Data Disk tool to copy the database data (DDSKn) minidisks to new minidisks of equal size.

Note about Minidisk Sizes: Only the directory minidisk can be copied to a larger minidisk using these tools. The log and data minidisks *must* be copied to minidisks of the same size.

Copy/Move Log Files

When a Copy/Move tool is run, Control Center creates a file of the database console messages produced during the running of the SQLCDBEX utility. This file is kept on Control Center's 191 A-disk and is identified by a filetype of CDBEXLOG (filename matches the database virtual machine name).

Note: The same log file (database CDBEXLOG) is used by each of the Copy/Move DBEXTENTS tools. Therefore, each time a Copy/Move tool is run, a new log is generated, replacing the previously generated log.

Copy/Move History Files

When a Copy/Move tool is run, Control Center updates a history file recording information about the dbextent copied. A separate history file is kept for each database on its 191 A-disk and is identified by a filetype of CDBEXHIS (filename matches the database virtual machine name).

Shown in Figure 150 is an example database Copy/Move DBEXTENTS history file. In this example, the database directory was copied on 03/15/97 to a new virtual address of 207 (N means the new minidisk size is not greater than the original directory disk and 512 specifies the blocking). On 03/15/97, one of the database's log disks was copied to a new minidisk at virtual address 200, and on 03/16/97 dbextent number 5 (DDSK5) was copied to a minidisk at virtual address 205.

```
03/15/97 11:28:26 COPIED: BDISK 207 N 512
03/15/97 11:38:10 COPIED: LOGDSK2 200
03/16/97 11:38:10 COPIED: DDSK5 205
```

Figure 150. Example Copy/Move History File (CDBEXHIS)

Copy/Expand Database Directory Tool

This tool can be used to move a database directory minidisk to another minidisk of equal or greater size. Moving the directory to a larger minidisk should be done when there is not enough space in the current directory to support the addition of new dbspaces. Refer to the *DB2 Server for VM System Administration* manual for information about the many technical considerations and associated precautions involved when copying the directory minidisk (BDISK).

You must also be sure the new directory minidisk (BDISK) is added to the database virtual machine's VM directory and linked in WRITE mode. You do not, however, need to format and reserve the minidisk, since Control Center will have the SQLCDBEX utility format and reserve it.

How the Copy/Expand Database Directory Tool Works

To begin, invoke the Copy/Expand Database Directory tool panel interface, Option **D** and specify the following information:

- the new minidisk virtual address
- the block size for the new minidisk (512 or 4097), and
- indicate whether the directory will be expanded (moved to larger minidisk) or not.

Once this information has been specified you can initiate the process immediately or schedule it to run later. When the process runs, the database machine will be terminated by Control Center. When the database has ended operations, Control Center starts and manages the copy/move of the directory minidisk in single user mode using the SQLCDBEX utility.

When the copy directory process has completed, the database is returned to its original status (either up or down).

Copy/Expand Database Directory Entry Panel

Shown in Figure 151 is the entry panel for the Copy/Expand Database Directory tool.

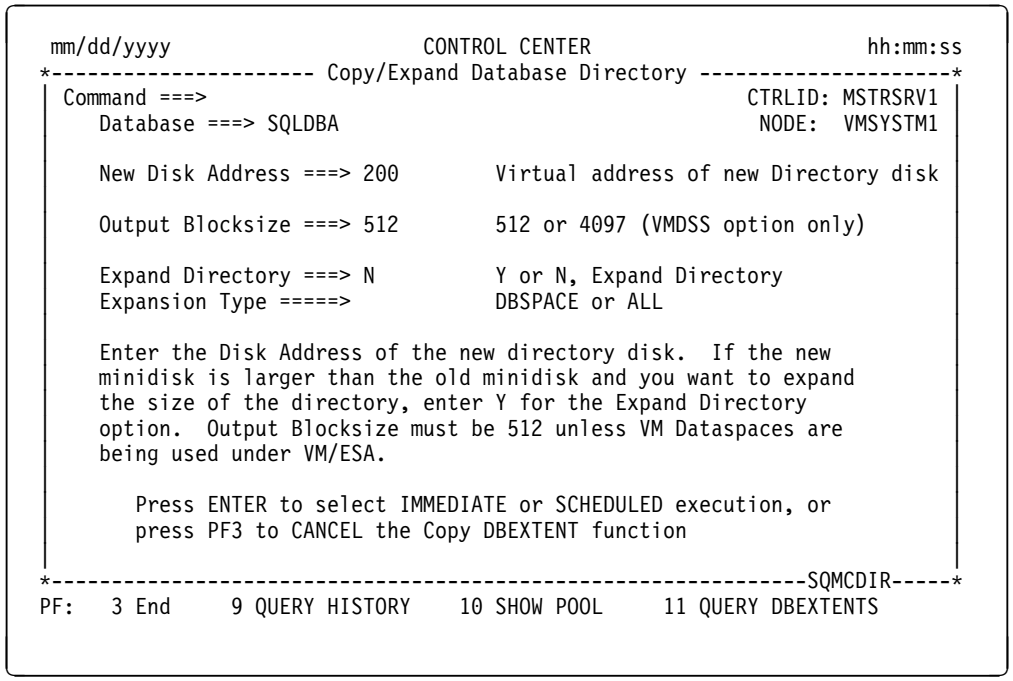


Figure 151. Control Center Copy/Expand Database Directory Entry Panel

Entry Field	Description
New Disk Address	This is the minidisk virtual address that the directory (BDISK) will be copied to. This minidisk must be linked by the database machine in WRITE mode.
Output Blocksize	This entry field should only be changed to 4097 if the database machine is using the database VMDSS feature. Refer to the *DB2 Server for VM System Administration* manual. If the database does not have the VMDSS feature, then the blocksize specified *must* be 512.
Expand Directory	If the directory minidisk (BDISK) is being moved to a larger minidisk, then enter Y; if not, then enter N (default).
Expansion Type	If you choose to expand the directory, then enter either DBSPACE to expand dbspace pages or ALL to expand both dbspace and dbextent pages in the directory. To expand all pages the database *must* be at SQL/DS 3.5 or DB2 Server for VM 5.1 or later.

PF Key Selections

PF Key	Description
QUERY HISTORY (PF9)	Displays file CDBEXHIS that contains a history of all the dbextents (directory, log, and data) that have been copied in date/time order. Figure 150 on page 330 shows that the directory (BDISK) minidisk was copied on 03/15/97 to a minidisk at virtual address 207. 207 is of the same size (N) as the original directory disk and has a block size of 512.
SHOW POOL (PF10)	Issues a database SHOW POOL command to the database and displays the response (refer to Chapter 16, “Database Operation Interface Tools” on page 197). You can use this information to review a database's current dbextent space usage.
QUERY DBEXTENTS (PF11)	Provides a detailed report of a database's dbextents, including a mapping to current virtual addresses, volids, and storage pools. Additional information includes a list of any deleted dbextents, the number of cylinders and pages allocated, and real device addresses. The report is presented in both dbextent and storage pool order. Refer to Chapter 37, “Control Center Administration Tools” on page 485.

Copy/Expand Database Directory Control File

A CDBEXDIR control file is created and kept on the Control Center's 191 A-disk for each database machine (filename matches the database virtual machine name). This file has a record of the virtual address of the new database directory (BDISK) minidisk, the block size for the new minidisk, and an indication of whether the new minidisk is larger than the old. In the example shown in Figure 152, the directory will be copied to a new minidisk with a virtual address of 207, the new 207 minidisk is not (N) larger than the existing directory minidisk, and 207 has a blocksize of 512.

```
BDISK 207 N 512 DBSPACE
END
```

Figure 152. Example Copy/Expand Database Directory Control File (CDBEXDIR)

Copy/Move Database Log Disk Tool

This tool can be used to move a database log minidisk to another minidisk of the same size. Use this tool when it is necessary to relocate a log minidisk to another device in support of migration or device performance requirements.

There are many technical considerations and associated precautions involved when copying a log (LOGDSK1, LOGDSK2) minidisk. Refer to the *DB2 Server for VM System Administration* manual.

The new log minidisks to be copied to must already be added to the database virtual machine's VM directory. The new minidisks, however, do not need to be formatted and reserved since Control Center will have them reformatted and reserved by the SQLCDBEX utility.

How the Copy/Move Database Log Disk Tool Works

To begin, you invoke the Copy/Move Database Log Disk tool panel interface (option L) and specify the new virtual address for the database log minidisks to be moved.

Once this information has been specified you can initiate the process immediately or schedule it to run later. When the process runs, the database machine will be terminated by Control Center. When the database has ended operations, Control Center starts and manages the copy/move of the log minidisk(s) in single user mode using the SQLCDBEX utility.

When the move/copy tool has completed, the database is returned to its original status (either up or down).

Copy/Move Database Log Disk Entry Panel

Shown in Figure 153 is the entry panel for the Copy/Move Database Log Disk tool.

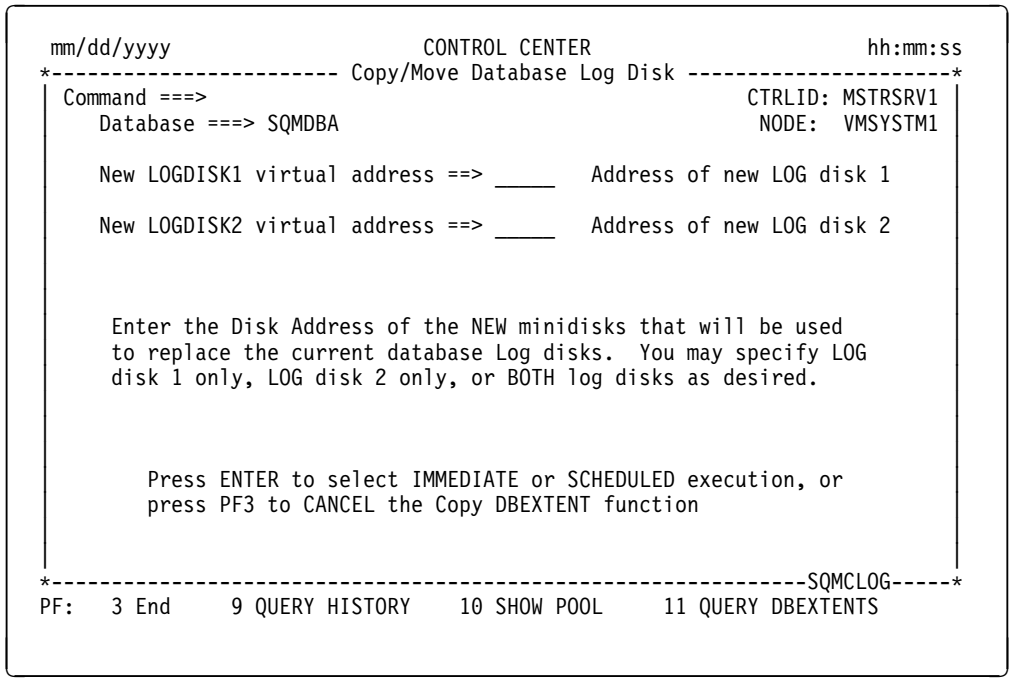


Figure 153. Control Center Copy/Move Database Log Disk Entry Panel

Entry Field	Description
New LOGDISK1 Virtual Address	Use this entry field to specify the new virtual address for the LOGDISK1 (defined as LOGDSK1 in the database's SQLFDEF file).
New LOGDISK2 Virtual Address	Use this entry field to specify the new virtual address for the LOGDISK2 (defined as LOGDSK2 in the database's SQLFDEF file).

PF Key Selections

PF Key	Description
QUERY HISTORY (PF9)	Displays file CDBEXHIS that contains a history of all the dbextents (directory, log, and data) that have been copied in date/time order. Refer to Figure 150 on page 330.
SHOW POOL (PF10)	Issues a SHOW POOL command to the database and displays the response (refer to Chapter 16, “Database Operation Interface Tools” on page 197). You can use this information to review a database's current dbextent space usage.
QUERY DBEXTENTS (PF11)	Provides a detailed report of a database's dbextents, including a mapping to current virtual addresses, volids, and storage pools. Additional information includes a list of any deleted dbextents, the number of cylinders and pages allocated, and real device addresses. The report is presented in both dbextent and storage pool order. Refer to Chapter 37, “Control Center Administration Tools” on page 485.

Copy/Move Database Log Disk Control File

The Copy/Move Database Log Disk tool creates a control file on Control Center's 191 A-disk. This control file is identified by a filetype of CDBLOGEX and has a filename matching the database virtual machine name. The file contains information to be used during the copy/move of a database's log minidisk.

The example shown in Figure 154 shows two logs (LOGDSK1 and LOGDSK2) that are to be moved to new minidisks at virtual addresses 301 and 302.

```
LOGDSK1 301
LOGDSK2 302
END
```

Figure 154. Example Copy/Move Database Log Disk Control File (CDBLOGEX)

Copy/Move Data Disk Tool

This tool can be used to move a database data (DDSKn) minidisk (dbextent) from one minidisk to another of the same size. Use this tool when it is necessary to relocate a data minidisk to another device in support of migration or device performance requirements.

There are many technical considerations and associated precautions involved when copying a data minidisk. Refer to the *DB2 Server for VM System Administration* manual.

The new data minidisks must be added to the database virtual machine's VM directory. The new minidisks, however, do not need to be formatted and reserved since Control Center will have them reformatted and reserved during the copy process.

How the Copy/Move Data Disk Tool Works

To begin, you invoke the Copy/Move Data Disk tool panel interface (option E) and specify the dbextent numbers to be moved and the new virtual addresses they will be moved to.

Once this information has been specified, you can initiate the process immediately or schedule it to run later. When the process runs the database machine will be terminated by Control Center. When the database has ended operations, Control Center starts and manages the copy/move of the data minidisk(s) in single user mode using the SQLCDBEX utility.

When the move/copy tool has completed, the database is returned to its original status (either up or down).

Copy/Move Data Disk Entry Panel

Shown in Figure 155 on page 336 is the entry panel for the Copy/Move Data Disk tool.

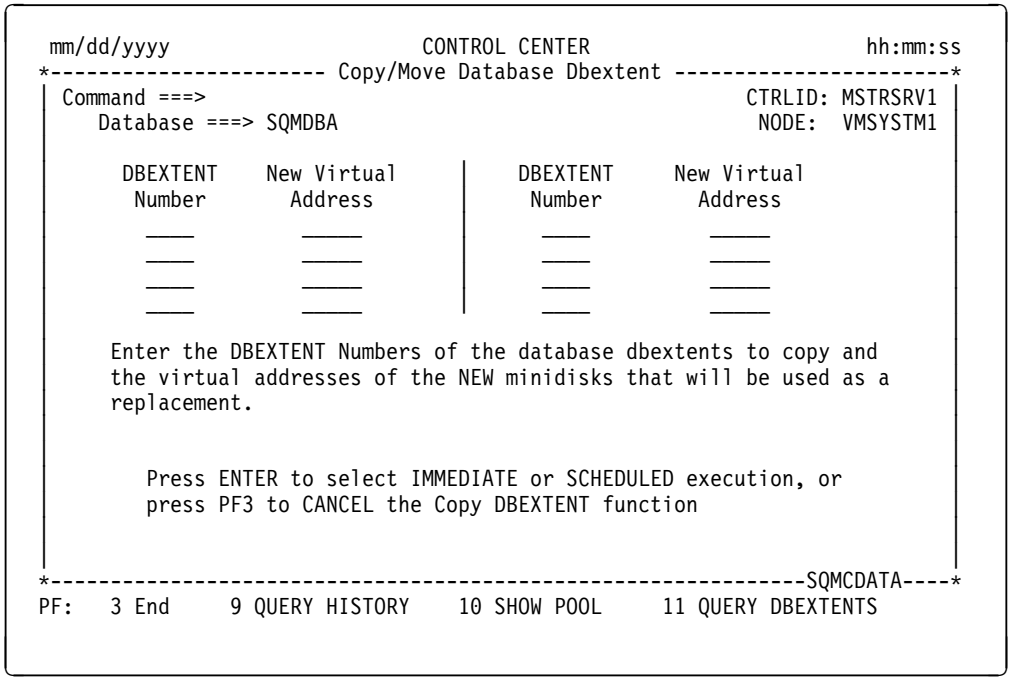


Figure 155. Control Center Copy/Move Data Disk Entry Panel

Entry Field	Description
DBEXTENT Number	The DBEXTENT Number entry field identifies each dbextent that you want copied. To determine a dbextent number use **PF11** key (QUERY DBEXTENTS), discussed below. You may enter up to eight dbextent numbers per Copy/Move data disk job (see note below).
New Virtual Address	This entry field identifies the virtual address for the new minidisks. The new minidisks must be the same size as the old minidisks and they must be added to the database virtual machine's VM directory prior to running the Copy/Move tool.

PF Key Selections

PF Key	Description
QUERY HISTORY (PF9)	Displays file CDBEXHIS that contains a history of all the dbextents (directory, log, and data) that have been copied in date/time order. Refer to Figure 150 on page 330.
SHOW POOL (PF10)	Issues a database SHOW POOL command to the database and displays the response (refer to Chapter 16, “Database Operation Interface Tools” on page 197). You can use this information to review a database's current dbextent space usage.

QUERY DBEXTENTS (PF11) Provides a detailed report of a database's dbextents, including a mapping to current virtual addresses, volids, and storage pools. Additional information includes a list of any deleted dbextents, the number of cylinders and pages allocated, and real device addresses. The report is presented in both dbextent and storage pool order. Refer to Chapter 37, “Control Center Administration Tools” on page 485.

Copy/Move Data Disk Control File

The Copy/Move Data Disk tool creates a control file on Control Center's 191 A-disk. This control file is identified by a filetype of CDBEXTNT and has a filename matching the database virtual machine name. The file contains information to be used during the copy/move of a database's data minidisks.

In the example shown in Figure 156, dbextent number 4 will be copied to a new minidisk virtual address of 209, and dbextent number 7 will be copied to a new minidisk virtual address of 210.

```
4 209
7 210
END
```

Figure 156. Example Copy/Move Data Disk Control File (CDBEXTNT)

Note:

The Copy/Move Data Disk tool uses a single control file for each database machine. You cannot, therefore, schedule multiple Copy/Move Data Disk jobs for the same database because each new job (scheduled or immediate invocation) will overlay the previously created control file.

Add DBSPACE Tool

The Add DBSPACE tool automates the execution of the SQLADBSP utility. Use this tool to add new dbspaces to your database's storage pools.

Prior to adding dbspaces to your database's storage pools, you should review and be familiar with the add dbspace information described in the *DB2 Server for VM System Administration* manual.

How the Add DBSPACE Tool Works

You start the Add DBSPACE process by using the panel interface to define new dbspaces and to identify which storage pools they are to be added to.

After defining the new dbspaces to be added, the panel interface will give you the option of executing the Add DBSPACE tool immediately or scheduling it for a later date and time using the Job Scheduling tool.

When the database is terminated, Control Center will manage the execution of the SQLADBSP utility. This DB2 Server for VM-provided utility runs in single user mode on the database machine's console. It will manage the database's console during the execution of this utility.

When the Add DBSPACE job completes, the database will be returned to its original status (either up or down).

After Adding a DBSPACE

If the Archive-addspace parameter (refer to Chapter 18, “Database Startup and Termination Tools” on page 213) in the database's PARMS file is set to Y, then a full database archive will be automatically initiated by Control Center when the Add DBSPACE processing completes.

Note:

If the database is not archived after adding dbspaces, the DBA should be aware that a subsequent database recovery will not include the added dbspaces, since this single user mode activity is not logged. Also, any updating done within the added dbspaces will not be applied during a database recovery (since the dbspaces where the updates occurred no longer exist).

About the Add DBSPACE Control File

Shown in Figure 157 is an example of the SQLADBSP file which is used to control the addition of dbspaces. You will need to create this file as specified in the *DB2 Server for VM System Administration* manual. You *must* specify the internal dbspaces *every* time that the SQLADBSP process is executed. When the Add DBSPACE tool is invoked under Control Center, the previous SQLADBSP file will be provided, allowing the internal statement to be retained for the next execution.

```
PRIVATE 128 7
PRIVATE 128 7
PRIVATE 128 7
PRIVATE 128 7
PUBLIC  512 9
PUBLIC  1024 9
INTERNAL  230  8192 3
```

Figure 157. Example Add DBSPACE Control File (SQLADBSP)

Note:

The Add DBSPACE tool uses a single control file for each database machine. You cannot, therefore, schedule multiple add dbspace jobs for the same database because each new job (scheduled or immediate invocation) will overlay the previously created control file.

About the Add DBSPACE Panel

You invoke the Add DBSPACE tool by selecting Option **A** (ADD DBSPACES) on the Database Utility Functions panel. The next panel displayed is informational only, and pressing ENTER on this panel will display the SQLADBSP control file in edit (XEDIT) mode.

The SQLADBSP control file displayed will be the previous Add DBSPACE control file established. If there is no SQLADBSP control file for the database, then a file will be generated with sample values included.

Add DBSPACE Log File

When the SQLADBSP process runs, Control Center creates a log file of all messages generated during the add dbspace process. This file is kept on Control Center's 191 A-disk and is identified by a filetype of ADDSPLOG (filename matches the database virtual machine name).

Each time a dbspace is added to a database, a new file will be generated, replacing the previous log file.

Example Add DBSPACE Log File

Shown in Figure 158 on page 340 is an example of an ADDSPLOG file. The **highlighted** fields indicate responses made by Control Center. All other lines are messages generated on the database machine while the SQLADBSP utility runs.

```
09:12:09       SQLDBA SQLADBSP requested
09:12:18  ARI0717I Start SQLADBSP EXEC: 09:12:16 EST.
09:12:18  ARI0646D SQLDBA SQLADBSP A was found. Should the database use this
09:12:18           file for processing?  Enter:
09:12:19              0(No)  to erase the file, or
09:12:19              1(Yes) to use the file.
09:12:18 1
09:12:19  ARI0638D Do you want to modify the SQLDBA SQLADBSP file?
09:12:20           Enter 0(No) or 1(Yes).
09:12:20 0
09:12:20  ARI6184D Do you want to continue the ADD DBSPACE process?
09:12:21           Enter 0(No) or 1(Yes).
09:12:20 1
09:12:21  ARI0717I Start SQLSTART EXEC: 09:12:21 EST.
09:12:24  ARI0320I The default database name is SQLDBA.
09:12:24  ARI0663I FILEDEFS in effect are:
09:12:25  Z        DISK     DMSNAM   LOADLIB  *
09:12:25  ARIARCH  TAP1  SL  00001  VOLID QU1192
09:12:25  SYSPRINT TERMINAL
09:12:26  SYSIN    DISK     SQLDBA   SQLADBSP A1
09:12:26  ARISQLLD DISK     ARISQLLD LOADLIB  Q1
09:12:26  BDISK    DISK     200
09:12:26  LOGDSK1  DISK     201
09:12:27  DDSK1    DISK     202
09:12:27  DDSK2    DISK     203
09:12:28  ARIUSRDD DISK     USERLIB  LOADLIB  *
09:12:28  ARITRAC  TAP2  SL  00001
09:12:29  ARI0025I The program ARISQLDS is loaded at 622000.
09:12:29  ARI0025I The program ARIXRDS is loaded at 4C9000.
09:12:30  ARI0025I The program ARIXSXR is loaded at 7AF000.
09:12:30  ARI0025I The program ARICMOD is loaded at 7C44B0.
09:12:30  ARI0015I ACCOUNT parameter value is N.
09:12:31  ARI0015I DUMPTYPE parameter value is P.
09:12:31  ARI0015I LOGMODE parameter value is A.
09:12:32  ARI0015I STARTUP parameter value is S.
09:12:32  ARI0015I SYSMODE parameter value is S.
09:12:33  ARI0015I CHARNAME parameter value is ENGLISH.
09:12:34  ARI0015I DBNAME parameter value is SQLDBA.
09:12:36  ARI0016I ARCHPCT parameter value is 80.
09:12:37  ARI0016I CHKINTVL parameter value is 10.
09:12:37  ARI0016I NCUSERS parameter value is 1.
09:12:38  ARI0016I NLRBS parameter value is 512.
09:12:39  ARI0016I NPAGBUF parameter value is 50.
09:12:40  ARI0016I SOSLEVEL parameter value is 10.
09:12:41  ARI0016I DISPBIAS parameter value is 10.
09:12:41  ARI0025I The program ARISDBMS is loaded at 42B000.
09:13:04  ARI0915I One or more DBSPACES were added to database.
09:13:07  ARI0032I The database has terminated.
09:13:07  ARI0043I The database manager return code is 0.
09:13:08 SQLDBA SQLADBSP completed
```

Figure 158. Example Add DBSPACE Log File (ADDSPLOG)

Add DBSPACE History File

When the SQLADBSP process runs, Control Center updates a file recording the information about the dbspace being added. This file is kept on Control Center's 191 A-disk for each database machine and is identified by a filetype of ADBSPHIS (filename matches the database virtual machine name).

Example Add DBSPACE History File

Shown in Figure 159 is an example of an ADBSPHIS file which provides a history of *all* dbspaces that have been added to the database using Control Center.

```
02/24/97 11:02:34 ADDED (TYPE, SIZE, POOL) PUBLIC 8192 4
02/24/97 11:02:34 ADDED (TYPE, SIZE, POOL) INTERNAL 230 1024 3
03/16/97 09:43:55 ADDED (TYPE, SIZE, POOL) PRIVATE 128 2
03/16/97 09:43:55 ADDED (TYPE, SIZE, POOL) PRIVATE 128 2
03/16/97 09:43:55 ADDED (TYPE, SIZE, POOL) PRIVATE 128 2
03/16/97 09:43:55 ADDED (TYPE, SIZE, POOL) PRIVATE 512 4
03/16/97 09:43:55 ADDED (TYPE, SIZE, POOL) PRIVATE 1024 4
03/16/97 09:43:55 ADDED (TYPE, SIZE, POOL) INTERNAL 230 1024 3
```

Figure 159. Example Add DBSPACE History File (ADBSPHIS)

Database COLDLOG Tool

The Database COLDLOG tool automates the use of the database SQLLOG utility. Use this tool to COLDLOG your database's logs. Refer to the *DB2 Server for VM System Administration* manual for information on performing a COLDLOG.

How the Database COLDLOG Tool Works

The tool is invoked through the panel interface and you have the option of running it immediately or scheduling it for later execution using the Job Scheduling tool.

When the tool runs, Control Center brings the database down and invokes the SQLLOG utility to COLDLOG the database's logs. Control Center will answer no (0 or N) to the database ARI6129D “(do you want to do a CMS FORMAT/RESERVE command)” question for DB2 Release 3.3 and earlier. For Release 3.4 the response is CONTINUE (the message context has been changed).

Upon completion of the COLDLOG, the database is returned to its original status (either up or down).

Database COLDLOG Options Panel

Figure 160 on page 342 shows the Database COLDLOG options panel displayed when Option **CL** (COLDLOG) on the Database Utility Functions panel is selected. COLDLOGs can be scheduled or submitted for immediate execution. If you run the Option **I** (Immediate), then the database will be terminated and the COLDLOG will begin. Option **S** will display a panel prompting you for scheduling-related information (refer to Figure 66 on page 171).

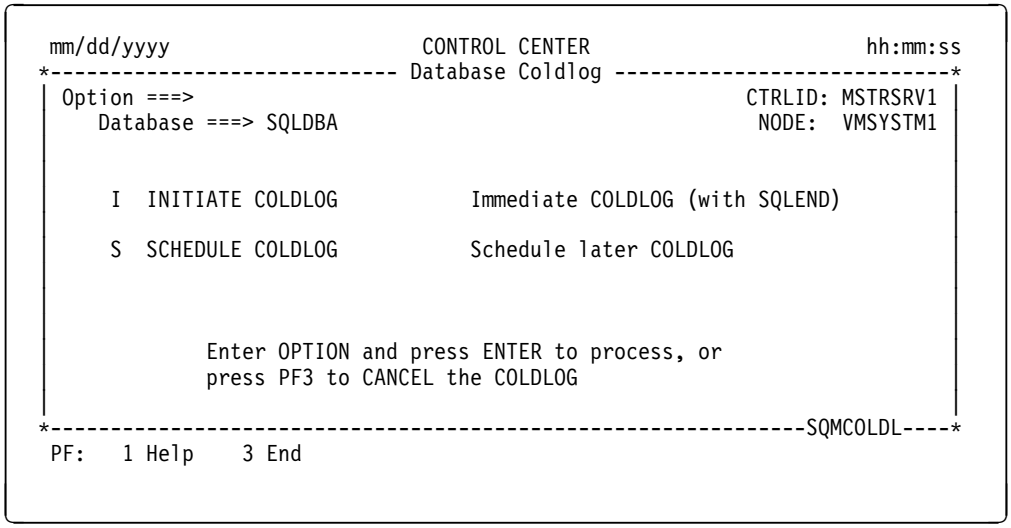


Figure 160. Database COLDLOG Options Panel

Database COLDLOG Log File

When the Database COLDLOG tool runs, the database's console messages are written to a log file (database COLDLOG), which is kept on Control Center's 191 A-disk. This file will be replaced with a new file containing the latest process log each time the Database COLDLOG tool is run.

Example Database COLDLOG Log File

Shown in Figure 161 on page 343 is an example of a COLDLOG file. The **highlighted** fields indicate responses made by Control Center. All other lines are messages generated on the database machine while the SQLADBSP utility runs.

```
03/16/97 10:22:24 COLDLOG request issued for database SQLDBA
03/16/97 10:22:24  DASD 0300 DETACHED
03/16/97 10:22:24  DASD 0201 DETACHED
03/16/97 10:22:25  DASD 0202 DETACHED
03/16/97 10:22:25  DASD 0200 DETACHED
03/16/97 10:22:25  DASD 0204 DETACHED
03/16/97 10:22:25  DASD 0220 DETACHED
03/16/97 10:22:25  DASD 0221 DETACHED
03/16/97 10:22:26  DASD 0222 DETACHED
03/16/97 10:22:26  DASD 0224 DETACHED
03/16/97 10:22:26  DASD 0240 DETACHED
03/16/97 10:22:26  DASD 0243 DETACHED
03/16/97 10:22:26  DASD 0244 DETACHED
03/16/97 10:22:27  DASD 0245 DETACHED
03/16/97 10:22:27  DASD 0225 DETACHED
03/16/97 10:22:27  DASD 0241 DETACHED
03/16/97 10:22:27  ARI0797I End SQLSTART EXEC: 03/16/97 10:22:16 EST
03/16/97 10:22:27  Ready; T=53.80/179.22 10:22:16
03/16/97 10:22:28  TAPE 0181 DETACHED
03/16/97 10:22:28  ARI0717I Start SQLLOG EXEC: 03/16/97 10:22:24 EST.
03/16/97 10:22:28  ARI6129D Do you want to do a CMS FORMAT/RESERVE command
03/16/97 10:22:28           on disks 201 and 202?
03/16/97 10:22:29           Enter 0(No), 1(Yes), or CANCEL to cancel COLDLOG.
03/16/97 10:22:29           Enter 0(No) if you are not reconfiguring the
03/16/97 10:22:29           logs.
03/16/97 10:22:29           Warning: If you are running LOGMODE=L,
03/16/97 10:22:29           a response of 1(Yes) invalidates previous
03/16/97 10:22:30           or current log archive restore sets.
03/16/97 10:22:30 0
03/16/97 10:22:30  ARI0717I Start SQLSTART EXEC: 03/16/97 10:22:28 EST.
03/16/97 10:22:30  ARI0663I FILEDEFS in effect are:
03/16/97 10:22:30  Z        DISK     DMSNAM   LOADLIB  *
03/16/97 10:22:31  ARIARCH  TAP1  SL  00001  VOLID QU1251
03/16/97 10:22:31  ARISQLLD DISK     ARISQLLD LOADLIB  Q1
03/16/97 10:22:31  BDISK    DISK     300
03/16/97 10:22:31  LOGDSK1  DISK     201
03/16/97 10:22:31  LOGDSK2  DISK     202
03/16/97 10:22:32  DDSK1    DISK     200
03/16/97 10:22:32  DDSK2    DISK     204
03/16/97 10:22:32  DDSK3    DISK     220
03/16/97 10:22:32  DDSK4    DISK     221
03/16/97 10:22:33  DDSK5    DISK     222
03/16/97 10:22:33  DDSK7    DISK     224
03/16/97 10:22:33  DDSK9    DISK     240
03/16/97 10:22:33  DDSK12   DISK     243
03/16/97 10:22:33  DDSK13   DISK     244
03/16/97 10:22:34  DDSK14   DISK     245
03/16/97 10:22:34  DDSK15   DISK     225
03/16/97 10:22:34  DDSK10   DISK     241
03/16/97 10:22:34  ARIUSRDD DISK     USERLIB  LOADLIB  *
03/16/97 10:22:34  ARITRAC  TAP2  SL  00001
03/16/97 10:22:35  ARILARC  TAP3  SL  00001
```

Figure 161 (Part 1 of 2). Example Database COLDLOG Log File (COLDLOG)

```
03/16/97 10:22:35  ARI0025I The program ARISQLDS is loaded at 6C9000.
03/16/97 10:22:35  ARI0025I The program ARIXRDS is loaded at 570000.
03/16/97 10:22:35  ARI0025I The program ARIXSXR is loaded at 8A9000.
03/16/97 10:22:36  ARI0025I The program ARICMOD is loaded at 8A7000.
03/16/97 10:22:36  ARI0015I DUALLOG parameter value is Y.
03/16/97 10:22:36  ARI0015I ACCOUNT parameter value is N.
03/16/97 10:22:37  ARI0015I DUMPTYPE parameter value is P.
03/16/97 10:22:37  ARI0015I LOGMODE parameter value is Y.
03/16/97 10:22:37  ARI0015I STARTUP parameter value is L.
03/16/97 10:22:38  ARI0015I SYSMODE parameter value is S.
03/16/97 10:22:38  ARI0015I EXTEND parameter value is N.
03/16/97 10:22:39  ARI0015I CHARNAME parameter value is ENGLISH.
03/16/97 10:22:39  ARI0015I DBNAME parameter value is CPVENDBA.
03/16/97 10:22:39  ARI0015I TRACDSC parameter value is 00.
03/16/97 10:22:40  ARI0015I TRACRDS parameter value is 0000000.
03/16/97 10:22:40  ARI0015I TRACDBSS parameter value is 00000000000.
03/16/97 10:22:41  ARI0016I CHKINTVL parameter value is 10.
03/16/97 10:22:41  ARI0016I NCSCANS parameter value is 30.
03/16/97 10:22:41  ARI0016I NCUSERS parameter value is 1.
03/16/97 10:22:42  ARI0016I NDIRBUF parameter value is 14.
03/16/97 10:22:42  ARI0016I NLRBS parameter value is 512.
03/16/97 10:22:43  ARI0016I NLRBU parameter value is 1000.
03/16/97 10:22:43  ARI0016I NPAGBUF parameter value is 14.
03/16/97 10:22:43  ARI0016I SLOGCUSH parameter value is 90.
03/16/97 10:22:44  ARI0016I SOSLEVEL parameter value is 10.
03/16/97 10:22:44  ARI0016I DISPBIAS parameter value is 10.
03/16/97 10:22:45  ARI0920D Cold-log function invoked.
03/16/97 10:22:45           Logs will be formatted and existing log data destroyed.
03/16/97 10:22:45
03/16/97 10:22:46           Enter either:
03/16/97 10:22:46             COLDLOG  to continue, or
03/16/97 10:22:46             CANCEL   to cancel.
03/16/97 10:22:46 COLDLOG
03/16/97 10:30:05  The database has terminated.
03/16/97 10:30:05  ARI0043I The database manager return code is 0.
```

Figure 161 (Part 2 of 2). Example Database COLDLOG Log File (COLDLOG)

Reorganize Catalog Indexes Tool

The Reorganize Catalog Indexes tool automates the use of the SQLCIREO utility. Use this tool to rebuild the database system catalog indexes.

Prior to using this tool, review the information presented in the *DB2 Server for VM Database Administration* manual regarding the SQLCIREO utility.

How the Reorganize Catalog Indexes Works

The tool is invoked through the panel interface and you have the option of running it immediately or scheduling it for later execution using the Job Scheduling tool.

When the tool runs, Control Center will bring the database down and invoke the DB2 SQLCIREO utility to reorganize the catalog indexes.

Upon completion of the SQLCIREO utility, the database will be returned to its original status (either up or down).

Reorganize Catalog Indexes Options Panel

Figure 162 shows the reorganize catalog indexes panel displayed when option RC (REORG CATALOG INDEXES) on the Database Utility Functions panel is selected. Reorganizations can be scheduled or submitted for immediate execution. If you run the immediate (I) option, then the database will be terminated and the SQLCIREO utility will begin. The schedule (S) option will display a panel prompting you for scheduling related information (refer to Chapter 14, “Job Scheduling Tool” on page 165).

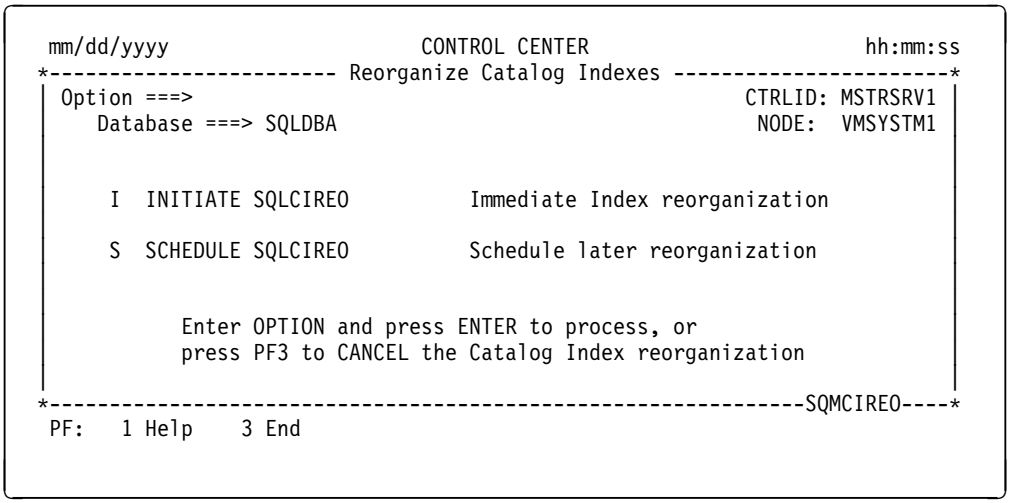


Figure 162. Reorganize Catalog Indexes Options Panel

Reorganize Catalog Indexes Log File

When the Reorganize Catalog Indexes tool runs, the database's console messages are written to a log file (database SQLCIREO), which is kept on Control Center's 191 A-disk. This file will be replaced with a new file containing the latest process log each time the Database COLDLOG tool is run.

Chapter 29. Database Monitoring Tools

Overview

The Database Monitoring tools are used to issue specific database operator commands at specified frequencies during user-designated time periods. The monitor tools collect and analyze the output to determine if any problems exist.

How the Monitors Work

Each morning the Control Center monitor scheduling routine reviews all database monitor schedules for databases managed by Control Center, and schedules the day's monitoring activity to be performed.

When a monitoring activity is started, Control Center issues a specific database operator command to the database's operator console, such as a SHOW LOG operator command.

After a brief wait (1 minute) an analyzer routine is executed to review the console response to determine if any further action is required. Upon completion of the analyzer routine, a monitor scheduler routine schedules the monitor for its next run.

Monitor Analysis Activity

Monitor tools issue database operator commands on a periodic basis. The responses to the database operator commands are collected and analyzed to determine if there are any problems or conditions that should be reported.

When a monitor is added, you identify conditions or problems to be flagged. For example, you can add a Database Log Check monitor that will tell you when your database's log has reached or exceeded a certain percentage used.

Notification of a Detected Condition or Problem

Notification is done upon the detection of a condition or problem by a monitor. Notification is made in the form of an online message and a note file sent to your virtual reader.

Who is Notified: All users who are listed in the Notify-dbstatus field of the database's PARMS file will receive the online and reader file notification. See the description in “Authorization/Notification Parameters” on page 215.

Notification does not repeat each time the monitor runs, unless the detected condition or problem changes. For example, a Database Log Check monitor detects that a database's log has exceeded a threshold of 65 percent full and is now 66 percent full. The Database Log Check monitor would issue notification of this condition upon its detection, but will not continue to notify you unless the condition changes to a different percentage full (67 percent or greater).

If you want the notification message (not the note file) to be reissued each time the monitor runs, then set the Notify-Detect entry field to Y on the monitor scheduling panel. Refer to Figure 179 on page 365.

Important:

The Notify-Detect entry field does *not* apply to these monitors: the User Activity monitors (Users Active, Users Connected, Users Locking), Database Counters monitor, Reset Counters monitor, or VMDSS monitors.

Monitor Reports

Each database monitoring tool has a report generation option. Monitors can log the collected database monitor information each time a monitor runs. All log entries are recorded with the date and time that the monitor ran, along with information obtained in response to a database operator command.

Since you can vary how often a given monitor runs during the day, the reports can be used to show daily database usage and activity trends. For example, you can enable a Database Counters monitor to run at regular intervals (every 15 minutes, for example) throughout the day to get a better understanding of how your database is performing. Figure 163 shows an example of a Database Counters monitor report.

```
Monitor Name: DBCOUNTERS                                    Date: 04/06/97
    Database: SQLDBA                                        Time: 16:00:00
        Type: DB COUNTERS DBSSCALL BEGINLUW PAGEREAD
                          PAGWRITE DIRREAD  DIRWRITE
    Run Days: MON-SUN
        Time: 14:00-16:30
   Frequency: 15 Minute
Next Refresh: 04/24/97

                           Monitor Report Log


  Date     Time   DBSSCALL BEGINLUW PAGEREAD PAGWRITE  DIRREAD DIRWRITE
________ ________ ________ ________ ________ ________ ________ ________
04-06-97 14:00:00      124        8      698      636       44       22
04-06-97 14:15:20       62        7       21       32       33      192
04-06-97 14:30:18       76        6       13        0        0        2
04-06-97 14:45:48       10        2        3        0        0        0
04-06-97 15:00:03        0        0        0        0        0        0
04-06-97 15:15:17       10        2        3        0        0        0
04-06-97 15:30:14        0        0        0        0        0        0
04-06-97 15:45:41       21        4        4        0        0        0
04-06-97 16:00:16        0        0        0        0        0        0
04-06-97 16:15:15        0        0        0        0        0        0
04-06-97 16:30:15        0        0        0        0        0        0
04-07-97 14:00:16        0        0        0        0        0        0
04-07-97 14:15:18        0        0        0        0        0        0
04-07-97 14:30:16      308       19       88       26        3       11
04-07-97 14:45:16        0        0        0        0        0        0
04-07-97 15:00:15      171       12       56       16        0        6
                                  .
                                  .
                                  .
                                  V
```

Figure 163. Monitor Report Example

Who Can Run Database Monitors

To create and schedule a monitor for a database, Control Center DBA or Control Center Administrator authority is required. Viewing of monitors, monitor schedules, and monitor reports requires a minimum level of database user authorization.

Advanced Usage: Control Center provides a default set of tools that are accessible by User, Operator, or DBA-level authorizations. These defaults can be changed so as to add or delete User, Operator or DBA authorizations to specific tools. For more details on how your Control Center authorization levels can be modified, refer to Appendix E, “Authorizations” on page 511.

Database Monitor Types

There are four categories of database monitor tools: Special Checks, User Activity, Space Usage, and Counter Monitors. These categories provide a broad range of database monitoring capability from user activity monitoring to space usage reporting.

Special Check Monitors

The basic intent of these monitors is to verify that your database environment is "alive and well". Monitors for this category are the Database Up and Running Check and Database Log Check monitors.

User Activity Monitors

Monitors in this category can be used to monitor and report on the activity of the users in your database. You can, for example, monitor users that are active, but not processing. Monitors in this category are the Users Connected, Users Active, and Users Locking monitors.

Space Usage Monitors

Use the monitors in this category to monitor and track your database space utilization. The monitors will notify you when space used thresholds are reached or exceeded. Monitors in this category are the DBSPACE Usage and POOL Usage monitors.

Database Counter Monitors

The Database Counter monitor can be used to collect and report on the database counters. All counters, specific counters, or subsets of counters can be monitored. In addition, use the Reset Counters monitor to reset your database counters as required.

Database Monitoring Panel

The initialization of monitors is done for each database using the panel shown in Figure 164 on page 350. You can add, delete, modify, view, and list monitors for a database. When adding (Option **A**) a monitor, you first select a specific monitor type; you are then prompted for additional information that will be used by the monitor.

All information pertaining to a given monitor is stored on the service machine's 191 A-disk and is available for update (Option **M**) at any time.

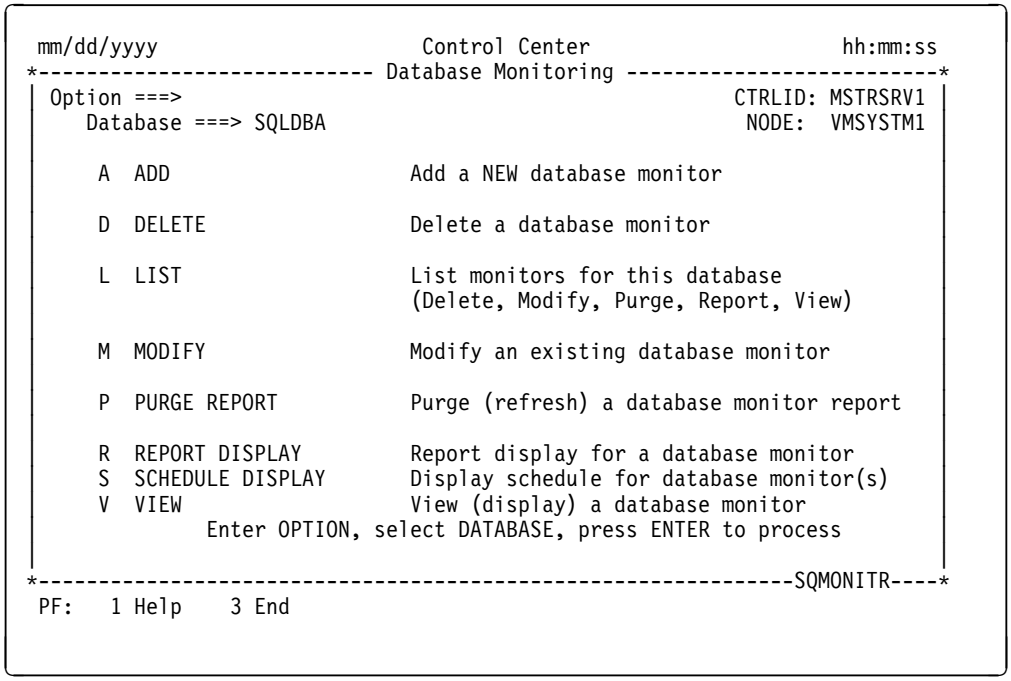


Figure 164. Control Center Main Monitor Panel

Options

Option	Description
ADD (A)	Use this option to add a new monitor for the specified database. The monitor will automatically be scheduled upon completion of this option.
DELETE (D)	This option will prompt you for the name of the monitor to be deleted. The monitor will be deleted from the schedule, and its report sent to a specified user.
LIST (L)	All monitors for a database are listed as shown in Figure 165 on page 351. Several options are made available on this panel for the convenient management of all monitors pertaining to the same database.

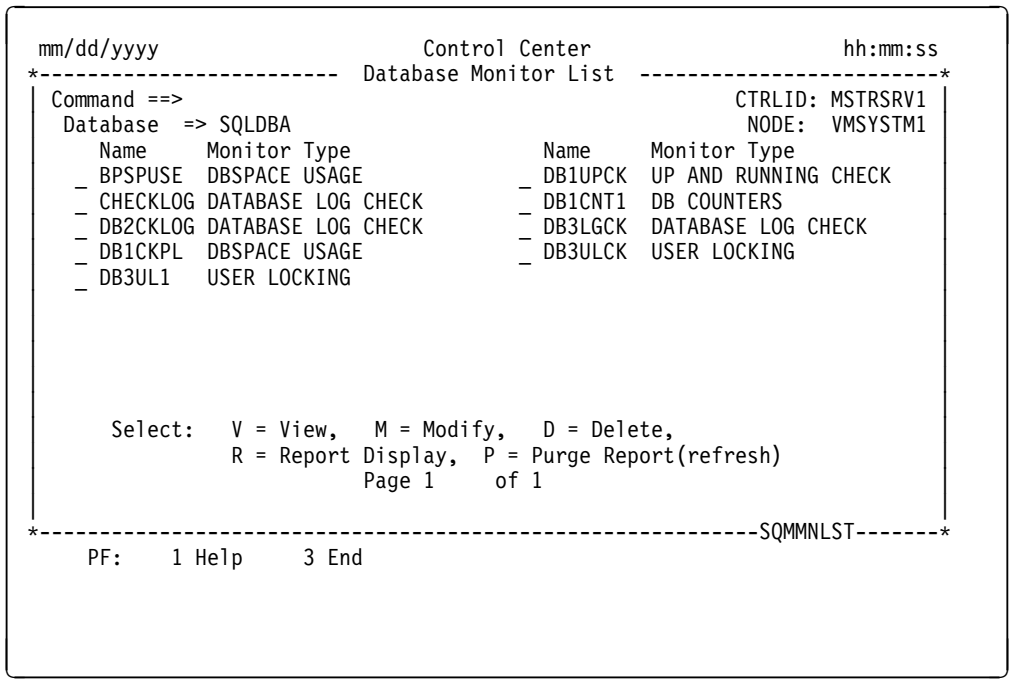


Figure 165. Database Monitor List Panel

MODIFY (M)

Once you have added a database monitor, you can update it by selecting this option. You can modify any specified thresholds or scan conditions, as well as change any scheduling information.

Operational Note: Modification of a monitor will cause automatic rescheduling to occur.

PURGE REPORT (P)

This option will perform a report refresh, which is a deletion of all entries logged to a monitor report. Logging to the report will continue each time the monitor is run.

REPORT DISPLAY (R)

Displays a specified database monitor report.

SCHEDULE DISPLAY (S)

All monitors that are currently active for the specified database will be displayed as shown in Figure 166 on page 352. If your monitor is not listed, then review its scheduling information by using the View **(V)** option.

Scheduling of all active monitor routines is done each morning by Control Center. Only those monitors that are to be run for the current day are shown on the Monitor Schedule Display panel.

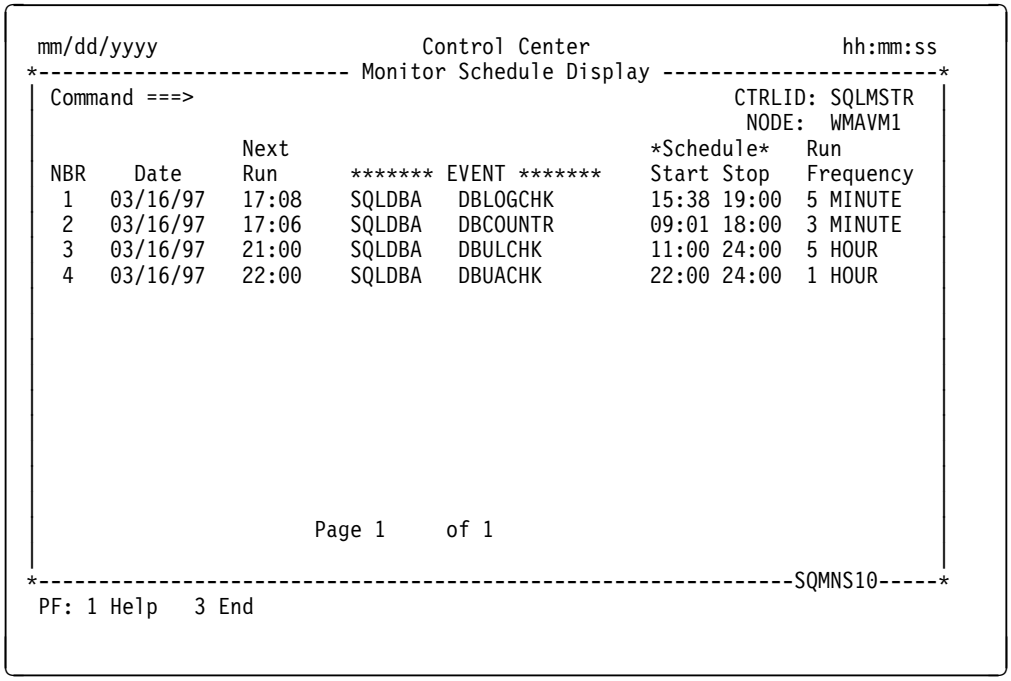


Figure 166. Monitor Schedule Display Panel

VIEW (V)

This option will allow you to view the monitor threshold and condition specifications, as well as the monitor's schedule. You will not, however, be able to modify the monitor with this option.

Selecting a Database Monitor

If Option **A** (Add) was selected from the main monitor panel, then the next panel presented will be the monitor selection panel shown in Figure 167 on page 353. After you have selected a monitor, you will be prompted for conditions to be scanned for, as well as scheduling information.

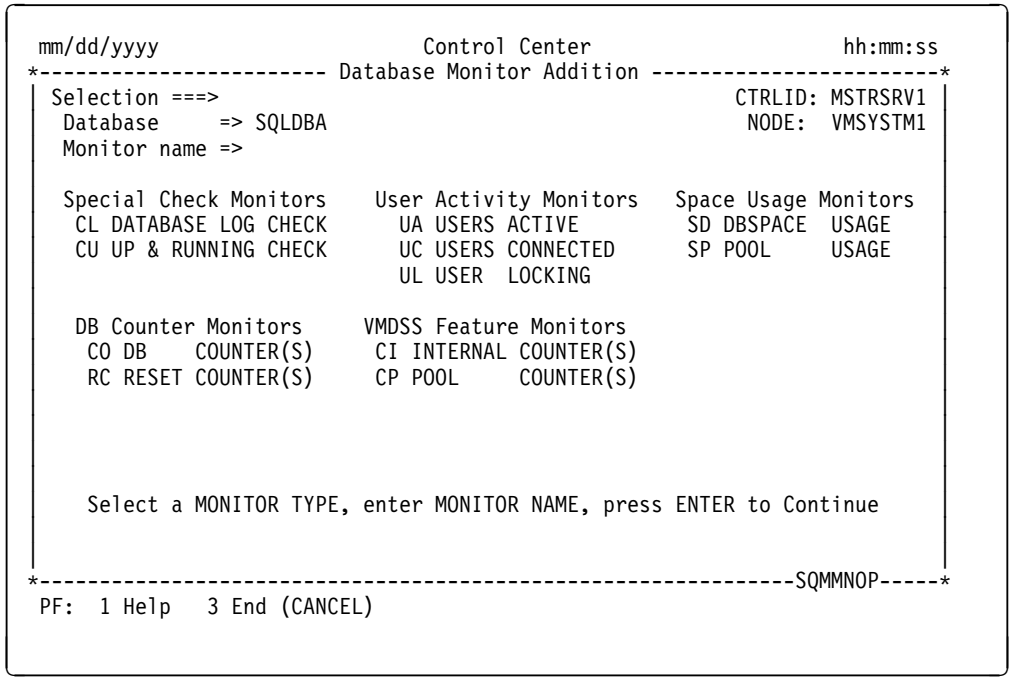


Figure 167. Monitor Selection Panel

Naming a Database Monitor

The name you indicate in the Monitor Name entry field (refer to Figure 167) will uniquely identify the monitor being added. This is done so that you may select the same monitor type for the same database more than once. Each monitor name entered will be verified to be sure that it does not conflict with other monitor names.

Database Log Check Monitor (CL)

This tool monitors database log usage. The Log Used Percentage entry field shown in Figure 168 on page 354 is set by you to a level at which you will receive notification if the database's log ever reaches or exceeds it. For example, the log for the SQLDBA database identified in Figure 168 on page 354 will be checked each time the monitor runs to see if the log percentage used has reached or exceeded 75 percent.

Action To Be Taken

In addition to notifying you that a database log has reached or exceeded a set percent used, the Database Log Check monitor can be instructed to take specific action. For example, if the database's log reaches or exceeds 75 percent used, then the monitor shown in Figure 168 will initiate an SQLEND ARCHIVE. Additionally, if the Schedule Time entry field has been specified, then the Action specified will be scheduled for the time indicated and will not be initiated immediately.

```
 mm/dd/yyyy                      Control Center                      hh:mm:ss
*------------------------ Database Monitor Addition ------------------------*
| Command ==>                                                CTRLID: MSTRSRV1 |
|  Database        => SQLDBA                                  NODE:  VMSYSTM1 |
|  Monitor Name    => LOGCHECK                                                |
|  Monitor Type    => DATABASE LOG CHECK                                      |
|                                                                             |
|  Log Used Percentage => 75      Actions occur when percentage is reached    |
|                                                                             |
|  Select ACTION to be taken(to SCHEDULE  specify a time);                    |
|    SQLEND ARCHIVE  => N         (Y/N; Full archive, database down)          |
|    SQLEND UARCHIVE => N         (Y/N; User archive, database down)          |
|    SQLEND LARCHIVE => N         (Y/N; Log archive, database down)           |
|    DVERIFY         => N         (Y/N; Verify the directory)                 |
|    ARCHIVE         => N         (Y/N; Full archive, database up)            |
|    LARCHIVE        => N         (Y/N; Log archive, database up)             |
|    SCHEDULED TIME  => 22:00     (HH:MM; 00:01 to 24:00, blank for           |
|                                  IMMEDIATE execution.)                      |
|          Press  ENTER to CONTINUE or PF3 to CANCEL                          |
|                                                                             |
*-----------------------------------------------------------------SQMMNCL-----*
 PF: 1 Help    3 End (Cancel)
```

Figure 168. Database Log Check Monitor Panel

Log Check Report

SHOW LOG is the database operator command used by this monitor. If the report option (“Monitor Scheduling Panel” on page 365) is selected, the results of the command are logged.

Database Up and Running Check Monitor (CU)

This tool (Figure 169 on page 355) verifies that your database is still up and running. A SHOW LOG database operator command is issued to the database to verify that the database is responding. If the database terminates, is logged off, does not respond, or responds incorrectly, you will be notified.

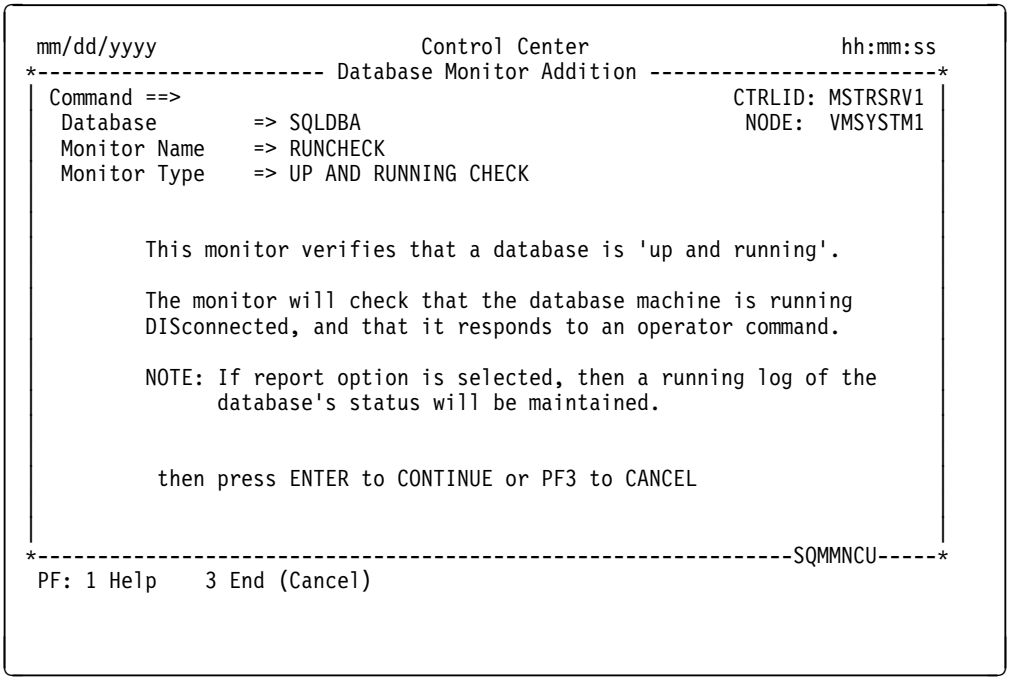


Figure 169. Database Up and Running Check Monitor Panel

Up and Running Check Report

SHOW LOG is the database operator command used by this monitor. If the report option (“Monitor Scheduling Panel” on page 365) is selected, then the status of the database machine is logged.

Users Active Monitor (UA)

This tool monitors active database users. This monitor can be instructed to scan for checkpoint active agents or for the database agents in lock, communication, or checkpoint waits. Refer to Figure 170 on page 356.

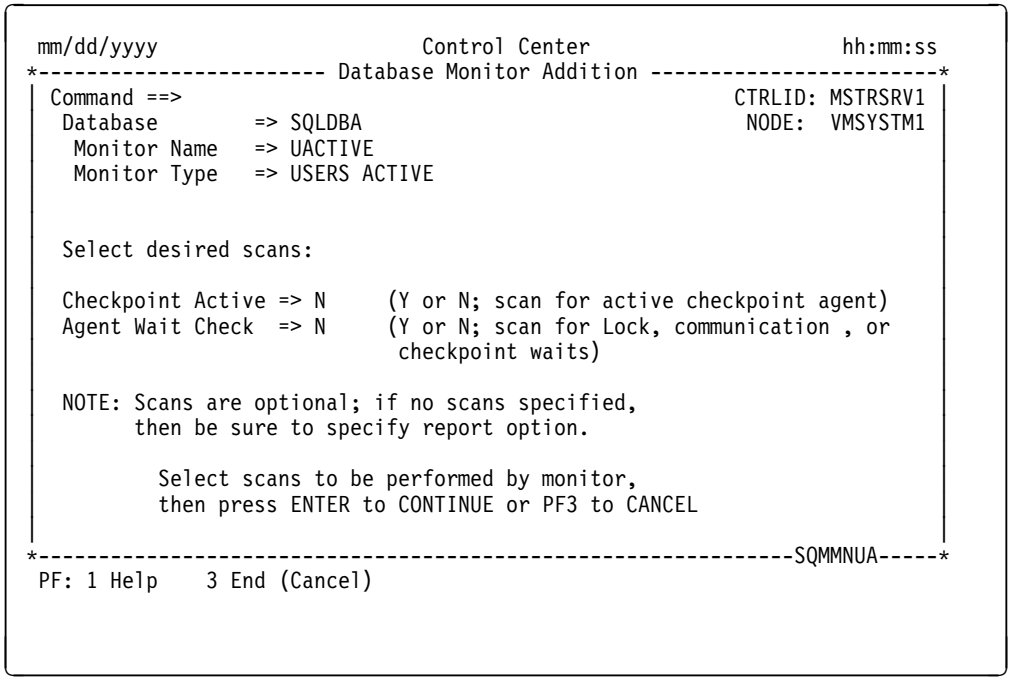


Figure 170. Users Active Monitor Panel

Users Active Report

SHOW ACTIVE is the database operator command used by this monitor. If the report option (see “Monitor Scheduling Panel” on page 365) is selected, the results of the command are logged.

Users Connected Monitor (UC)

This tool monitors users connected to a database. This monitor can be used to scan for active users not processing, users waiting for agents, or for inactive users (refer to Figure 171 on page 357).

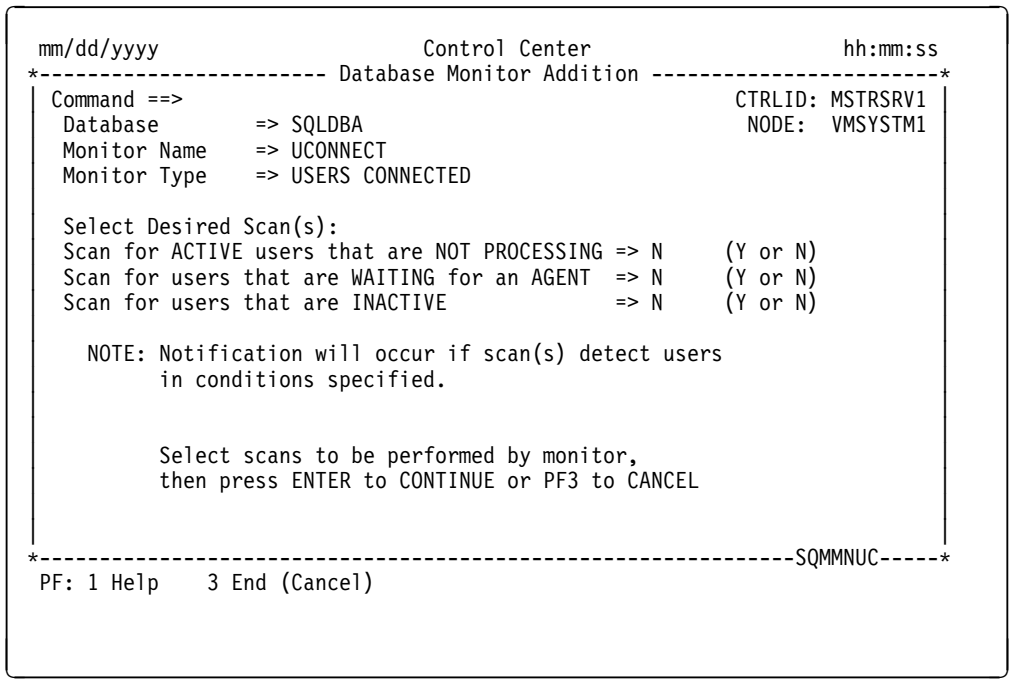


Figure 171. Users Connected Monitor Panel

Users Connected Report

SHOW USERS is the database operator command used by this monitor. If the report option (see “Monitor Scheduling Panel” on page 365) is selected, the results of the command are logged.

User Locking Monitor (UL)

This tool monitors database lock contention. The monitor can be used to scan for lock holders that are not processing, locking due to a checkpoint, or any lock contention at all. Refer to Figure 172 on page 358.

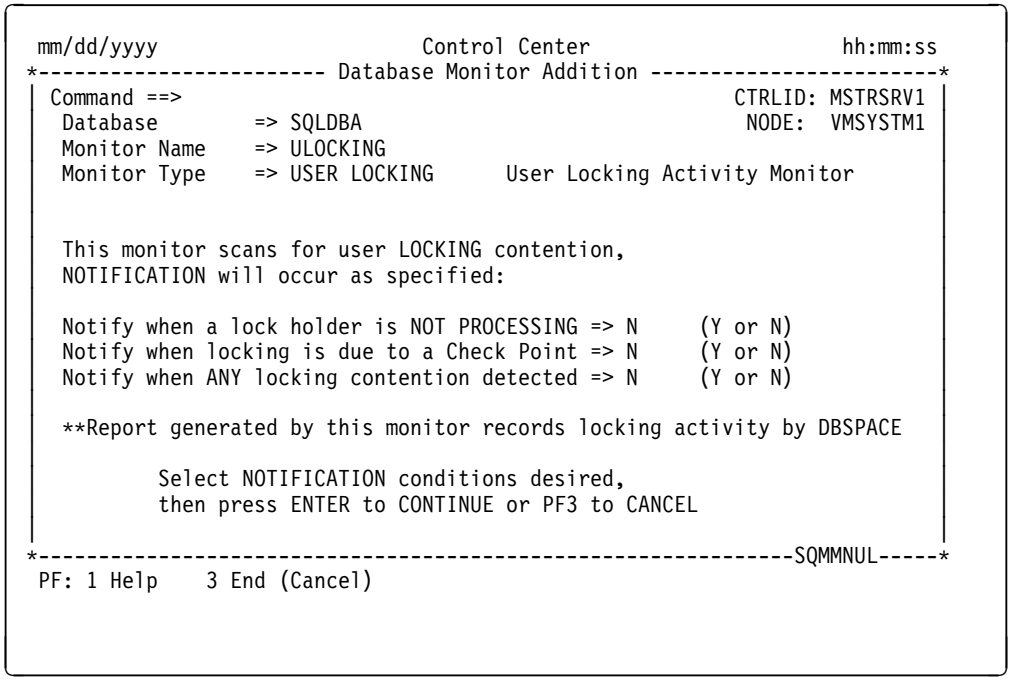


Figure 172. Users Locking Monitor Panel

User Locking Report

SHOW SYSTEM is the database operator command used by this monitor. If the report option (see “Monitor Scheduling Panel” on page 365) is selected, the results of the command are logged.

DBSPACE Usage Monitor (SD)

This tool monitors DBSPACE usage. You must specify a DBSPACE number, which you can obtain from using the Object Search and List tool. You can enter several thresholds for header, data, and index pages. Refer to Figure 173 on page 359. You will be notified if any of the thresholds are reached or exceeded.

```
 mm/dd/yyyy                      Control Center                      hh:mm:ss
*------------------------ Database Monitor Addition ------------------------*
| Command ==>                                                CTRLID: MSTRSRV1 |
|  Database        => SQLDBA                                  NODE:  VMSYSTM1 |
|  Monitor Name    => DBSPUSE                                                  |
|  Monitor Type    => DBSPACE USAGE   Logical Space Monitor                    |
|                                                                              |
|  DBSPACE Number => ___          (Number of DBSPACE to be monitored;          |
|                                  obtained from DBSPACENO col SYSDBSPACES)    |
|  Header Pages   => ____          (Percentage of occupied pages)              |
|  Header Used    => ____          (Percentage of space used)                  |
|  Data   Pages   => ____          (Percentage of occupied pages)              |
|  Data   Used    => ____          (Percentage of space used)                  |
|  Index  Pages   => ____          (Percentage of occupied pages)              |
|  Index  Used    => ____          (Percentage of space used)                  |
|  NOTE:  NOTIFICATION occurs if any of the specified values are exceeded.     |
|                                                                              |
|          Enter parameter values to be checked by monitor,                    |
|          then press ENTER to CONTINUE or PF3 to CANCEL                       |
|                                                                              |
*------------------------------------------------------------------SQMMNSD-----*
 PF: 1 Help    3 End (Cancel)
```

Figure 173. DBSPACE Usage Monitor Panel

DBSPACE Usage Report

SHOW DBSPACE is the database operator command used by this monitor. If the report option (see “Monitor Scheduling Panel” on page 365) is selected, the results of the command are logged.

POOL Usage Monitor (SP)

Use this tool to monitor pool usage for a database. This monitor will review the amount of space used for the storage pool(s) specified (refer to Figure 174 on page 360). To have all storage pools monitored leave the POOL Number parameter field blank. The Percent Used parameter field is optional. If specified, the monitor will notify the database's DBAs when the percentage specified is reached or exceeded for any of the storage pools entered. If a SHORT ON STORAGE condition is detected for any storage pool, then the database's DBAs will be notified. Note, *any* means all the database's storage pools, not just those listed for this monitor. The monitor's report will record the date/time, storage pool number, total pages, number in use, number free, percent used, number of extents, and short-on-storage indicator.

```
 mm/dd/yyyy                      Control Center                      hh:mm:ss
*------------------------ Database Monitor Addition ------------------------*
| Command ==>                                                CTRLID: MSTRSRV1 |
|  Database        => SQLDBA                                  NODE:  VMSYSTM1 |
|  Monitor Name    => SP                                                      |
|  Monitor Type    => POOL USAGE  Pool Usage Monitor                          |
|                                                                             |
|  POOL Number => ___             (Storpool number to be monitored;           |
|                                  leave BLANK to monitor ALL storpools)      |
|                                                                             |
|  Percent Used=> ___             (Percent of storpool pages in use)          |
|                                                                             |
|                                                                             |
|                                                                             |
|  NOTE:  NOTIFICATION occurs if any STORPOOL exceeds the specified value     |
|                                                                             |
|          Enter parameter values to be checked by monitor,                   |
|          then press ENTER to CONTINUE or PF3 to CANCEL                      |
|                                                                             |
*-----------------------------------------------------------------SQMMNSE-----*
 PF: 1 Help    3 End (Cancel)
```

Figure 174. POOL Usage Monitor Panel

POOL Usage Report

SHOW POOL is the database operator command used by this monitor. If the report option (see “Monitor Scheduling Panel” on page 365) is selected, the results of the command are logged.

Database Counters Monitor (CO)

This monitor can be used to record the changes in a database's internal counter(s). The COUNTER operator command will be issued to the database at the frequency that you designate for this monitor. Refer to Figure 175 on page 361. The information returned will then be recorded in the monitor's report log, so that you can monitor changes in counter(s) values.

Specify only those counter(s) that you want to monitor, or specify the ALL COUNTERS option to have all the internal database counters monitored. Place ON next to those counter names that you want to have monitored, and OFF next to the counter names that will not be monitored.

This monitor is used for report purposes only and does not perform condition notification; therefore, be sure to specify the report option (see “Monitor Scheduling Panel” on page 365) for this monitor.

Usage Consideration: The Reset Counters monitor can be used to set counter values to zero at regular intervals.

```
 mm/dd/yyyy                      Control Center                      hh:mm:ss
*------------------------ Database Monitor Addition ------------------------*
| Command ==>                                                CTRLID: MSTRSRV1 |
|  Database        => SQLDBA                                  NODE:  VMSYSTM1 |
|   Monitor Name   => COUNTMON                                                |
|   Monitor Type   => DB COUNTERS                                             |
|                                                                             |
|  ALL COUNTERS  => NO            (YES/NO for ALL COUNTERS)                   |
|                (Indicate  ON or OFF next to each counter)                   |
|       RDSCALL  => OFF        DBSSCALL => OFF       BEGINLUW => OFF         |
|       ROLLBACK => OFF        CHKPOINT => OFF       LOCKLMT  => OFF         |
|       ESCALATE => OFF        WAITLOCK => OFF       DEADLCK  => OFF         |
|       LPAGBUFF => OFF        PAGEREAD => OFF       PAGWRITE => OFF         |
|       LDIRBUFF => OFF        DIRREAD  => OFF       DIRWRITE => OFF         |
|       LOGREAD  => OFF        LOGWRITE => OFF       DASDREAD => OFF         |
|       DASDWRIT => OFF        DASDIO   => OFF                               |
|                                                                             |
|          Specify UP to 8 counter(s) to be monitored (or YES for ALL),       |
|          then press ENTER to CONTINUE or PF3 to CANCEL                      |
|                                                                             |
*-----------------------------------------------------------------SQMMNCO-----*
 PF: 1 Help    3 End (Cancel)
```

Figure 175. Database Counters Monitor Panel

DB Counters Report

The COUNTER database operator command is used by this monitor. If the report option (see “Monitor Scheduling Panel” on page 365) is selected, the results of the command are logged.

Reset Counters Monitor (RC)

This tool resets database counter(s) on a regular basis. The report log for this monitor will record the date and time that the counter(s) are reset.

Specify only those counter(s) that you want to reset, or specify the RESET ALL option to have all the internal database counters reset. Place "ON" next to those counter names that you want to have reset, and "OFF" next to the counter names that will not be reset. Refer to Figure 176 on page 362.

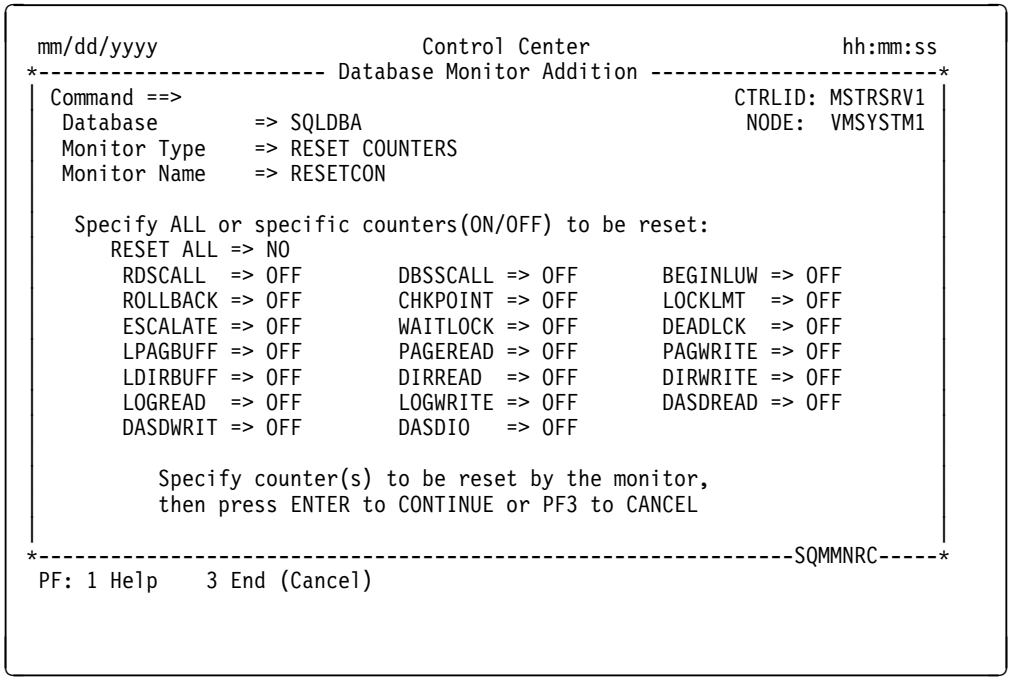


Figure 176. Reset Counters Monitor Panel

Reset Counters Report

The RESET database operator command is used by this monitor. If the report option (see “Monitor Scheduling Panel” on page 365) is selected, then the date and time of the counter reset is logged.

VMDSS Counter Internal Monitor (CI)

This monitor collects internal information about data spaces usage in a VM/ESA environment.

The COUNTER INTERNAL command will be issued at the frequency that you designate for this monitor. The information captured will be recorded in the monitor's report log, so that you can monitor changes in counter(s) values. Refer to Figure 177 on page 363.

```
 mm/dd/yyyy                      Control Center                      hh:mm:ss
*------------------------ Database Monitor Addition ------------------------*
| Command ==>                                                CTRLID: MSTRSRV1 |
|  Database        => SQLDBA                                  NODE:  VMSYSTM1 |
|  Monitor Name    => VMDSS                                                    |
|  Monitor Type    => INTERNAL COUNTERS                                        |
|                                                                              |
|  ALL COUNTERS  => NO            (YES/NO for ALL COUNTERS)                    |
|                (Indicate  ON or OFF next to each counter)                    |
|  ESASEG   => OFF    MAPSEGPG => OFF    REQMAPPG => OFF   MAPPG    => OFF    |
|  SAVESLD  => OFF    SAVEMXRN => OFF    SAVEBLK  => OFF   WAITSLD  => OFF    |
|  REFBLOCK => OFF    REFBPAGE => OFF    REFBSPAN => OFF   REFLST   => OFF    |
|  REFLPAGE => OFF    DIAG10   => OFF    DIAG10PG => OFF   SCDIAG10 => OFF    |
|  SCREF    => OFF    SCBLKREF => OFF    SCSPNREF => OFF   SAVEGNRL => OFF    |
|  SAVECHK0 => OFF    SAVECHK1 => OFF                                          |
|                                                                              |
|          Specify UP to 7 counter(s) to be monitored (or YES for ALL),        |
|          then press ENTER to CONTINUE or PF3 to CANCEL                       |
|                                                                              |
*------------------------------------------------------------------SQMMNCI-----*
 PF: 1 Help    3 End (Cancel)
```

Figure 177. VMDSS Internal Counters Monitor Panel

Specify only those counter(s) that you want to monitor, or specify the ALL COUNTERS option to have all the internal counters monitored. Place "ON" next to those counter names that you want to have monitored, and "OFF" next to the counter names that will not be monitored.

This monitor is used for report purposes only and does not perform condition notification; therefore, be sure to specify the report option for it.

Usage Consideration: The Reset Internal Counter command can be used to set values to zero.

Counter Internal Report

The COUNTER INTERNAL operator command is used by this monitor. If the report option is selected, the results of this command are logged.

VMDSS Counter Pool Monitor (CP)

This tool monitors the performance of VM Data Spaces Support and the standard database DASD I/O system.

The COUNTER POOL command displays two sets of counters. One is for storage pools using Data Spaces Support, a database directory using Data Spaces Support, and unmapped internal dbspaces. The other set is for pools using the standard database DASD I/O system and a directory using the standard system.

This monitor can be utilized to collect performance information for a single storage pool, the database directory, the unmapped internal dbspaces or all storage pools plus the database directory and unmapped internal dbspaces.

The COUNTER POOL command will be issued at the frequency you designate. The information captured will be recorded in the monitor's report log, so that you can monitor changes in the counter pool values. Refer to Figure 178 on page 364.

```
 mm/dd/yyyy                    Control Center                     hh:mm:ss
*------------------------ Database Monitor Addition ------------------------*
| Command ==>                                                CTRLID: MSTRSRV1 |
|  Database        => SQLDBA                                  NODE:  VMSYSTM1 |
|  Monitor Name    => DSSPOOL                                                  |
|  Monitor Type    => POOL COUNTERS                                            |
|  Counter Pool type => ___ (Enter one of the following to be monitored        |
|                                                                              |
|                                                                              |
|                               a Storpool number, or                          |
|                               DIR - for database directory, or               |
|                               UNM - for Unmapped internal dbspaces, or       |
|                               *   - for all storpools, database directory    |
|                                     and unmapped internal dbspaces.)         |
|                                                                              |
|  NOTE:  Remember to specify the report option.                               |
|                                                                              |
|         Enter Pool Type(s) to be checked by the monitor                      |
|         then press ENTER to CONTINUE or PF3 to CANCEL                        |
|                                                                              |
*------------------------------------------------------------------SQMMNCI-----*
 PF: 1 Help    3 End (Cancel)
```

Figure 178. VMDSS Pool Counters Monitor Panel

Specify only the counter pool type that you want to monitor; using a Storpool number, DIR (Directory), UNM (Unmapped Internal Dbspaces), or * to have all of the environment monitored (storage pools, the database directory, and unmapped internal dbspaces).

The counter pool monitor is used for report purposes only and does not perform condition notification; therefore, be sure to specify the report option.

Usage Consideration: The Reset Pool command can be used to set values to zero.

Counter Pool Report

The COUNTER POOL operator command is used by this monitor. If the report option is selected, the results of this command are logged.

Scheduling a Monitor Routine

You can specify or change monitor scheduling information each time you use the Add (A) or Modify (M) options from the Main Monitor panel. After indicating any conditions or thresholds to be scanned for, you will be presented with the panel shown in Figure 179 on page 365.

Monitor Scheduling and Rescheduling

The scheduling or rescheduling of a database monitor automatically occurs after the addition of a new monitor, or after the successful modification of an existing monitor.

Deactivating A Monitor

To deactivate (stop a monitor from running or being scheduled to run), use the Modify (M) option from the Main Monitor panel. On the monitor schedule panel (Figure 179), set the Active entry field to N; the monitor will be deactivated and removed from the schedule and will not be rescheduled to run until the Active entry field is set to Y.

Monitor Scheduling Panel

```
 mm/dd/yyyy                    Control Center                    hh:mm:ss
*------------------------- Database Monitor Addition -------------------------*
|Command ==>                                              CTRLID: MSTRSRV1    |
| Database        => SQLDBA                                NODE:  VMSYSTM1    |
| Monitor Type    => DATABASE LOG CHECK                                       |
| Monitor Name    => LOGCHECK                                                 |
| Active          => Y                          (YES/NO)                      |
|                                                                             |
| Run Days        => MON TUE WED THU FRI SAT SUN (MON,TUE,WED,...)            |
| Start Time      => 06:15                      (hh:mm)                       |
| Stop Time       => 24:00                      (hh:mm, >= start time)        |
| Frequency       => 15M                        (nnt, t=H or M HR/MIN)        |
| Notify - Detect => N                          (YES/NO)                      |
|                                                                             |
| Generate Report => Y                          (YES/NO)                      |
| Report Reset    => SUN                        (MON,TUE,WED,...)             |
| Send Reports To => MSTRUSR/VMSYSTM1         (Userid/Nodeid OR Nickname)     |
|           Enter all parameters and press PF2 to install MONITOR             |
|                                                                             |
*-----------------------------------------------------------------SQMNTR05-------*
 PF: 1 Help   2 Process (SAVE)   3 End (CANCEL)
```

Figure 179. Monitor Scheduling Panel

Entry Field	Description
Monitor	Fixed field; entry established on the previous panel.
Monitor Name	Fixed field; entry established on the previous panel.
Active	N is used to deactivate a monitor without having to delete the monitor completely. Y means the monitor is active and should be scheduled by the Control Center monitor scheduler.
Run Days	Indicates the days of the week that the monitor is to run. You can specify all days, specific days, or day ranges (MON-FRI).
Start Time	Indicates when the monitor is to start for the day(s) specified. If you specify a start time equal to the stop time, then the monitor will be run only one time.

Stop Time	Indicates when the monitor is to terminate for the day(s) specified. If you specify a start time equal to the stop time, then the monitor will be run only one time on the specified days.
Frequency	The frequency indicates how often the monitor will run during the time period between the start and stop times. Monitors can be scheduled to run with frequencies ranging from 1 minute to multiple hours.
Notify - Detect	Indicates that you want repeated notification of a **detected** condition. See “Who is Notified” on page 347 and “Authorization/Notification Parameters” on page 215. The default is N, which means the monitor will notify you with a note file and an online message when the condition is initially detected and when the condition changes. Set this entry field to Y if you want continued online messaging of a detected condition every time the monitor runs, regardless of whether the detected condition has changed or not.
Generate Report	A value of Y (default) in this entry field indicates that you want a report generated by this monitor (N not to generate a report). All reports will be stored on Control Center's 191 A-disk and can be viewed using the Report option from the main monitor panel.
Report Reset	You must indicate a day on which the monitor report will be refreshed. Refreshing involves sending the report to a user or nickname identified in the Send Reports To entry field, after which the report log entries are deleted. This is done to limit the size of reports kept on Control Center's 191 A-disk.
Send Reports To	Prior to resetting a report (see “Report Reset” description above), the monitor report will be sent to the user or CMS nickname identified by this entry field.

Chapter 30. DBSPACE Reorganization Tools

Overview

The fundamental purpose of the DBSPACE Reorganization tools is to provide a means to reorganize DBSPACES within databases. This function is necessary to provide optimal database performance and to allow control and flexibility during database growth.

The DBSPACE Reorganization tools consist of SQLREORG for multiple user mode (MUM) DBSPACE reorganizations, a single user mode (SUM) DBSPACE reload tool, and a full single user mode DBSPACE reorganization tool. The SUM tools use the DDL generator portion of the SQLREORG tool as part of their processing. The remarks about the SQLREORG tool can also be generally applied to the SUM tools as far as the actual reorganization part goes. The SUM tools also encompass other processing in addition to the reorganization function.

Prior to using the Control Center DBSPACE Reorganization tools, review the *DB2 Server for VM Database Administration* manual for an understanding of the DBSPACE reorganization process.

How the DBSPACE Reorganization Tools Work

Processing Flow

When the Multiple User Mode DBSPACE Reorganization tool (SQLREORG) is invoked, these steps will be performed:

1. Link and establish database communications to the target database.
2. Verify that you have DBA authority.
3. Grant temporary DBA authority to a temporary user ID on the target database.
4. Connect as temporary user ID.
5. Verify availability of the specified new DBSPACE.
6. Create an EXEC for recovery purposes.
7. Gather all system catalog information pertaining to the specified DBSPACE and create corresponding database statements within the Control Center Database Services Utility (SQLDBSU) command file in the following order:

    a. Table create statements
    b. Table reload statements
    c. Table comments
    d. Column comments
    e. Referential integrity constraints
    f. Unique column definitions
    g. Index create statements
    h. Table grants
    i. Table column grants

j. View creates/grants/comments/labels
k. Access Module unload/reload statements

8. Define and format a temporary disk (if required).
9. Unload all DBSPACE data to the specified disk or tape.
10. If the PAUSE option was specified; stop execution, otherwise continue.
11. Send a copy of the unloaded DBSPACE to your reader if a temp disk has been used for the unload.
12. Execute the SQLDBSU command file under the Database Services Utility.
13. Revoke DBA authority from the temporary user ID.
14. Detach the temporary disk (if used).
15. Purge the unloaded data file from your reader.
16. Reprep (or UNLOAD/RELOAD) any invalidated access modules.

All grants are issued in the same chronological order in which they were originally issued, in order to retain hierarchical dependencies.

In order to grant authority to an object, SQLREORG must first connect as the user who originally issued the grant. Therefore, Control Center must gather database connect passwords for all grantors. If a grantor does not have a connect password, SQLREORG will give the user a temporary one and will later remove it.

The database does not remove grant information from the system catalogs when a user is removed from the SYSTEM.SYSUSERAUTH table. Therefore, the possibility exists that SQLREORG needs to connect as a nonexistent user in order to re-establish a grant. If this situation occurs, SQLREORG will temporarily grant connect authority to the user and will later revoke it.

Operational Note: In some cases (such as reload failure), temporarily granted IDs will not be revoked from the database. The administrator should revoke these IDs at some point in time. The IDs are identified by the starting letters REO*nnnnn* (where *nnnnn* is some random number).

Related files

Files created during the reorganization will have a filename that is similar to the DBSPACE name that is being reorganized. If the dbspace name has any special characters, they will be removed. If the DBSPACE name is longer than eight characters, then it will be truncated to eight.

If the ALTERNATE FILENAME parameter is used for a MUM reorganization, then the filename designated below as *DBSPACEname* will be that chosen by the submittor.

The basic files created will be:

Filename	Description
dbspacename UNLOAD	The UNLOAD SQLDBSU command - used as input to the SQLDBSU (unload DBSU command file).
dbspacename UNLIST	The output listing of the UNLOAD DBSPACE portion of the reorganization (UNLOAD DBSU execution listing).

dbspacename DBSPACE	The DBSPACE in unloaded format - an output of the SQLDBSU. It would be needed in the event of recovery from a failed reload step of the reorganization. It can also be used as a point-in-time backup copy of a DBSPACE.
dbspacename SQLDBSU	The SQLREORG-generated SQLDBSU input file containing all the commands needed to reorganize the DBSPACE.
dbspacename LISTING	The output listing of the RELOAD portion of the reorganization (DBSU execution listing).
dbspacename PROGDBSU	The input to the Database Services Utility (DBSU) to unload the access modules.
dbspacename PROGLIST	The output listing of the unload access modules step.
dbspacename PROG*nnnn*	These files contain the unloaded access modules (packages).
dbspacename EXEC	Exec used to recover from an interrupted reorganization.

The disposition of these files depends on whether the reorganization is successful or fails and on whether a temporary disk is used or a permanent disk is available during the reorganization. In all cases, the files needed for recovery from a failed reorganization (the EXEC, DBSPACE, SQLDBSU) and those pertaining to access modules (packages), if present, will always be available on either your A-disk or on the permanent disk specified, and in the reader of the user running the reorganization if temp disk was used. The other files will be on the A-disk or on the permanent disk, if used.

Figure 180 on page 370 is an example of the exec that will be created when SQLREORG executes, providing recoverability for the reorganization process. This exec will connect to the database and complete the second phase of the reorganization process (Create/Reload), regardless of when the failure occurred in the original reorganization. This exec is also used to complete a reorganization when the PAUSE option is specified.

```
/*************************************************************/
/*                  SQLREORG RECOVERY EXEC                   */
/*                                                           */
/*    This exec is used to recover from a failed SQLREORG.   */
/*  It will attempt to recreate and reload a DBSPACE from    */
/*  files saved on disk by SQLREORG.                         */
/*************************************************************/
/*  The files required for recovery are:                     */
/*      1) DBSPNAME DBSPACE C4                               */
/*      2) DBSPNAME SQLDBSU A                                */
/*      3) DBSPNAME PROGnnnn A (for any packages)            */
/*************************************************************/
trace 'o'
say 'Reorganizing DBSPACE MSTRUSR.DBSPNAME on database SQLDBA...'
say 'Using: MSTRUSR DBSPACE C'
'ESTATE MSTRUSR DBSPACE C'
estrc = RC
if estrc ¬= 0 then do
                  say 'ERROR! ==> UNLOADED DBSPACE NOT FOUND!'
                  EXIT -5
                end
 conn = '
 'QUERY' userid() '(STACK'
 if queued() > 0 then pull . '-' conn
 if conn ¬= 'DSC'
   then do
          say '  Do you wish to continue (Y/N)?'
          pull ans
          if substr(ans,1,1) ¬= 'Y' then exit
        end
starttime = time('E')
'DBINIT SQLDBA'
 if RC ¬= 0 then do
                   say '  ERROR during DBINIT to SQLDBA'
                   exit RC
                 end
'SET CMSTYPE HT'
'FILEDEF DBSFILE CLEAR'
'FILEDEF ARISQLLD CLEAR'
'SET CMSTYPE RT'
'FILEDEF DBSFILE DISK MSTRUSR DBSPACE C4 (RECFM VBS BLKSIZE 4097'

'SQLDBSU IN(MSTRUSR SQLDBSU A) SYSPRINT(USERA LISTING A)'
erc = RC
if erc ¬= 0 then do
                  say 'ERROR running SQLDBSU, RC=' RC
                  say 'Check MSTRUSR LISTING for error details.'
                end
```

Figure 180 (Part 1 of 2). Example Reorganization Recovery Exec

```
 else do
   stoptime = time('E')
   sttime = trunc(starttime,2)
   sptime = trunc(stoptime,2)
   elaptime = trunc((sptime - sttime) / 60,2)
   tlaptime = elaptime +  2.04
   tlaptime = strip(format(tlaptime,5,0))
   if tlaptime = 0 then tlaptime = 1
   sqlmaint_cmd = 'UPDATE SQLMAINT.SQLMAINT_TABLE ',
   "SET REORG_STATUS = '00', REORG_DATE = '"||date('J')"',",
   "REORG_TIME = '"time()"', REORG_ELAPSED = "tlaptime",",
   "UPSTAT_TIME = '"time()"', UPSTAT_DATE = '"date('J')"'",
   "WHERE DBSPACENAME = '"M760595"' AND OWNER = '"M760595"'"
   SET CMSTYPE HT
   ERASE $SQLREO$ $OUTPUT$ A
   FILEDEF SYSPRINT CLEAR
   fdef = 'FILEDEF SYSPRINT DISK $SQLREO$ $OUTPUT$ A'
   fdef = fdef || ' (RECFM V LRECL 120'
   ADDRESS CMS fdef
   'SET CMSTYPE RT'
   'EXEC ARISQLLD'
   'SET CMSTYPE HT'
   'NUCXDROP ARIDBS'
   'SET CMSTYPE RT'
   'NUCXLOAD ARIDBS ARIDBS ARISQLLD'
   dcmd = 'ARIDBS PROMPTS(NO) MESSAGES(SQLONLY)'
   dcmd = dcmd || ' FORMAT(LO) PAGECTL(NO)' sqlmaint_cmd
   ADDRESS CMS dcmd
   dbsurc  = RC
   'NUCXDROP ARIDBS'
   if dbsurc ¬= 0
     then do
       say ' ERROR during update to SQLMAINT table'
       exit dbsurc
       end
   end
'SET CMSTYPE HT'
'ERASE $SQLREO$ $OUTPUT$ A'
'FILEDEF SYSPRINT CLEAR'
'FILEDEF SYSPRINT DISK $SQLREO$ $OUTPUT$ A (RECFM V LRECL 255'
'SET CMSTYPE RT'
'EXEC ARISQLLD'
'SET CMSTYPE HT'
'NUCXDROP RXSQL'
'NUCXDROP ARISRRM'
'NUCXLOAD ARISRRM ARISRRM ARISQLLD'
'ARISRRM ENDSBCMD'
'NUCXDROP ARISRRM'
'NUCXDROP ARIDBS'
'SET CMSTYPE RT'
'NUCXLOAD ARIDBS ARIDBS ARISQLLD'
'ARIDBS PROMPTS(NO) MESSAGES(SQLONLY) FORMAT(LO) PAGECTL(NO)
 REVOKE CONNECT FROM REO55553;'
'NUCXDROP ARIDBS'
'SET CMSTYPE HT'
'ERASE $SQLREO$ $OUTPUT$ A'
'SET CMSTYPE RT'

EXIT erc
```

Figure 180 (Part 2 of 2). Example Reorganization Recovery Exec

Figure 181 on page 372 is a simple example of the SQLDBSU file created when SQLREORG executes. This SQLDBSU command file contains all DDL statements associated with the reorganized DBSPACE and the RELOAD statements required to perform the reorganization.

For DBSPACES with very complex data structures (many tables, referential integrity, views, grants), the SQLDBSU file will contain considerably more commands, which would be very difficult to build manually without a comprehensive reorganization tool similar to SQLREORG.

```
COMMENT '******************** SQLREORG DBSU ********************'
COMMENT '*     DBSPACE: MSTRUSR.DBSPNAME (SQLDBA)              *'
COMMENT '*       Date: 03/14/97 09:52:23                       *'
COMMENT '*************************************************************'
GRANT DBA TO REO35543 IDENTIFIED BY ********;
COMMIT WORK;
CONNECT REO35543 IDENTIFIED BY ********;
COMMIT WORK;
COMMENT '******************** Drop DB Space ********************'
SET ERRORMODE (CONTINUE)
DROP DBSPACE "MSTRUSR"."DBSPNAME";
COMMIT WORK;
SET ERRORMODE (OFF)
SET AUTOCOMMIT (ON)
COMMENT '********************** Acquire DB Space ********************'
ACQUIRE PRIVATE DBSPACE NAMED "MSTRUSR"."DBSPNAME"
 (PAGES = 2048, PCTINDEX = 15,
 PCTFREE = 0, NHEADER = 8,
 LOCK = DBSPACE, STORPOOL = 9);
COMMENT '********************** Create Tables **********************'
CREATE TABLE  "MSTRUSR"."CPRCNTRL" (
  "OWNER" CHAR(8)
 ,"PROJECT" CHAR(8)
 ,"TBLID" CHAR(1)
 ,"PROJECT_NAME" CHAR(20)
 ,"XFER_USERID" CHAR(8)
 ) IN "MSTRUSR"."DBSPNAME";
COMMENT '*************      Deactivated Primary Keys      *************'
COMMENT '*************    Inactive Unique Constraints     *************'
COMMENT '********************** Reload Tables **********************'
RELOAD TABLE("MSTRUSR"."CPRCNTRL")
 PURGE
 INTABLE("MSTRUSR"."CPRCNTRL")
 INFILE(DBSFILE);
CONNECT REO35543 IDENTIFIED BY ********;
COMMENT '*************           Comment Tables           *************'
COMMENT '*************          Comment Columns           *************'
COMMENT '*************        Primary Cluster Keys        *************'
COMMENT '*************  Clustering Unique Constraints     *************'
COMMENT '********************** Create Indexes **********************'
CREATE INDEX "MSTRUSR"."CPR1110X"
 ON "MSTRUSR"."CPRCNTRL"
 ("OWNER")
 PCTFREE = 0;
COMMENT '*************           Primary Keys             *************'
COMMENT '*************       Remaining Foreign Keys       *************'
COMMENT '*************      Active Unique Constraints     *************'
COMMENT '*************           Table Grants             *************'
CONNECT MSTRUSR IDENTIFIED BY ********;
 GRANT SELECT ON "MSTRUSR"."CPRCNTRL" TO "PUBLIC";
 GRANT UPDATE ON "MSTRUSR"."CPRCNTRL" TO "APCSDEV";
GRANT SELECT, INSERT, UPDATE, DELETE, ALTER, INDEX ON
 "MSTRUSR"."CPRCNTRL" TO "APCSDEV";
CONNECT REO35543 IDENTIFIED BY ********;
COMMENT '*************          Column Grants             *************'
COMMENT '*************       Views with Grants            *************'
COMMENT '********************  PACKAGE REBIND ********************'
SET AUTOCOMMIT (ON)
```

Figure 181. Example SQLDBSU Command File

Who Can Use DBSPACE Reorganization Tools

Use of the DBSPACE Reorganization tools requires Control Center Database administration-level or greater authorization.

Multiple User Mode DBSPACE Reorganization Tool

Although SQLREORG can be selected from the Control Center panel interface, it is actually a product that is designed to execute from the CMS command level. It can be invoked in panel mode by either entering SQLREORG at the CMS READY prompt, or by using Option **U** (Database Utilities) on the Control Center Main Menu and then entering Option **R** (SQLREORG) or **RS** (REORG SINGLE USER). Additionally, SQLREORG can be invoked through Control Center's Search List tool, see Figure 217 on page 454. If invoked through SL, the PCTFREE, ALTER PCTFREE, and PCTINDEX reflect values used from the last reorganization.

SQLREORG can also be invoked in command mode (in BATCH) by supplying all parameters required to define the desired reorganization. This chapter only addresses the Control Center panel interface option. The other method of invoking SQLREORG is discussed in Appendix G, “Command Mode Interface” on page 523.

SQLREORG executes as a database application rather than as a function within the database virtual machine. SQLREORG therefore, normally executes on your virtual machine (or a support machine) and thus requires that you (or the support machine) have DBA authority within the target database. SQLREORG is also invoked (used) by the SQLMAINT tool when performing reorganizations.

Before You Begin

These comments apply to a full reorganization of a DBSPACE and not to the PAUSE option.

SQLREORG should be used during non-peak hours to prevent locking contention with other users of the database. Due to the intensive updating of the database system catalogs, use of this utility is prone to lock contention during heavy multi-user sessions. Running more than one SQLREORG against different DBSPACES simultaneously within a single database should also not be done due to catalog contention.

SQLREORG should be used whenever database statistics indicate that a DBSPACE is in need of reorganization. This need is usually indicated when indexes are no longer clustered or when a considerable amount of delete activity has occurred, leaving holes of deleted data. The SQLMAINT tool can help determine when DBSPACES need reorganization; see “DBSPACE Reorganization Maintenance Tool” on page 437.

SQLREORG should also be used when a larger DBSPACE is needed due to a growth in the volume of data within a DBSPACE, or when a performance problem indicates that heavily used DBSPACES should be moved to storage pools with less usage. This movement can also solve a storage pool short-on-storage problem without adding a new dbextent to the database.

Installations that want to schedule reorganization and maintenance jobs should implement a separate Control Center support machine to provide the scheduling

capability without impacting the database operation control functions of the Control Center service machine. SQLREORG jobs should *not* be scheduled to execute on the service machine, since all automated operations functions with *all* databases will be disabled while the maintenance functions execute. Although SQLREORG can be executed in batch mode with any scheduling product, a separate Control Center support machine will allow these jobs to be scheduled easily from the Control Center panel interface.

Features

The SQLREORG tool has these features:

- Frees up unused pages by dropping and re-acquiring DBSPACES
- Saves and restores these database objects related to a DBSPACE:
  - Tables
  - Data
  - Referential Integrity constraints
  - Unique column definitions
  - Indexes
  - Views
  - Grants
  - Table and Column Comments
  - Table and Column Labels
  - Packages (Access Modules)
- Loads data in clustering index sequence
- Loads data with 10% free space for future inserts
- Rebuilds clustered indexes (where possible)
- Updates statistics
- Repreps invalidated access modules
- Allows these options:
  - Reload DBSPACE to a different database
  - Reload DBSPACE with a different owner
  - Reload DBSPACE with a different dbspace name
  - Use DBSPACE in a new storage pool
  - Use different size DBSPACE
  - Change DBSPACE header size
  - Change free space percent
  - Change index percent
  - Change lock mode
- Can run in Multiple or Single User Modes

Single user mode usage of the SQLREORG tool is a special topic which is covered later. Many components of Control Center are involved in a Single User Mode reorganization and the entire process is explained in detail.

Multiple User Mode DBSPACE Reorganization Entry Panel

Shown in Figure 182 is the entry panel of the Multiple User Mode DBSPACE Reorganization tool.

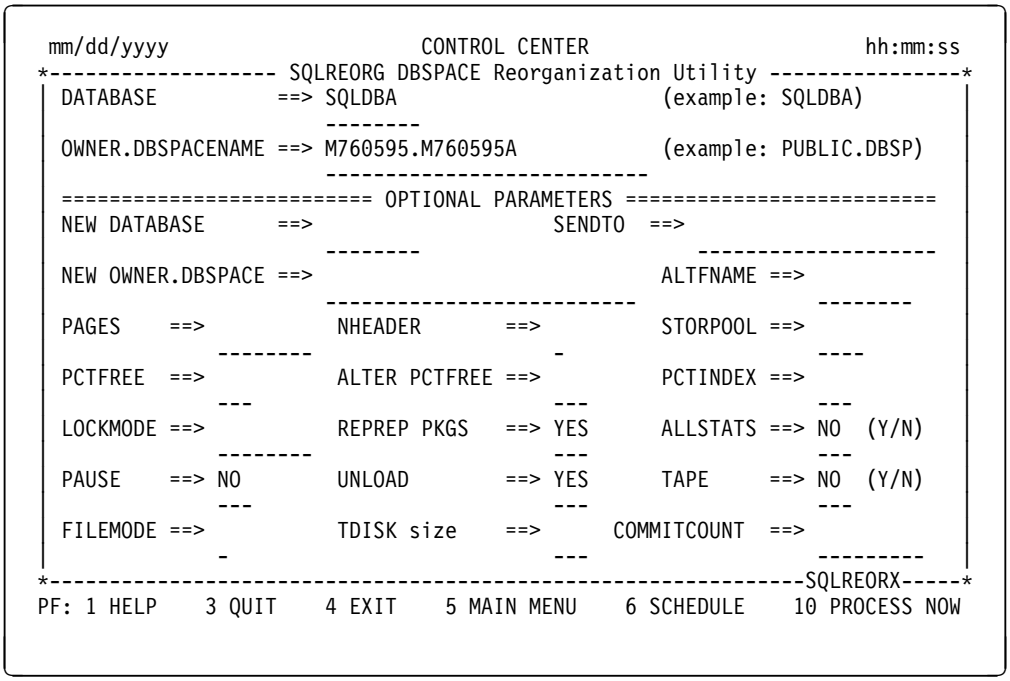


Figure 182. SQLREORG Screen

The first two parameters (DATABASE and OWNER.DBSPACENAME) are required to identify the DBSPACE that is to be reorganized. The database specified must match either the database parameter or the nickname parameter of some entry in the DBINIT CONTROL file. The OWNER.DBSPACENAME parameter must identify a valid DBSPACE within the targeted Database.

Optional Parameters

The parameters below the line labeled "Optional Parameters" do not require an entry.

Parameter	Description
NEW DATABASE	Used to specify that the DBSPACE should be reloaded to a different database. This option allows a DBSPACE to be migrated from one database to another, such as from development to production. Due to differences within the two databases, such as user IDs within the SYSTEM.SYSUSERAUTH table, certain reload statements cannot execute successfully unless you first assure that the two environments are compatible. When the NEW DATABASE parameter is used, the DBSPACE within the old database will remain unchanged.

SENDTO	Used to specify the ID to which the unloaded file is sent. The default value is the ID of the user running the SQLREORG. Specify the value As "userid" or "userid AT node".
NEW OWNER.DBSPACE	Used to specify a new owner or a new DBSPACE name for the reloaded DBSPACE. When the source dbspace and the target dbspace are both PRIVATE, the old DBSPACE will remain unchanged. If either or both DBSPACES are PUBLIC, then the old DBSPACE will be DROPPED prior to the creation of the new DBSPACE.
ALTFNAME	Used to specify a filename of the user's choice instead of the default one built from the DBSPACE name. This is useful for those environments which have more than one DBSPACE starting with the same 8 character name.
PAGES	Used to specify a new DBSPACE page size for the reorganized dbspace. An empty (unacquired) DBSPACE of the indicated number of pages must be available within the database. If this parameter is not specified, a DBSPACE equal in size to the current dbspace will be acquired.
NHEADER	The number of pages in a DBSPACE reserved for dbspace header information. The value entered must be a number between 1 and 8. If the number chosen is smaller than what is required for all header information, the reload could fail.
STORPOOL	Used to specify a new storage pool for the acquired DBSPACE. This option allows you to balance the I/O load of a database by spreading the most active DBSPACES over multiple packs of DASD. Non-recoverable pools should be indicated with a negative sign in front of the storage pool number.
PCTFREE	Use to indicate a percent of each DBSPACE page to be reserved for later insertion. The default for this parameter is 10 percent. After SQLREORG reloads all data into the DBSPACE, the PCTFREE will be dropped to zero to allow insertion.
ALTER PCTFREE	Used to indicate a PCTFREE value to be used AFTER the data has been reloaded to the DBSPACE. This value should be lower than the PCTFREE parameter value to allow future inserts to use reserved space on existing DBSPACE pages, allowing cluster properties to be maintained. If ALTER PCTFREE is not indicated on the panel, the default of 0 will be used.
PCTINDEX	Used to change the ratio of data pages and index pages within a DBSPACE. If not specified, the same ratio as the original dbspace will be used.

LOCKMODE	Use to change the lock mode of a DBSPACE. Valid choices for PUBLIC DBSPACES are DBSPACE, PAGE, and ROW.
REPREP PKGS	The SQLREORG tool repreps (rebinds) all access modules (packages) during a DBSPACE reorg. If this is not wanted, specify NO to this field. The default value is YES.
ALLSTATS	By default, SQLREORG will use the UPDATE STATISTICS command for the DBSPACE. If YES is specified for the ALLSTATS parameter, SQLREORG will use the UPDATE ALL STATISTICS command.
PAUSE	Allows you to interrupt the reorganization process between the UNLOAD and RELOAD steps. By specifying YES, SQLREORG will unload all data and capture all DBSPACE objects but will not drop the dbspace. An exec will be created which will complete the reorganization when executed. The name of this exec will be the first eight alphanumeric characters of the DBSPACE name. This option must be used with caution because some files are on temporary disk. If you logoff, these disks will be released and the files will be lost.
UNLOAD	Allows the DBSPACE DDL to be captured without unloading the data. This option is only available with the PAUSE option.
TAPE	Allows you to unload the DBSPACE to TAPE rather than DISK. By specifying YES, SQLREORG will expect you to have performed the appropriate FILEDEF, LABELDEF, and TAPE MOUNT commands to prepare for the SQLDBSU UNLOAD processing. Refer to “Special Considerations” on page 378 for details.
FILEMODE	Allows you to specify a linked minidisk that is to be used for the unloaded data. This minidisk must be linked in R/W mode prior to running SQLREORG. If this option is not specified, SQLREORG will define a temporary disk large enough to hold all unloaded data. If the same DBSPACE is reorganized to the same minidisk, only the restart exec, unload and SQLDBSU files are backed up; the unloaded DBSPACE (filetype of DBSPACE) is erased.
TDISK size	Used to ask for a larger allocation of temporary disk than that calculated by SQLREORG.
COMMITCOUNT	Allows you to specify a value from 1 to 2147483647 to be used in the RELOAD processing. A value of 0 means to disregard the parameter.

After entering all desired SQLREORG parameters, **PF10** can be used to perform the DBSPACE reorganization immediately.

Usage Consideration: When used in this manner, the tool will run on your virtual machine, thus tying up your ID for the duration of the reorganization job.

Optionally, **PF6** can be used to schedule the reorganization activity using the Job Scheduling tool. Refer to Chapter 14, “Job Scheduling Tool” on page 165.

Uses of SQLREORG

SQLREORG can be used to do a complete database reorganization of a DBSPACE for purposes of maintenance and performance improvement.

Another usage of this tool is for providing a backup (snapshot) of a DBSPACE. This is done by running the tool with the PAUSE option. This option creates all the DDL needed to restore a DBSPACE and unloads the data to a designated medium. In addition, a recover exec is provided.

Running the tool with the PAUSE option *and* the UNLOAD=NO option provides all the DDL associated with a given DBSPACE. This can be helpful in analyzing grants and indexes. This file can also be modified and used as input to a subsequent SQLDBSU run.

Special Considerations

Unloading to Tape

When using the TAPE option, you must issue the appropriate FILEDEF, LABELDEF, and TAPE MOUNT commands prior to invoking SQLREORG. The DDNAME for the unloaded data must be DBSFILE. SQLREORG executes the UNLOAD utility. For UNLOAD, always specify a record format (RECFM) of variable-length blocked spanned (VBS). For instance, an example of a FILEDEF and LABELDEF commands would be:

```
FILEDEF DBSFILE TAP1 SL (RECFM VBS BLKSIZE 28672
LABELDEF DBSFILE VOLID SCRATCH
```

If the unloaded DBSPACE will fit on a single tape, then SQLREORG can be executed without the PAUSE option, there is no need to rewind the tape. It will automatically be rewound between the UNLOAD and RELOAD steps.

If the unloaded DBSPACE will span multiple tapes, then the proper LABELDEF command must be issued to handle the multi-volume tape. The PAUSE option must be specified so that the appropriate sequence of DETACH and MOUNT commands can be issued between the UNLOAD and RELOAD processing. After the PAUSE, the last tape must be detached and the first tape must again be mounted before the reload process can begin.

Failure Restart

SQLREORG will display status information at your terminal to indicate what step is being performed. If SQLREORG fails at any time prior to the "Reloading..." message, the DBSPACE has not been changed and SQLREORG should be restarted from the beginning. If the failure occurs during the "Unloading..." step and you are using a temporary disk, you can detach the temporary disk by issuing the CMS RELEASE command with the DET option.

If SQLREORG fails at any time after the "Reloading..." message has been displayed, the DBSPACE must be recovered using the created exec. This exec will begin the reload process by using the SQLREORG SQLDBSU file and the unloaded data file. If the data was unloaded to TAPE, then the proper FILEDEF, LABELDEF, and tape MOUNT commands must be used before the created exec can be invoked.

NOTE

If SQLREORG unloads data to a temporary disk and SQLREORG fails, you must be aware that the temporary disk (and unloaded data) will be purged if you log off. Therefore, the created exec must be run successfully before logging off.

As a special precaution to prevent loss of data, SQLREORG sends the unloaded data file to your reader. If a temporary disk is lost, the reader copy can be received onto a new disk and the created exec can then be executed. Under this scenario, you should verify that all file mode references in the created exec correctly point to the new location of the data file.

If you used the COMMITCOUNT option, the RELOAD TABLE statements generated by the tool in the SQLDBSU file, will have to be inspected and removed for the failing table prior to using the RESTARTCOUNT option. In general, care must be used in restarting a failed RELOAD. The entire SQLDBSU file must be examined closely and altered accordingly if you wish to restart at any specific point rather than to rerun the entire RELOAD process (which option is always available).

Pause Option

The same cautions discussed in “Failure Restart” also apply to the use of the SQLREORG PAUSE option. Each time that SQLREORG is run, a new SQLDBSU and exec file will be created. Any previous copy of these files will be renamed by reversing the filename and filetype and reversing all letters in each. This implies that if a second SQLREORG is executed prior to completion of the first SQLREORG, the first one can be used again if the files are named back again.

The PAUSE option can be very useful to an experienced user of SQLREORG and very dangerous to an inexperienced user. If you have a good working knowledge of the Database Services Utility and of SQLREORG, the PAUSE option will allow a great deal of flexibility and control in the reorganization process. The SQLDBSU file can be examined and modified prior to running the created exec in order to change or remove certain characteristics of the DBSPACE. Once again, extreme care should be exercised to prevent loss or contamination within the database.

Unloaded Data File

SQLREORG will unload the database data within a DBSPACE to an OS format file on a CMS minidisk or tape using the DBS Utility UNLOAD command. This unload format is defined internally by the unloaded data file, therefore, it can not be altered in any way. Extreme caution should be exercised to ensure that this data file is not altered prior to successful completion of SQLREORG. The file should not be edited (through XEDIT or ISPF, for example). The filemode must always be defined with a filemode number of 4 (for OS simulated dataset).

Failure Analysis

During the capture portion of SQLREORG processing, SQL statements are used to capture information from the database system catalogs. If any SQL statement fails, SQLREORG will terminate and display the SQLCODE associated with the failure.

SQLREORG uses a DBS Utility command file to execute the data UNLOAD portion of the reorganization. Detailed output from the UNLOAD portion is captured in a file on your 191 disk with a file type of UNLIST. This file can be examined to determine the reason for failure.

During the "Reloading..." portion of SQLREORG processing, the SQLDBSU command file is executed using the DBS Utility (SQLDBSU). Detailed output of this processing is captured within a file on the target disk with a filetype of LISTING and a filename equal to the first eight alphanumeric characters of the DBSPACE name. If a failure occurs during this step of processing, this file can be examined to determine the cause of failure.

One common problem to be aware of is the possible LOG FULL condition that may occur during the RELOAD processing. The DBS Utility RELOAD TABLE command executes as a single LUW, which means that the entire reload of all rows is capable of being rolled back if an error occurs. The database must therefore record the entire LUW within the database LOG.

If the target table is very large, or the database LOG file was nearly full when the reorganization began, the possibility for a LOG FULL condition exists. Depending on the logmode of the database, it will either attempt to perform a full archive, a log archive, or a checkpoint within the LOG.

If the RELOAD process continues until the LOG file completely fills, then the database will begin to ROLLBACK the entire RELOAD. Since the DROP DBSPACE has already completed and has been COMMITTED to the database, the target DBSPACE will be in an incomplete state if this occurs. There are several possible solutions to this problem.

If the RELOAD failed because the log file was nearly full prior to the reorganization, then you can perform a log archive, full archive, or coldlog (depending on whether you are using logmode L, A, or Y respectively). After this completes, you can run the DBSPACE exec created by SQLREORG to restart the process.

Note: If you are using a temporary disk, be careful not to lose it by logging off.

If the RELOAD cannot fit into the LOG file, even if empty, you have three options:

1. Increase the size of the LOG file, then complete the reorganization.
2. Run the RELOAD in SUM with logmode N (no logging).

3. Move the DBSPACE to a nonrecoverable (non-logging) storage pool by changing the ACQUIRE statement within the SQLDBSU file to the appropriate storpool.

Single User Mode DBSPACE Reload Tool

The Single User Mode DBSPACE Reload tool provides a method of reloading a database DBSPACE in SUM without logging onto the database. This tool reloads the data from a DBSPACE previously unloaded by the MUM SQLREORG tool.

Before You Begin

Review the *DB2 Server for VM Database Administration* manual for an understanding of the database running in single user mode. For the SUM reload, the database machine ID must have DBA authority in the DB2 Server for VM database. In addition, it must be enabled for link and access authority to the disks where the files it uses reside.

Single User Mode DBSPACE Reorganization Panels

The following three figures show the panels which you will be presented with when invoking the Single User Mode DBSPACE Reorganization tool.

```
 mm/dd/yyyy                    CONTROL CENTER                        hh:mm:ss
*-------------------- Single User Mode DBSPACE Reload ----------------------*
| Command ==>                                                CTRLID: MSTRSRV1 |
|                    ****  I M P O R T A N T  ****            NODE:  VMSYSTM1 |
|                                                                            |
|    In order to Reorganize a DBSPACE in Single User Mode, you must first    |
|    use the SQLREORG package in multi-user mode, with the PAUSE option,     |
|    and with output placed on a PERMANENT disk that can be linked to        |
|    by the database machine (DIRMAINT or RACF authorized).                 |
|                                                                            |
|    You must assure that no updates have occurred within the DBSPACE        |
|    after the SQLREORG PAUSE step and prior to the CTRLCTR REORG step .     |
|                                                                            |
|    This function will switch the database from multi-user mode to          |
|    single user mode with a specified logmode. If logmode N is requested,   |
|    no logging of the updates will be done, and CTRLCTR will automatically  |
|    handle all database requirements to switch the logmode to N and back,   |
|    including all archives and coldlogs.                                    |
|                                                                            |
|        Press ENTER to continue or press PF3 to CANCEL the reload           |
*-----------------------------------------------------------------SQMSUMR1----*
 PF:   1 Help    3 End (Cancel)
```

Figure 183. Single User Mode DBSPACE Reload Usage Information

```
 mm/dd/yyyy                    CONTROL CENTER                      hh:mm:ss
*-------------------- Single User Mode DBSPACE Reload ----------------------*
| Command ===>                                                CTRLID: MSTRSRV1 |
|    Database ===> SQLDBA                                      NODE:  VMSYSTM1 |
|                                                                              |
|    Userid       ===> MSTRUSR        Userid of SQLREORG file output           |
|    Disk Address ===> 191            Disk address of SQLREORG output          |
|    File name    ===>                SQLREORG output file name                |
|    Logmode      ===>                Logmode for s.u.m. operation,            |
|                                     defaults to current logmode              |
|                                                                              |
|     Enter the Userid and Disk Address where the SQLREORG output files       |
|     (SQLDBSU and DBSPACE) are located.  Also enter the File Name of          |
|     the output files and what logmode you desire (remember that using       |
|     logmode N will require archives/coldlogs/log archives).                 |
|                                                                              |
|                                                                              |
|        Press ENTER to select IMMEDIATE or SCHEDULED execution, or           |
|        press PF3 to CANCEL the Single User Mode reload                       |
|                                                                              |
|                                                                              |
*------------------------------------------------------------------SQMSUMR2----*
 PF:   1 Help    3 End
```

Figure 184. Single User Mode DBSPACE Reorganization Entry Panel

```
 mm/dd/yyyy                    CONTROL CENTER                      hh:mm:ss
*-------------------- Single User Mode DBSPACE Reload ----------------------*
| Option ===>                                                 CTRLID: MSTRSRV1 |
|    Database ===> SQLDBA                                      NODE:  VMSYSTM1 |
|    Reorg    ===> MSTRUSR 191 DBSPNAME                                        |
|                                                                              |
|     I  INITIATE RREORG            Immediate Single User Mode reload          |
|                                                                              |
|     S  SCHEDULE RREORG            Schedule later reload                      |
|                                                                              |
|                                                                              |
|                                                                              |
|                                                                              |
|                                                                              |
|                                                                              |
|           Enter OPTION and press ENTER to process, or                        |
|           press PF3 to CANCEL the Single User Mode reload                    |
|                                                                              |
*------------------------------------------------------------------SQMSUMR3----*
 PF:   1 Help    3 End
```

Figure 185. Single User Mode DBSPACE Reorganization Options Panel

SUM DBSPACE reload process will require you to perform two separate processing steps: first, unloading the data in multiple user mode and then reloading the data in single user mode.

Prior to invoking the panel interface, you should use the SQLREORG utility in MUM with two special options: PAUSE and DISK. The PAUSE option will cause

SQLREORG to complete the capture and unload of the DBSPACE but stop the process prior to the drop and reload function. The DISK option must be used to place the unloaded DBSPACE on a permanent linked disk. The default within SQLREORG is to use a temp disk, which will not allow the single user reload to be completed.

The database machine must have link access to the disk that is used for the unloaded DBSPACE. After the unload portion of SQLREORG has completed, then the panel interface can be used to complete the SUM reorganization. This function will ask for the unloaded file name (given by SQLREORG), the user ID which owns the disk where the DBSPACE was unloaded, and the address of the disk.

This process will cause the database to be brought down, the minidisk to be linked, and the reload to be executed in SUM with logmode N (no logging). After the reorganization completes, the database will be brought back up with whatever logmode the database was running under previously. Switching from logmode N to logmode Y, A, or L will require some combination of coldlogs, archives, or log archives. These activities will be performed automatically by Control Center once the SUM activity completes.

If you wish to perform multiple SUM reloads and only a single archive, you will need to issue the SQLEND command to the database before invoking the SUM Reorganization function. Control Center will always return the database to the same state that it was in prior to the invoked function. Therefore, if the database is down when the SUM process is performed, the archive will be delayed until the STARTUP command is issued. You can, therefore, perform as many SUM activities as desired prior to bringing the database back up and causing the archive process to begin.

Full Single User Mode DBSPACE Reorganization Tool

The SQLREORG utility provides an optional full Single User Mode DBSPACE Reorganization tool that will process both the unload and reload reorganization steps in Single User Mode execution.

Before You Begin

Review the *DB2 Server for VM Database Administration* manual for an understanding of the database running in single user mode. For the SUM tool, the database machine ID must have DBA authority in the database.

How the Single User Mode Reorganization Works

The SQLREORG utility is executed in Single User Mode on the database virtual machine to reorganize a specified DBSPACE. By performing the reorganization in Single User Mode, contention with interactive users can be avoided. Also, reorganizations for very large DBSPACES can execute with logmode N (No Logging) to avoid the logging overhead of Multiple User Mode, and also enable reorganizations of DBSPACES that cannot execute in multiple user mode because the data reload completely fills the database log disk, causing the reorganization to fail.

The Single User Mode functions of SQLREORG are identical to those of Multiple User Mode, with a few specific exceptions. When SQLREORG is invoked in single user mode by using the "REORG SINGLE USER" (SU) option on the UTILITIES

menu of the Control Center panel interface, a special "Single User Mode Reorganization Utility" panel will be presented. This SUM panel has two new parameters and eliminates two other parameters available in multiple user mode.

The first new parameter available in single user mode is LOGMODE. The value presented on the panel is the current logmode of the database. It can be changed to "N" if you wish to perform the reorganization without logging (logmode N). Choosing logmode N will require a logmode change for the database, with an accompanying set of archives, log archives, and/or coldlogs (depending on the current logmode before switching to logmode N). Control Center will handle these requirements automatically, but they may impose a considerable amount of additional time, especially if your database is quite large, and a full database archive is necessary.

Another parameter is STAYDOWN, which defaults to NO (N) but can be changed to YES (Y). Normally, when functions execute on the database virtual machine, the database will be returned to Multiple User Mode operation (available to all users) when the function completes. The STAYDOWN option can be used to force the database to remain DOWN after the reorganization completes, rather than starting the database back up. This can be useful when multiple reorganizations are done in single user mode on the database machine. A number of reorganizations can be scheduled at the same time, dependent on successful completion of the previous job, with all but the last one specifying STAYDOWN=YES. Each SUM reorganization will then execute in turn, with the database returning to normal Multiple User Mode operation only after the last one completes.

The two eliminated parameters are PAUSE and UNLOAD since they are not applicable to this tool.

If the TAPE parameter is specified as Y, an additional panel is presented. This is discussed in detail below.

Who Can Use the Single User Mode DBSPACE Reorganization Tool

To invoke SQLREORG in single user mode, Control Center database administrator (Level 3) or greater authority is required. Additionally, the database virtual machine must have DBA authority within the database.

Features

The SQLREORG SUM Tool has these features:

- Allows reorganization with/without logging
- Unloads to tape, permanent or temporary disk
- Saves and restores database objects related to a DBSPACE
- Loads data in clustering index sequence
- Rebuilds clustered indexes (where possible)
- Includes the STAYDOWN option for more efficient multiple SUM executions
- Provides DBSPACE attribute modifications on reload (storpool, page size, number of header pages, lockmode, pctfree, and pctindex)
- Updates the SQLMAINT table, if SQLMAINT is installed
- Can be scheduled for later execution or initiated immediately

- Updates Statistics
- Repreps invalidated access modules

Single User Mode DBSPACE Reorganization Panels

The following two figures show the panels which you will be presented with when invoking the Single User Mode DBSPACE Reorganization tool.

```
 mm/dd/yyyy                    CONTROL CENTER                        hh:mm:ss
*------------------ Single User Mode Reorganization Utility ----------------*
| DATABASE          ==> CPMICDBA                  (example:  SQLDBA)        |
|                                                                           |
| OWNER.DBSPACENAME ==> M760595.M760595A          (example: PUBLIC.DBSP)    |
|                                                                           |
| ========================== OPTIONAL PARAMETERS ========================== |
| LOGMODE  ==> L  (defaults to current )    SENDTO ==>                      |
|              -                                  --------------------      |
| STAYDOWN ==> NO  (Y or N)                       ALTFNAME ==>              |
|              ---                                                 --------  |
| PAGES    ==>           NHEADER       ==>       STORPOOL ==>              |
|              --------                 -                          ----      |
| PCTFREE  ==>           ALTER PCTFREE ==>       PCTINDEX ==>              |
|              ---                      ---                        ---       |
| LOCKMODE ==>           REPREP PKGS   ==> YES   ALLSTATS ==> NO  (Y/N)     |
|              --------                 ---                        ---       |
| TAPE     ==> NO  (Y/N)                        COMCOUNT ==>              |
|              ---                                                 ---       |
| FILEMODE ==>    ( Linked R/W disk,       TDISK size ==>     (Defaults to   |
|              -  defaults to TEMP DISK )                  ---  calculation) |
*-------------------------------------------------------------SQLSUMR$-----*
 PF1 HELP   PF3 QUIT   PF4 EXIT   PF6 Schedule   PF10 Process immediately
```

Figure 186. Single User Mode DBSPACE Reorganization Options Panel

The first two parameters (DATABASE and OWNER.DBSPACENAME) are required to identify the DBSPACE that is to be reorganized.

Optional Parameters

The parameters below the line labeled "Optional Parameters" do not require an entry. The optional parameters not described just below are explained in the parameter descriptions for Figure 182 on page 375. The two different parameters for the SUM Reorganization screen are:

Parameter	Description
LOGMODE	Use this parameter to run the reorganization process under logmode N (without logging). The LOGMODE parameter should remain blank if you do NOT want to switch logmodes during the reorganization process; or use the defaulted current logmode setting of the database.
STAYDOWN	Specify YES or Y for this parameter if the database should remain DOWN after the reorganization. The default will be to return the database to normal multiple user mode operation after the reorganization.

Tape Management for Reorganization

Tape processing is fully supported in single user mode if you currently have automated archiving to tape using one of the supported tape managers (VMTAPE, EPIC, DYNAM/T , and CMS).

The SUM reorg tape manager is assumed to be the same as the tape manager used for archive activity; therefore the TAPE option is only valid if you have tape archives automated without any customization to the SQMOUNT EXEC. Any changes made to the SQMOUNT EXEC must also be made to the SQMSTAPE EXEC used by SQLREORG before the TAPE=YES option can be utilized.

An additional panel will be displayed if the TAPE=YES option is selected on the primary single user mode DBSPACE Reorganization panel. This panel will allow the user to enter tape parameters that will be used when the reorganization is performed, automatically mounting and using any tapes required for the reorganization.

```
 mm/dd/yyyy                    CONTROL CENTER                        hh:mm:ss
*------------------------- SQMREORG Tape Screen -----------------------------*
|  DATABASE ==> SQLDBA                                      CTRLID: MSTRSRV1  |
|                                                            NODE:  WMAVM1    |
| Tape-manager    ==> EPIC         (EPIC, VMTAPE, CMS, Other)                 |
| Tape-user-id    ==> EPIC         (ID of the Tape Server Machine)            |
| Tape-code-disk  ==> 19E          (Virtual address (CUU) of tape code disk)  |
| Tape-code-owner ==> $MAINT       (ID that owns the tape code disk)          |
| Tape-density    ==> 38K          (4K, 38K)                                  |
| Tape-blksize    ==> 28672        (28672)                                    |
| Tape-premount   ==> N            (Y or N, premount tape with TVI)           |
| Tape-retention  ==> 30           (Tape retention (days))                    |
|                                                                             |
| ===== LIST WORK TAPES TO BE USED DURING REORG (BLANK for SCRATCH)  =======  |
|    ______  ______  ______  ______  ______  ______  ______  ______           |
|                                                                             |
| ===== OPTIONAL INFORMATION: CATALOG DATASET NAME / TAPE SCRATCH POOL =====  |
| Catl-dataset-name  ==> ____________________________________________________ |
| Tape-scratch_pool  ==> ____________________________________________________ |
|                                                                             |
*-----------------------------------------------------------------SQMREOD5-----*
  PF1 HELP   PF3 PREVIOUS SCREEN                        ENTER CONTINUE
```

Figure 187. Single User Mode DBSPACE Reorganization SQMREORG Tape Screen

The menu above contains information SQLREORG uses to control the use of tape for the unload/reload data portion of the DBSPACE reorganization.

The parameters displayed are the current settings of either the last values the program used, or the values from various Control Center control files. Note, these values are modifiable and will be saved in the user's LASTING GLOBALV file.

The list of work tapes to be used is optional. Complete it with the number of standard label tapes you think your job will need to hold the DBSPACE to be unloaded. All filedef, labeldef, mounting, rewinding, detaching, and reuse involved will be managed for you. If this parameter is left blank, the default will be one scratch tape.

For multi-volume processing, only standard label tapes can be used. If the scratch option is specified, the unloaded DBSPACE must fit on 1 tape, since multi-scratch tape handling is not supported.

The result of processing this screen is a file with the name dbmachine DBREOSUF which is sent to the Control Center service machine. It is later requested by the SUM Reorganization execs and read to obtain the information needed to support tape processing. Figure 188 is an example of the file created from the sample Tape Screen shown above:

```
*TAPE: TAPE_MGR VMTAPE
*TAPE: TAPE_UID VMTAPE
*TAPE: TAPE_OWNER NERTOOLS
*TAPE: TAPE_DISK 19E
*TAPE: DEN 38K
*TAPE: PREMOUNT N
*TAPE: BLKSIZE 28672
*TAPE: LABEL SL
*TAPE: RETPD 30
```

Figure 188. Example dbmachine DBREOSUF Tape File

Special Considerations

Unloading to Tape

All tapes used must be standard label due to SQLDBSU's handling of non-labelled tapes (it forces a label to be written over the tape). Therefore non-labelled tapes are not supported.

A special mount/detach exec is included. It is called SQMSTAPE EXEC. The same file is also supplied as SQMSTAPE $EXEC. The DBA should not have to modify this $EXEC file. It is there simply to remain consistent with our modifiable code supplied. The SQMSTAPE EXEC is supplied to ensure that the exec resides on the code disk and is accessed by the database machine during SUM reorgs. The DBA should try the supplied exec. It is possible that modifications might have to be made depending on one's tape management system, but the exec as supplied should work for EPIC, and CMS. Depending on the type of tape drive, if VMTAPE, DYNAM/T , or another tape management product is used, the SQMOUNT EXEC may need to be modified. See “Post-Installation” on page 47.

For tape processing, the options below apply:

1. Scratch tape - When specifying scratch tape, the job will be limited to the use of one scratch tape for unload/reload.
2. Predefined volumes - When specifying volume ids on the tape option screen, SQLREORG will use the volume(s) for unloading the data. The DBA must be sure to supply enough volids to hold the DBSPACE to be reorganized.

Files Created by the SQLREORG Exec

The files created by the SQLREORG exec have the dispositions described below.

1. When using temp disk to reorganize, only the unloaded DBSPACE file (dbspacename DBSPACE Mode4) is written to the temp disk. All other files are written to the database machine's A-disk. For multiple invocations of SQLREORG, these files can add up. Therefore, the DBA must assure that there is enough space on the A-disk to hold these files. (See more information about this same topic in Chapter 34, “Automated DBSPACE Maintenance Tools” on page 427.) These files are NOT erased from the database's A disk at the end of a reorganization.
2. When using a permanent disk, all files generated by the SQLREORG exec are kept on the permanent disk. None are erased. The DBA must be certain that enough space exists to hold all files generated during SQLREORG processing. (See more information about this same topic in Chapter 34, “Automated DBSPACE Maintenance Tools” on page 427.)
3. When using tape, the same considerations apply to tape as to temporary disk usage. (See above).

Viewing Job Output (Scheduled Single User Mode SQLREORG)

The output of a single user mode SQLREORG job is available on the database's A-disk. As a result, the output is not available from the normal VJ (View Job) option of the panels. You must link and access the database machine's 191 disk and/or view the log of the Control Center machine and the file dbmachine SQMREORG for results.

Running Single User Mode DBSPACE Reorganization in Command Mode

The Single User Mode DBSPACE Reorganization Tool is best used as a scheduled job from the Control Center panels. The utility can be run, however, from the command mode. The HELP file guides you as to the parameters needed when invoking the utility. Just enter HELP REORGF from the CMS command line to view the help available.

An example of invoking single user mode SQLREORG from command mode, with parameters for unloading to a permanent disk and logmode=L, follows:

Command Mode SUM Reorg

```
SQM dbname REORGF dbname dbspaceowner.dbspacename ( DISK=C
LOG=L
```

If you wish to use a tape and execute the tool in command mode, you must have created the dbmachine DBREOSUF file on the Control Center service machine prior to invoking the tool. This can be accomplished by using the panels only through the job schedule step (then cancelling the scheduled job) or by simply creating a file and sending it to the service machine.

Failure Analysis

The failures from SQLREORG can be analyzed using the information provided in “Failure Analysis” on page 380. The new functions of SQLREORG (tape control temp-disk handling) can provide additional error detection (missed tape mounts, more than one scratch tape). These errors will be noted in the SQLREORG console. Use this console to further analyze the problem.

Chapter 31. DBSPACE Reorganization Driver Tool

Overview

The fundamental purpose of the DBSPACE Reorganization Driver tool is to provide a capability to schedule a single job that will process multiple DBSPACE reorganizations. A variety of selection options allows the DBA to choose subsets of DBSPACES to be reorganized. For example, VMDSS users can select DBSPACES by storage pool to take advantage of Data Spaces Support and striping.

This tool executes in multiple user mode as a database application and should be scheduled to run on a Control Center support machine. The tool consists of 2 parts: a panel driven candidate selection process and job scheduling function; and the actual execution of the DBSPACE reorganizations under the control of the DBSPACE Reorganization Driver program (SQLREODR). The SQLREODR program normally executes on a support machine, and requires that the support machine have DBA authority within the target database.

The SQLREODR Program is primarily intended to be run as a scheduled job rather than from command mode. It can be invoked in panel mode by either entering SQLREODR at the CMS READY prompt, or by using **U** (Database Utilities) on the Control Center Main Menu, and then entering Option **RD**. Command mode execution of SQLREODR will be described later.

Review the *DB2 Server for VM Database Administration* manual for an understanding of the DBSPACE reorganization process.

How the DBSPACE Reorganization Driver Tool Works

Processing Flow

The DBSPACE Reorganization Driver Tool (SQLREODR) consists of a two-step process. The first step is the creation of an input file of DBSPACE names to be reorganized and the setting of the parameters which will drive the second step, namely the execution of SQLREODR (the actual DBSPACE reorganizations). The first step can be done manually as explained in “Running SQLREODR in Command Mode” on page 403. However, the simpler way of preparing the file and parameters is by using the panel which is invoked either through the RD function of the Utilities panel, or by invoking SQLREODR without any parameters (which will cause a panel to be displayed). The first step results in a list of DBSPACEs to be reorganized. The list is created in a file with a filename of the database name and a filetype of REODRIPK. The file is sent to the Control Center service machine which controls the support machine where the job will run.

The second step of the driver function is invoking the SQLREODR exec. When the SQLREODR job is run on the support machine, it asks for the list file from the Control Center service machine. The status flags are all set to a blank. As the driver job is run, the status flag is set to R for those DBSPACEs which were reorganized. If all DBSPACEs were reorganized, then when the job completes, all flags are set back to blanks. The file is then returned to the service machine for

reuse in the event the job is rescheduled. Note that if all DBSPACEs were not reorganized in one run of the job, the remaining ones (those with a blank status flag) would be chosen when the job was run again (those whose flag contains an R are bypassed). When all DBSPACEs are set to R, the list is reset to blanks in preparation for another cycle. Prior to execution or re-execution, the DBA can modify the list in any fashion by manipulating the status flag, modifying the order of the DBSPACES, adding an entry, deleting entries, as well as changing the values of the columns PCTFREE, ALTFREE, NPAGES, and PCTINDX (used by SQLREORG in the ACQUIRE DBSPACE command).

During the driver run, the remaining time left in the job is checked at several points. Before a new reorganization starts, time left is checked to determine whether time remains for it to be done. The last elapsed time from the SQLMAINT table is used to determine this. If enough time does not remain, the next DBSPACE is tried. During the reorganization, after the DBSPACE unload step (just prior to starting the actual DBSU to drop and reload), the time left is checked again. If there is insufficient time to complete the remaining reorg, the DBSPACE is NOT reorganized.

If a log archive is requested upon job completion, the log archive will be initiated by SQLREODR only if at least one DBSPACE was successfully reorganized.

If a reorganization should encounter an error, the run will stop if any of these conditions are met:

1. Stop on Failure was specified.
2. Tape is being used for unload device.
3. Permanent disk is being used for unload device.

Note that if you wish to run in command mode, the input file of DBSPACE names must be set up properly before invoking SQLREODR (“Processing Flow” on page 391). For more details, refer to “Running SQLREODR in Command Mode” on page 403.

A report is produced with each run of SQLREODR, as shown in Figure 189 on page 393. The report contains information about the options chosen, start/stop times of various activities, and other informative messages.

```
                        SQLREORG DRIVER REPORT
                      Database: WMAVM1.CPMICDBA     04/09/97 13:34:16

13:34:16 This run of SQLREODR had the following options chosen:

      TIME=20 SOF=Y TAPE LPOOL LMODE=L LTHRESH=69 TACT=ARC LBA=N/Y

TIME STAMP    ACTIVITY
************************************************************

13:34:16      SQLREODR started

13:34:24      REORG of M262264.M262264A started
13:39:02      REORG of M262264.M262264A completed Elapsed Time =  4.63

13:39:02      REORG of M356959.M356959A started
13:39:46      REORG of M356959.M356959A completed Elapsed Time =  0.73

13:39:47      REORG of M661995.M661995A started
13:42:08      REORG of M661995.M661995A completed Elapsed Time =  2.35

13:42:08      REORG of M668047.M668047A started
13:46:05      REORG of M668047.M668047A completed Elapsed Time =  3.97

13:46:05      REORG of M760595.M760595A started
13:47:01      REORG of M760595.M760595A completed Elapsed Time =  0.92

13:47:01      LOG ARCHIVE after reorgs starting
13:50:25      LOG ARCHIVE completed Elapsed Time =  16.15

13:50:35      Job complete. SQLREODR terminating.


                Number of DBSPACEs successfully reorged =  5

             Total elapsed time for entire run = 16.15 minutes


                 End of Report for  CPMICDBA  SQLREODR JOB
```

Figure 189. SQLREORG DRIVER REPORT

Related files

The file created in the first step of the DBSPACE Reorganization Driver tool process contains entries for each DBSPACE to be reorganized. Each record is a fixed length 128-byte entry consisting of the following fields:

```
Columns    Length     Description
001 - 001       1      Status flag indicating whether the DBSPACE
                       has been reorganized (R) or not (blank or ' ').
003 - 010       8      DBSPACE owner
012 - 029      18      DBSPACE name
031 - 034       4      Pool
036 - 039       4      DBSPACE number
041 - 048       8      Size of DBSPACE (NPAGES)
050 - 057       8      Number of active pages (NACTIVE)
059 - 061       3      Percent Index (PCTINDEX)
063 - 065       3      Current PCTFREE of DBSPACE
067 - 069       3      PCTFREE to be used in ACQUIRE DBSPACE
071 - 073       3      ALTFREE to be used after loading the DBSPACE
075 - 081       7      Last REORG date from SQLMAINT table  (or blank)
083 - 090       8      Last REORG time from SQLMAINT table (or blank)
092 - 095       4      Last elapsed time of REORG from SQLMAINT table
097 - 103       7      Last Update statistics date from SQLMAINT table
105 - 112       8      Last Update statistics time from SQLMAINT table
114 - 114       1      Sort control character (sort by pool number, '-'
                       for descending order, '>' for ascending)
116 - 119       4      Absolute pool number (used for sorting)
121 - 121       1      Sort control character (as above, but for sort
                       by number of active pages)
123 - 130       8      Absolute active pages (used for sorting)
132 - 141      10      COMMITCOUNT value
```

Figure 190. REODRIPK List File layout

This file contains one record for each DBSPACE selected. If the unload is to be done to tape, the information needed to control the tape processes is added to the end of the file. See “Unloading to Tape” on page 401 for a more detailed discussion of tape use.

Files created during the reorganization will have a filename that is similar to the DBSPACE name that is being reorganized. If the dbspace name has any special characters, they will be removed. If the DBSPACE name is longer than eight characters, then it will be truncated to eight. The basic files created are:

Filename	Description
dbspacename UNLOAD	The UNLOAD SQLDBSU command - used as input to the SQLDBSU (unload DBSU command file).
dbspacename UNLIST	The output listing of the UNLOAD DBSPACE portion of the reorganization (UNLOAD DBSU execution listing).
dbspacename DBSPACE	The DBSPACE in unloaded format - an output of the DBSU. It would be needed in the event of recovery from a failed reload step of the reorganization. It can also be used as a point-in-time backup copy of a DBSPACE.
dbspacename SQLDBSU	The SQLREORG-generated DBSU input file containing all the commands needed to reorganize the DBSPACE.
dbspacename LISTING	The output listing of the RELOAD portion of the reorganization (DBSU execution listing).
dbspacename PROGDBSU	The input to the DBSU to unload the access modules.

dbspacename PROGLIST	The output listing of the unload access modules step.
dbspacename PROG*nnnn*	The unloaded access modules (packages).
dbspacename EXEC	Exec used to recover from an interrupted reorganization.

The disposition of these files depends on whether the reorganization is successful and on whether a temporary disk is used or a permanent disk is available during the reorganization. In all cases, the files needed for recovery from a failed reorganization (the EXEC, DBSPACE, SQLDBSU, and those pertaining to packages), if present, will always be available on either the Support machine's A-disk or on the permanent disk specified, and/or in the reader of the user id running the reorganization. The other files will be on the A-disk or on the permanent disk, if used.

Who Can Use the DBSPACE Reorganization Driver Tool

Use of the DBSPACE Reorganization Driver tool requires Control Center Database administration-level or greater authorization.

Features

The SQLREODR tool has these features:

- Gives the DBA a variety of selection criteria for DBSPACE candidate choices by storage pool, size, and type
- Provides a list of DBSPACE candidates and allows the DBA to issue commands (update statistics or show DBSPACE) to display the output using different sort options, to change DBSPACE attributes, and to exclude specific DBSPACES from reorganization using the OMIT selection feature
- Provides for an optional log archive to be taken before and/or after the reorganizations are done
- Allows for optional archives to be taken during the job run when a specified log threshold is reached, either log archives (if logmode = L) or database archives (if logmode = A)
- Includes parameters to unload to disk (temporary or permanent) or tape
- Updates the SQLMAINT table, if SQLMAINT is installed
- Executes in multiple user mode
- Allows scheduling of multiple DBSPACE reorganizations
- Provides several options for unload media: automated tape support, temporary disk management, and permanent disk option

DBSPACE Reorganization Driver Entry Panel

Shown in Figure 191 on page 396 is the entry panel of the DBSPACE Reorganization Driver tool.

```
 mm/dd/yyyy                    CONTROL CENTER                          hh:mm:ss
*------------------------ SQLREORG Driver Utility ---------------------------*
|                                                          CTRLID: SQLMOD1    |
|  Database ==> SQLDBA                                     NODE:   VMSYSTM1   |
|                                                                             |
|  DBSPACE OWNER ==> ________  (blank for ALL, PUBLIC, PRIVATE, ownername)    |
|                                                                             |
|  DBSPACE NAME  ==> ___________________  (blank for ALL, use % for wildcard) |
|                                                                             |
|  STORPOOL      ==> ____             (blank for ALL Pools or specific        |
|                                     including minus sign for non-logging pool)
|  REPREP PKGS   ==> YES  (YES/NO dflt)  ALLSTATS    ==> NO   (YES/NO dflt)   |
|                                                                             |
|  DBSPACE PAGES:  MINIMUM ==> 1         COMMITCOUNT ==> 0                    |
|                  MAXIMUM ==> 9999999                                        |
|  DBSPACE NACTIVE PAGES   ==> P         (A=All, P=positive, N=negative       |
|                                         values 1 - 9999999 )                |
|       The % SQL wildcard character can be used when specifying the          |
|       Dbspace Owner or Dbspace Name values for selection.                   |
|                                                                             |
*-----------------------------------------------------------------SQMREODR------*
    PF1 HELP    PF3 QUIT    PF4 EXIT    PF5 Main Menu    Enter Retrieve List
```

Figure 191. SQLREODR Screen

From the menu displayed, the DBA can select the criteria to be used to obtain a list of candidate DBSPACES for reorganization. DBSPACES can be selected by storage pool, by owner (ALL, PUBLIC, PRIVATE, or an owner can be specified which may include the database wildcard character), by DBSPACE name (which may include the wildcard character), or by any combination of these. In addition, the choice of DBSPACES can be further limited by specifying a number or range of DBSPACE pages or active pages.

The DBA can also specify three additional options on this menu; whether to rebind packages, update all statistics, and a COMMITCOUNT value (valid in DB2 Server for VM Version 5.1 and above). These options apply to all selected DBSPACEs when the reorganization portion of the job is executed.

DBSPACE Reorganization Driver List Panel

The display panel below shows the DBSPACE candidates meeting the search criteria (in this example, DBSPACE OWNER = PUBLIC; DBSPACE NAME = %PMP%). From this screen, you have the ability to execute commands, for example, to update statistics, to issue show DBSPACE commands, or to exclude DBSPACES from reorganization, by using the OMIT selection option.

Certain DBSPACE attributes can be changed in the panel's modifiable fields. Using the provided fields, the amount of free space, the percentage of space reserved for indexes, and the size of the DBSPACE can be altered. By changing the value in the POOL field, DBSPACES can be migrated from one storage pool to another.

Sort function - The DBSPACES will be reorganized in the order they appear in this list. The list can be sorted by name, number of active pages, number of pages (descending), or storage pool, through the use of the appropriate PF key.

Shown in Figure 192 on page 397 is an example of the Selection List panel:

```
 mm/dd/yyyy                      CONTROL CENTER                        hh:mm:ss
*------------------------ SQLREORG Driver Utility --------------------------*
|  Database => SQLDBA                                                       |
|                                  PCTFREE  PCT                             |
|  SEL OWNER     NAME              ACQ ALT  INDX  POOL  NPAGES   NACTIVE    |
|  --- -------- ------------------ ------- ----  ----  -------  -------     |
|   _  PUBLIC   PMPACM_DBS_T        10   0    20   -21     4096      334     |
|   _  PUBLIC   PMPACM_DEV          10   0    20    15     4096      270     |
|   _  PUBLIC   PMPACM_TRN_DBS_T    10   0    10    18     2560     1123     |
|   _  PUBLIC   PMPACP_DBS_T        10   0    20    -8     1024        1     |
|   _  PUBLIC   PMPACP_DEV          10   0    20   -12     1024        1     |
|   _  PUBLIC   PMPACP_TRN_DBS_T    10   0    10   -20     2048       40     |
|   _  PUBLIC   PMPBPM_DBS_T        10   0    20   -11     2048      119     |
|   _  PUBLIC   PMPBPM_DEV          10   0    20   -12     2048       98     |
|   _  PUBLIC   PMPBPM_TRN_DBS_T    10   0    25    18     2048      384     |
|   _  PUBLIC   PMPBPP_DBS_T        10   0    20    -9     1024        1     |
|                                                                           |
|            The fields in red are modifiable for SQLREORG use              |
|   Select:  O = Omit DBSPACE,    S = Show Dbspace,    U = Update Statistics|
|                                                                           |
|                               Page 1     of 3                             |
*-------------------------------------------------------------SQMREOD2-----*
PF:  1 Help         3 QUIT        4 StatHistory    7 Bkwd          8 Fwd
    9 Sort/Name   10 Sort/Nact   11 Sort/Pages   12 Sort/Pool  ENTER CONTINUE
```

Figure 192. SQLREODR Selection List Screen

A description of the fields on the selection list panel follows:

Column Heading	Description
PCTFREE ACQ	Percentage of space on each page to be reserved, at the time the database is reorganized, for later data inserts. The default value is 10%. This value is used in the ACQUIRE DBSPACE command by SQLREORG. It is the value you requested at reorg time, **NOT** the current value of free space and **NOT** the current value of reserved space (for this value, see the PCTFREE ALT value below).
PCTFREE ALT	Current percentage of reserved free space. This amount should be lower than the PCTREE ACQ value to allow future inserts to use the reserved space on the existing DBSPACE pages, thus allowing clustering properties to be maintained. The free space value is altered by SQLREORG after all tables are reloaded. If the actual free space falls below this value, the DBSPACE will be split, reducing retrieval efficiency.
PCT INDX	Current amount of pages in the DBSPACE reserved for index use. To alter, change the value.
NPAGES	The current size of the DBSPACE. To change the size of the dbspace specify the new value.
POOL	The storage pool currently associated with the DBSPACE. Can be changed to direct the reorganized DBSPACE into another pool.

NACTIVE	The number of currently active data pages in the DBSPACE. This is displayed for informational purposes only.

The select options may be invoked immediately against the listed DBSPACES.

Select Option	Description
Omit DBSPACE (O)	Excludes DBSPACES from the final reorganization selection file.
Show DBSPACE (S)	Executes the database operator command SHOW DBSPACE.
Update Statistics (US)	Executes the database DBSU statement UPDATE STATISTICS for chosen DBSPACE and updates the SQLMAINT table if available.

The PF Keys offer these options:

PF Key	Description
Stat History (PF4)	If the SQLMAINT table has been installed, the last maintenance dates and times for the DBSPACES will be displayed.
Sort/Name (PF9)	Sorts the list by DBSPACE name (default).
Sort/Nact (PF10)	Sorts list by number of active DBSPACE pages.
Sort/Pages (PF11)	Sorts list by DBSPACE size (number of pages) in descending order.
Sort/Pool (PF12)	Sorts list by storage pool number.

DBSPACE Reorganization Driver Job Control Panel

After selection has been made for reorganization, the DBA is shown a menu which specifies how SQLREODR is to control the reorganization step of the job. It contains required parameters and, if any DBSPACES chosen are in logging pools (recoverable storage pools), a logging pool section that includes fields for setting thresholds should archive activity be necessary before or after reorganization. To assist the DBA in archive management, relevant database logging parameters are highlighted in the lower portion of the panel.

Shown in Figure 193 on page 399 is the job control panel of the DBSPACE Reorganization Driver tool.

```
 mm/dd/yyyy                      CONTROL CENTER                          hh:mm:ss
*------------------------- SQLREORG Driver Utility --------------------------*
|                                                          CTRLID: SQLMOD1   |
|  DATABASE ==> SQLDBA                                      NODE:  VMSYSTM1  |
|  DBSPACE OWNER ==> PUBLIC           DBSPACE NAME  ==> %PMP%                |
|                                                                            |
|  NOTE:   THERE ARE DBSPACES IN LOGGING POOLS CHOSEN FOR THIS RUN. *****    |
|                                                                            |
| ========================= 'REQUIRED PARAMETERS' ========================== |
| TIME LIMIT  ==> 120   (Minutes; dflt 120)  STOP ON FAILURE ==> Y  (Y or N)  |
|                 ----                                        -              |
| FILEMODE  ====> _ (Linked R/W disk; dflt TEMP DISK)   TAPE ==> Y  (Y or N)  |
|                 -                                           -              |
| ============== FOLLOWING OPTIONS ARE FOR LOGGING POOLS ONLY: ============= |
|                                                                            |
| LOG THRESHOLD ==> 70  LOG ARCHIVE BEFORE/AFTER JOB: ==> N/N  (logmode L)   |
| ACTION WHEN THRESHOLD REACHED: ==> ARCH  (ARCH or STOP)                    |
|                                                                            |
| ==FYI: CURRENT DB SETTINGS: LOGMODE ==> L SLOGCUSH ==> 90 ARCHPCT ==> 79   |
|                                                                            |
*----------------------------------------------------------------SQMREOD4-----*
PF1 HELP   PF3 QUIT     PF5 REVIEW/MODIFY LIST         ENTER CONTINUE
```

Figure 193. SQLREODR Job Control Screen

If there are DBSPACEs chosen which are in a logging pool, this will be indicated on the panel.

Required Parameters

The required job control parameters used with the DBSPACE Reorganization Driver tool are:

Parameter	Description
TIME LIMIT	Limits the duration of the reorganization job. After this time limit has passed, the current reorganization will be allowed to complete, but no new one will begin. The default is 120 minutes (2 hours).
STOP ON FAILURE	Enter **Y** if the SQLREORG Driver **Y** should stop if any DBSPACE reorganization fails; enter **N** if the reorganizations should continue.
FILEMODE	Allows you to specify a linked minidisk that is to be used for the unloaded data. This minidisk must be linked in R/W mode prior to running SQLREODR. If this option is not specified, SQLREORG will define a temporary disk large enough to hold all unloaded data. If temporary disk space is used for unloading, it will be reused for subsequent unloading and reloading, if large enough to handle the next DBSPACE, thereby saving the overhead and time needed to acquire and format new temporary disk space.

TAPE	Allows you to unload the DBSPACES to tape rather than to disk. If Y is entered, more parameters will be needed. An additional panel will be presented. (See menu and descriptions below).

Logging Pool Parameters

These parameters are present only if any of the DBSPACES selected are in logging pools:

Parameter	Description
LOG THRESHOLD	Percent of log pages filled at which point some action is to be taken.
ACTION WHEN THRESHOLD REACHED	Stop job, or initiate archive (log archive, if logmode = L; database archive, if logmode = A) then resume reorganizations if time remains.
LOG ARCHIVE BEFORE/AFTER JOB	Take a log archive before starting reorganizations and/or after the reorganizations are completed; applies only for logmode = L.

DBSPACE Reorganization Driver Tape Control Panel

If the DBA specifies that the reorganization job is to unload to tape, a Tape Control panel is presented. The DBSPACE Reorganization Driver tool will issue the required filedef and labeldef commands for one or more volumes. Multiple volume standard label tapes, single volume scratch tapes, catalogued tapes and scratch pool processing will be managed.

Shown in Figure 194 is an example of the Tape Control panel:

```
 mm/dd/yyyy                    CONTROL CENTER                        hh:mm:ss
*------------------------- SQLREORG Driver Utility --------------------------*
|  DATABASE ==> SQLDBA                                        CTRLID: MSTRSRV  |
|                                                              NODE:  WMAVM1   |
| Tape-manager    ==> EPIC         (EPIC, VMTAPE, CMS, Other)                  |
| Tape-user-id    ==> EPIC         (ID of the Tape Server Machine)             |
| Tape-code-disk  ==> 195          (Virtual address (CUU) of tape code disk)   |
| Tape-code-owner ==> EPIC         (ID that owns the tape code disk)           |
| Tape-density    ==> 38K          (4K, 38K)                                   |
| Tape-blksize    ==> 28672        (28672)                                     |
| Tape-premount   ==> N            (Y or N, premount tape with TVI)            |
| Tape-retention  ==> 30           (Tape retention (days))                     |
|                                                                              |
| ===== LIST WORK TAPES TO BE USED DURING REORGS (BLANK for SCRATCH) =======   |
|    ______  ______  ______  ______  ______  ______  ______  ______            |
|                                                                              |
| ===== OPTIONAL INFORMATION: CATALOG DATASET NAME / TAPE SCRATCH POOL =====   |
| Catl-dataset-name  ==> ____________________________________________________  |
| Tape-scratch_pool  ==> ____________________________________________________  |
|                                                                              |
*-------------------------------------------------------------------SQMREOD5-----*
  PF1 HELP   PF3 QUIT     PF7 PREVIOUS SCREEN            ENTER CONTINUE
```

Figure 194. SQREODR Tape Control Panel

The Tape Control panel is used to contain information the DBSPACE Reorganization Driver tool uses to control the use of tape for the unload/reload data portion of the DBSPACE reorganization.

The parameters displayed are the current settings of either the last values the program used, or the values from various Control Center control files. Note, these values are modifiable and will be saved in the user's LASTING GLOBALV file.

The list of work tapes to be used is optional. Complete it with the number of standard label tapes you think your job will need to hold the largest DBSPACE to be unloaded. All filedef, labeldef, mounting, rewinding, detaching, and reuse involved will be managed for you. If this parameter is left blank, the default will be one scratch tape.

NOTE: For multiple volume processing, only standard label tapes can be used. If the scratch option is specified, the largest DBSPACE unloaded must fit on 1 tape, since multiple scratch tape handling is not supported.

Uses of SQLREODR

SQLREODR may be used to do a complete database reorganization of one or more DBSPACEs for purposes of maintenance and performance improvement.

Another purpose for this tool is to allow VMDSS users to take advantage of Data Spaces Support and striping by providing reorganization at the storage pool level.

Special Considerations

Unloading to Tape

The following is a description of the services, options, and restrictions which the DBSPACE Reorganization Driver tool provides when the unload to tape option is chosen. A tape use option panel is presented when the Tape option is specified. The tape parameters are written to the REODRIPK file following the DBSPACE entries. The use of tape is for unloading/reloading only and not for storage of unloaded DBSPACES. A tape (or tapes) will be reused by SQLREORG during the course of the job. The DBA must know how many tapes will be needed to hold the data from the largest DBSPACE unloaded. After each reorganization, the tape (or tapes) will be rewound and reused for the next reorganization.

First, all aspects of tape processing are automated. SQLREORG will take care of the FILEDEF and LABELDEF needed for the DBSU unload/reload steps. In addition, all mounting and rewinding between unloads/reloads and between reorganizations will be handled automatically.

Second, all tapes used must be standard label due to SQLDBSU's handling of labeled tapes (it forces a label to be written over the tape). SQLREODR does not support non-labeled tapes.

Third, a special mount/detach exec is supplied with SQLREODR. It is called SQMSTAPE EXEC. The same file is also supplied as SQMSTAPE $EXEC. The DBA should not have to modify this $EXEC file. It is there simply to remain consistent with our sample, reference code supplied. The SQMSTAPE EXEC is supplied to ensure that the exec resides on the code disk and is accessed by the support machine during a SQLREODR run. The DBA should try the supplied exec.

It is possible that modifications might have to be made depending on one's tape management system, but the exec as supplied will work for VMTAPE, CA-DYNAM/T , EPIC**, and CMS.

These options are available and an explanation of how they work is provided:

1. Scratch tape - when specifying scratch tape, the job will be limited to the use of one scratch tape for unload/reload. Should the DBSPACE spill over to a second (or more) volumes, SQLDBSU will ask for another scratch tape to complete the unload step. Prior to starting the reload portion of the DBSPACE reorganization, SQLREORG checks to determine whether more than one scratch tape was used during the unload. If so, the reorganization will stop, and be flagged as incomplete for that DBSPACE. The job will continue on to the next DBSPACE. This is a limitation of the driver tool since the second volume id is not identified in the LABELDEF, and a subsequent reload would fail. So, when choosing scratch tape the DBA must be certain that the largest DBSPACE to be reorganized will fit on one volume when unloaded.
2. Predefined volumes - when specifying volume ids on the tape option screen, SQLREORG will use the volume(s) for unloading the data. If only one volume is specified, SQLREORG will use that same volume over again for each DBSPACE reorganized. The utility will attend to all rewinds and only one mount activity will occur, as long as the unload fits on the one tape. Should it spill over, SQLDBSU will ask for a scratch to be mounted. The same logic will then apply as to multiple scratch tapes: if the unload uses more tapes than were supplied in the REODRIPK file, the reorganization for that DBSPACE will stop after the unload step. The tapes will be demounted, then the first tape will be mounted again for use in the next DBSPACE to be reorganized. The DBA must be sure to supply enough volids to hold the largest DBSPACE to be reorganized.

Unloading to TEMP Disk

When using TEMP disk to hold the unloaded data, these considerations apply.

The use of the TEMP disk by SQLREORG is changed somewhat when SQLREORG is invoked by SQLREODR. Temp disk will be handled thus: SQLREORG determines whether a TEMP disk is currently attached (the label used by SQLREORG when defining and formatting a TEMP disk is SQLTMP). If SQLREORG does not have a TEMP disk attached (which would normally be the case upon the first invocation of SQLREORG by SQLREODR), it will determine the size required and acquire one, as usual. When the reorganization is completed, SQLREORG will NOT release the TEMP disk if it has been invoked by SQLREODR (in anticipation of more invocations). If the reorganization is successful, we will erase the DBSPACE file on that TEMP disk before returning to SQLREODR. When invoked again, it checks whether there is a TEMP disk available (yes); it then checks whether it will be large enough to hold the current DBSPACE to be reorganized. If so, it uses the current TEMP disk. If not, it will release the old TEMP disk and acquire a new one. If a reorganization fails, the temp disk will be relabeled to SQLBAD. This way the temp disk will not be reused for other reorganizations and the unloaded dbspace is saved. When the last DBSPACE has been reorganized, SQLREODR will release and detach the final TEMP disk. Therefore, the DBA can see that it would be more efficient to order the reorganizations in descending order of DBSPACE size since only one TEMP disk would be acquired by SQLREORG for the entire job. This option is available in the selection screen by using the sort on pages option.

Files Created by SQLREORG

The files created by the SQLREORG exec have the following dispositions:

1. Use of TEMP disk. When using TEMP disk to reorg, only the unloaded DBSPACE file (dbspacename DBSPACE Mode4) is written to the TEMP disk. All other files associated with the reorganization are written to the support machine's A-disk. For multiple invocations of SQLREORG, these files can add up. Therefore, the DBA must assure that there is enough space on the A-disk to hold them. (See more information about this same topic in Chapter 34, “Automated DBSPACE Maintenance Tools” on page 427.)
2. Use of a permanent disk. All files generated by the SQLREORG program are kept on the permanent disk. None are erased. The DBA must be certain that enough space exists to hold all files during a run of SQLREODR which entails multiple reorganizations.
3. Use of tape. The same considerations apply to tape as to temporary disk use.

Running SQLREODR in Command Mode

SQLREODR is best used as a scheduled job from the Control Center panels. The utility can be run, however, from the command mode. The HELP file guides you as to the parameters needed when invoking the utility. Just enter SQLREODR HELP to view the help available.

The most important item to remember is that the dbname REODRIPK file must be established on the Control Center service machine's A-disk. Previous sections explain the layout of this file. The file can be generated from the panels (by going all the way through to the schedule step). You can also copy and edit the file and send it to the Control Center service machine.

Note these points concerning the support machine on which the tool runs:

1. The user ID must have DBA authority on the database.
2. The user ID must initialize to the proper CTRLCTR service machine id. This is done by issuing the command SQM (ctrlctrid).
3. The user ID must have proper authorization to the Control Center service machine.
4. If using tape for unload, the console must be started by SQLREODR to automate the tape functions. SQLREODR will always spool the console to the reader, regardless of where it was spooled at the start of the job. It will not be re-directed at the end of the job (this applies only to command mode processing). Jobs initiated by Control Center on a support machine always spool the console to the reader for the duration of the job.

If using tape for the unload, the proper tape parameters MUST be present in the REODRIPK file. The best way to determine these is to run a sample retrieval and invoke the tapes panel, schedule the job with I (to mark it inactive), then look at the tape parameters generated on the REODRIPK file sent to the Control Center service machine.

A sample of the tape parms appended to a REODRIPK file is:

```
*TAPE: TAPE_MGR VMTAPE
*TAPE: TAPE_UID VMTAPE
*TAPE: TAPE_OWNER $MAINT
*TAPE: TAPE_DISK 19E
*TAPE: DEN 38K
*TAPE: PREMOUNT Y
*TAPE: BLKSIZE 28676
*TAPE: RETPD 30
*TAPE: DSN REODRIVR.TEST1
*TAPE: VOLID QU2221
*TAPE: VOLID QU2222
```

Figure 195. SQREODR Tape Parameters in List File

These parms start in column one. The VOLID parm is present for EACH tape volume used when using pre-defined volumes. When using scratch tape, the VOLID parameter is not used.

Finally, a sample of the invocation of the Driver utility from command mode, with parameters, follows:

```
SQLREODR dbname (TIME=60 SOF=Y LPOOL NORE LMODE=L TAPE
                 LTHRESH=70 TACT=STOP LBA=Y/N
```

Figure 196. SQREODR Command Mode Example

A brief explanation is:

1. The run duration is 60 minutes
2. The Stop on Failure option is YES
3. There are DBSPACES in logging pools
4. Packages will not be re-prepped
5. Log mode for the database is L
6. Tape will be used for unloading by SQLREORG
7. The log threshold is 70 at which time, if reached,
8. The utility will stop processing
9. There will be a log archive taken before the reorgs start but not after the reorganizations are completed.

Failure Analysis

The failures from SQLREORG can be analyzed using the information provided in the section on that utility. The new functions of SQLREORG (tape control, temp-disk handling) may provide additional error detection (missed tape mounts, more than one scratch tape). These errors will be noted in the SQLREORG console as well as referenced in the RUNDRIVR report file created by SQLREODR. Use this file to determine the nature of the problem. Use the console which is created with each job to further analyze the problem. Restart can be done after adjustments are made to the REODRIPK file if necessary.

Chapter 32. Table Reorganization Tool

Overview

The Table Reorganization and Redefinition tool of Control Center provides a means to reorganize, migrate, backup, and redefine tables within databases. This function provides you with improved control and flexibility to manage database objects for better performance and adaptability to changing needs.

The Table Reorganization and Redefinition tool can be executed in command mode (see Appendix G, “Command Mode Interface” on page 523) and is more commonly referred to by its command name, SQLTABLE.

SQLTABLE differs from the System Administration tools of Control Center because it executes as a database application, rather than as a function within the database virtual machine. SQLTABLE therefore normally executes on the user's virtual machine and only requires that you have needed authority within the target database, rather than Control Center authority.

The database authorities necessary to use SQLTABLE differ depending on which SQLTABLE function is executed. Users with INSERT and DELETE authority to a table can perform the DATAUNLOAD and DATALOAD functions on that table. The DDL capture required for the full table reorganization or redefinition function of SQLTABLE (see Figure 197 on page 409, Selection Options **2, 3,** or **4**) will need DBA authority within the target database.

Before You Begin

The following comments apply to a full table reorganization and not to several of the other options available. Refer to the *DB2 Server for VM Database Administration* manual for a thorough understanding of the database TABLE reorganization process.

SQLTABLE should be used during non-peak hours to prevent locking contention with other users of the database. Due to the intensive updating of the database system catalogs, use of this tool is prone to lock contention during heavy multiple user sessions. To avoid catalog contention, do not run more than one SQLTABLE job against different tables simultaneously within a single database.

Installations that want to schedule reorganizations and maintenance jobs should implement a separate Control Center support machine to provide the scheduling capability without impacting the responsibilities of the primary Control Center service machine. SQLTABLE jobs should not be scheduled to execute on the managing Control Center service machine, since all automated operations functions with all databases will be disabled while the maintenance functions execute. Although SQLTABLE can be executed in batch mode with any scheduling product, a separate Control Center support machine will allow these jobs to be scheduled easily from the Control Center panel interface.

Who Can Use the Table Tool

Use of this tool requires Control Center database Administration-level or greater authorization.

How the SQLTABLE Tool Works

Processing Options

The following is a summary of the various SQLTABLE processing options:

- Table Reorganization:

  If only the source database and table is specified (meaning that this is not a migration), SQLTABLE will perform a table reorganization, using DATAUNLOAD to capture the data from the table and DATALOAD to restore the data to the table. If the DATA only option is specified, SQLTABLE will delete all rows in the table prior to performing the DATALOAD. The DATA only option will require that you have SELECT and INSERT authority to the table.

  If the DDL only option is specified, SQLTABLE will capture the table create statement and all associated database objects, including indexes, views, and grants.

  When data and DDL are both captured for a reorganization, you must have DBA authority for complete capture of *all* dependent objects within the database (including those of other users). The table reorganization for DDL and data will capture the data using DATAUNLOAD, will DROP and CREATE the table, DATALOAD the data, and then recreate all dependent objects (indexes, views, and grants).

- Table Migration:

  SQLTABLE will migrate data or table DDL between databases if the NEW DATABASE parameter is specified. The source table will not be dropped or modified in any way.

- Table Copy:

  SQLTABLE will copy either data or the table definition (DDL) or both if a NEW OWNER.TABLE is specified. If DATA only is specified, the target table must already exist and the column definitions must match the source table exactly. If DDL is specified, you must have DBA authority to capture all dependent database objects. It is important to note that if the target OWNER is the same as the source table OWNER, the INDEX and VIEW create names will be identical to the existing ones on the old table, so the create statements will *fail* unless you take steps to eliminate the duplication.

- Table Move:

  SQLTABLE will move a table to a new DBSPACE if the DATA and DDL option is specified with the NEW OWNER.DBSPACE parameter. Note that the old table will *only* be deleted if the new OWNER and TABLE remain the same as the old OWNER and TABLE.

- REDEFINE Function

  SQLTABLE will allow you to REDEFINE a table, changing column names, adding columns, deleting columns, changing column datatypes and lengths. Since the DATAUNLOAD/DATALOAD functions of the Database Services Utility (DBSU) are used for this process, all restrictions described in the “Data

Conversion Summary" section of the *DB2 Server for VSE & VM Database Services Utility* manual apply. You should be aware of which datatypes are compatible and which ones will cause data truncation or dataload failures.

Processing Flow

When SQLTABLE is invoked, the following steps will be performed:

1. The target database is linked using DBINIT CONTROL file information.
2. Communication is establish with the database (SQLINIT).
3. User authority is verified.
4. Availability of the specified new table, DBSPACE, and database is verified.
5. An EXEC is created for recovery purposes.
6. All system catalog information pertaining to the specified table is gathered and corresponding SQL statements are created within the SQLDBSU file:
   a. Table create statements
   b. Table dataload statements
   c. Table comments
   d. Column comments
   e. Referential integrity constraints
   f. Unique column definitions
   g. Index create statements
   h. Table column grants
   i. Table grants
   j. View creates/grants/comments/labels
   k. Access Module unload/reload statements
7. If required, a temporary disk is defined and formatted.
8. Table data is DATAUNLOADed to the specified disk or tape.
9. If the PAUSE option was specified, execution ends at this point; otherwise, processing continues.
10. The SQLDBSU command file is executed under SQLDBSU.
11. The temporary disk, if used, is detached.
12. Any invalidated access modules are reprepped.

Files Created

Each file created by SQLTABLE will have a filename that is created from the specified target table name. If the table name has any special characters (non-alphanumeric), they will be removed. If the table name is longer than eight characters, then it will be truncated to eight. The basic files created are:

"tablename EXEC"	Used to complete an interrupted process.
"tablename SQTABLE"	The unloaded data.
"tablename LOADDBSU"	Table create and/or dataload DBSU file.
"tablename DDLDBSU"	DDL DBSU file. Indexes, views or grants.

The LOADDBSU file is processed first to drop the old table or delete data (as required), create the new table, and perform the dataload. The subsequent DDLDBSU file is then processed separately, since several of the SQLTABLE options can result in duplicate view and index names, causing the DBSU process to

fail. Manual intervention by you will then be required to either modify the failing statements in the DBSU file before rerunning it, or to issue comparable statements within ISQL or another database interface.

All grants are issued in the same chronological order in which they were originally issued, in order to retain hierarchical dependencies.

In order to grant authority to an object, SQLTABLE must first connect as the user who originally issued the grant. Therefore, SQLTABLE must gather database connect passwords for all grantors. If a grantor does not have a connect password, SQLTABLE will give the user a temporary one and will later remove it.

The database does not remove grant information from the system catalogs when a user is removed from the SYSTEM.SYSUSERAUTH table. Therefore, the possibility exists that SQLTABLE needs to connect as a nonexistent user in order to reestablish a grant. If this situation occurs, SQLTABLE will temporarily grant connect authority and will later revoke it.

Features

The SQLTABLE tool has these features:

- Saves and restores the following database objects related to a TABLE:
  - Table Definition
  - Data
  - Referential Integrity constraints
  - Unique column definitions
  - Indexes
  - Views
  - Grants
  - Table and Column Comments
  - Table and Column Labels
  - Packages (Access Modules)
- Loads data in clustering index sequence
- Updates statistics
- Repreps invalidated access modules
- Allows these options:
  - Commitcount usage
  - Perform table backup (image copy)
  - Copy table to a different database
  - Copy table to a different owner and/or name
  - Move table to a different DBSPACE
  - Reload data into a different table
  - Redefine table:
    - Add columns
    - Delete columns
    - Reorder columns
    - Change column names
    - Change column datatypes
    - Change column widths
    - Change null attributes

SQLTABLE Panels

The SQLTABLE tool panel shown in Figure 197 is reached from the Control Center Main Menu by choosing the Database Utilities (**U**) option and then choosing the SQLTABLE (**T**) option.

```
  ********************     SQLTABLE TABLE UTILITY     ******************
  SELECTION   ==> 3   ( 1=DATA only <DELETE *>, 2=DDL only,
                -     3=DATA and DDL <DROP TABLE>, 4=REDEFINE columns)
  DATABASE (DBNAME) ==> SQLDBA                      ( example:  SQLDBA )
                        --------
  OWNER.TABLE_NAME  ==>                             ( example: MY.USER_TABLE )
                        ---------------------------
  NEW DATABASE      ==>                             ( defaults to same )
                        --------
  NEW OWNER.TABLE   ==>                             ( COPY table and/or )
                        ---------------------------
  NEW OWNER.DBSPACE ==>                             ( MOVE table and/or )
                        ---------------------------
  DATALOAD PCTFREE  ==>            ALTER PCTFREE ==>     ( after dataload )
                        ---                          ---
  COMMITCOUNT ==> 1000  (1-50000)    REPREP PKGS ==> YES     ALLSTATS ==> NO
                        -----                        ---                   ---
  PAUSE       ==> NO    (YES / NO)      TAPE     ==> NO  (YES / NO)
                        ---                          ---
  FILEMODE    ==>       (R/W disk)
                        ---
PF:  1 HELP   3 QUIT   4 EXIT   5 Main Menu   6 SCHEDULE  10 PROCESS NOW
```

Figure 197. SQLTABLE Data Entry Panel

The SELECTION parameter specifies which of several functions should be performed.

Option	Description
Data only <DELETE *> (1)	Option 1 specifies that only the data within the table should be manipulated. The table definition, indexes, views, grants, and related Data Definition Language (DDL) statements will not be affected. This option therefore, only requires that you have select and/or insert authority for the target table. Other parameters within this panel will indicate what you desire to do with the table data. The PAUSE option can be used to merely perform a DATAUNLOAD to create an external copy of the data within a CMS file. Without the PAUSE option, SQLTABLE will perform a DATAUNLOAD, delete all data from the table, then will DATALOAD the data back into the table (essentially performing a table reorganization). The NEW OWNER.TABLE parameter can be used to instruct SQLTABLE to DATAUNLOAD the data from the old table and DATALOAD the data into a new table (which must exactly match the column definitions of the old table). This option will delete data from the NEW TABLE, but the old table data will remain intact.

DDL only (2)

Option 2 will capture all Data Definition Language (DDL) statements related to the specified table, but will not perform a DATAUNLOAD. When used with the PAUSE option, this provides a way to capture the DDL for backup or informational purposes. When the NEW OWNER.TABLE parameter is used, SQLTABLE will copy the table definition, creating an empty duplicate table under a different owner or new table name. For complete capture of all DDL, this option requires you to have DBA authority within the database.

DATA and DDL <DROP TABLE> (3)

Option 3 will capture both DDL and the data itself for a specified table. This option can be used to perform a table-level reorganization with a DROP TABLE being performed between the DATAUNLOAD and DATALOAD. This reorganization process is more complete than the DATA only reorganization performed by Option 1 because all indexes will be rebuilt for better efficiency. (However, the TABLE REORGANIZATION is not as complete or efficient as the DBSPACE REORGANIZATION performed by the DBSPACE Reorganization tools, which should be used as the primary reorganization process.)

REDEFINE columns (4)

Option 4 is used to redefine table columns and will require DBA authority within the database. Essentially, this function performs the DDL and data capture of Option 3, but provides you with an intermediate panel which allows the column characteristics to be changed before the table is recreated. Since the DATAUNLOAD and DATALOAD commands of the Database Services Utility (DBSU) are used for this function, column datatype changes are restricted to those that are supported within this underlying facility. The REDEFINE columns option cannot be scheduled since it requires interactive user inputs to define the desired changes. Please note, the unload option for REDEFINE **MUST** be disk (either temp or permanent).

The DATABASE and OWNER.TABLE_NAME parameters are required for all SQLTABLE options to indicate the target table for the selected function. The database specified must match either the database parameter or the nickname parameter of an entry in the DBINIT CONTROL file. The OWNER.TABLE_NAME parameter must identify a valid table within the target database.

All additional parameters on the SQLTABLE panel are optionally used to specify additional characteristics of the SQLTABLE function to be performed.

Parameter	Description
NEW DATABASE	Use this option to specify that the table data and/or definition should be copied to a different database. This option allows a table to be migrated from one database to another, such as from development to production. Due to differences within the two databases, such as userids within the SYSTEM.SYSUSERAUTH table, certain DDL statements cannot execute successfully unless the DBA first assures that the two environments are compatible. When the NEW DATABASE parameter is used, the table within the old database will remain unaltered.
NEW OWNER.TABLE	Use this option to specify a new owner *or* a new table name for the DDL or data from the old table. This option will essentially cause the source table to be copied, with the source table remaining unchanged. Note that table view names and index names will not be changed, causing failure of these create statements if the original table owner is the same as the new table owner.
NEW OWNER.DBSPACE	Use this option to specify a new DBSPACE for the table. If the NEW OWNER.TABLE option is used, a copy will be performed. If a new table and owner is not specified, the table will be moved from the current DBSPACE to the new DBSPACE.
DATALOAD PCTFREE	Use this option to indicate a percent of each DBSPACE page to be reserved for later insertion when the data is loaded into the table. The default will be to use the current characteristic of the target DBSPACE.
ALTER PCTFREE	Use this parameter to indicate a PCTFREE value to be used after the data has been reloaded into the table. This value should be lower than the PCTFREE parameter value to allow future inserts to use reserved space on existing DBSPACE pages, allowing cluster properties to be maintained.
COMMITCOUNT	Use this parameter to specify how frequently inserts should be committed during the DATALOAD process. This parameter can have a significant effect on performance, locking, and log file usage during the dataload.
REPREP PKGS	The SQLTABLE tool repreps (rebinds) all access modules (packages) during a table reorganization. If this is not wanted, specify NO in this field. The default value is YES.

ALLSTATS	By default, SQLTABLE will use the UPDATE STATISTICS command for the table. If YES is specified for the ALLSTATS parameter, SQLTABLE will use the UPDATE ALL STATISTICS command.
PAUSE	Allows you to interrupt the SQLTABLE function between the DATAUNLOAD and DATALOAD steps. By specifying YES, SQLTABLE will capture data and DDL, but will not perform the DELETE or DROP TABLE function, or the DATALOAD. An exec will be created which will complete the function if executed. The name of this exec will be the first eight alphanumeric characters of the TABLE name. If a temp disk is used when the exec is run, it will not be detached, and the files created during reorganization will remain on disk.
TAPE	Allows you to unload the table data to TAPE rather than DASD. By specifying YES, SQLTABLE will expect the user to have performed the appropriate FILEDEF, LABELDEF, and TAPE MOUNT commands to prepare for the SQLDBSU DATAUNLOAD processing. Remember, TAPE cannot be used for unloading with the REDEFINE option. Please refer to the SQLTABLE Special Considerations section for details on TAPE usage.
FILEMODE	Allows you to specify a linked minidisk that should be used for the captured table data and DDL. This minidisk must be linked in R/W mode prior to running SQLTABLE. If this option is not specified, SQLTABLE will define a temporary disk large enough to hold all unloaded data and will use your 191 A-disk for the created EXEC and DBSU files. Backups of all the DBSU, SYSIN and SYSPRINT files will be made prior to execution. The datafile will be replaced.
REFRESH ROWCOUNT	Specifies that SQLTABLE will perform an UPDATE STATISTICS on the table reorganization so that catalog entries needed to calculate the required temp disk size will be accurate. The UPDATE STATISTICS will not be performed if the FILEMODE, TAPE, or DDL ONLY option has been selected. The default is YES.

After entering all desired SQLTABLE parameters, PF10 can be used to perform the table reorganization immediately.

Usage Consideration: When used in this manner, the tool will run on your virtual machine, thus tying up your ID for the duration of the reorganization job.

Optionally, PF6 can be used to schedule the reorganization activity using the Job Scheduling tool. Refer to Chapter 14, “Job Scheduling Tool” on page 165.

More on the REDEFINE Columns Option

The Definition Change function of the database allows columns to be added, deleted, column order within a table to be changed, column names to be changed, and column datatypes and sizes to be changed. Tape cannot be specified using the SQLTABLE REDEFINE tool.

SQLTABLE uses the DATAUNLOAD/DATALOAD function of the Database Services Utility (DBSU) to transfer data between the previous table definition and the new table definition. Any datatype changes for column definitions are therefore restricted to those that are compatible as described in the *DB2 Server for VSE & VM Database Services Utility* manual under DATALOAD/DATAUNLOAD Data Conversion Summary. The DATALOAD for the new table definition will fail if any of the Data Conversion restrictions are violated.

When the REDEFINE columns option is selected, SQLTABLE will provide the existing table column definitions to you for modification. An example of the display panel format is:

```
****************      SQLTABLE TABLE DEFINITION CHANGE       ***************
                          Source Table: OWNER.TABLE
 Colno   Column Name          Datatype                          Nulls(Y/N)
   1     CCODE                CHAR(    3)                          Y
   2     CUSTNO               CHAR(    6)                          N
   3     CUSTOMER             CHAR(   32)                          Y
   4     NAME                 CHAR(   34)                          Y
   5     ADDRESS              CHAR(   65)                          Y
   7     CITY                 CHAR(   30)                          Y
   8     COUNTRY              CHAR(   25)                          Y
   9     NEW_TEST             CHAR(    5)                          N









      Make desired changes on ALL pages, press PF10 when ready to PROCESS

                               Page 1 of 1
 PF:  1 Help   2 Reset values   3 QUIT
      4 ADD new column         10 PROCESS Table Redefine
```

Figure 198. SQLTABLE REDEFINE Columns Panel

Column Heading	Description
COLNO	COLNO indicates the column position of each column within the original table. These column numbers can be renumbered to modify the relative position of each column in the new table definition. The column number can also be changed to 0 (zero) to indicate that the column should be deleted from the new table definition. (A deleted column will be unloaded from the source table but will not be included in the table definition of the output table, and the unloaded data will not be reloaded.)

Column Name	The Column Name data field can be modified to change the column name of the new table definition. SQLTABLE will update the CREATE INDEX, CREATE VIEW, and GRANT information associated with the original table to reflect the new column names specified.
Datatype	The Datatype data field indicates the datatype of the original table. These datatypes can be changed to any other datatype valid within the database, as long as the restrictions of the DATAUNLOAD/DATALOAD function of DBSU are not violated. These restrictions are discussed in the *DB2 Server for VSE & VM Database Services Utility* manual under DATALOAD and DATAUNLOAD Data Conversion. It is your responsibility to understand which datatypes are compatible and which ones will cause DATALOAD failures or truncation errors.
Nulls (Y/N)	The Nulls (Y/N) data field indicates whether the column in the original table allowed NULLS or was defined as NOT NULL. This column characteristic can be changed, but you should be aware that the DATALOAD will fail if you attempt to load NULL data from the source table into a column that has been redefined as NOT NULL.

REDEFINE - Adding Columns (includes non-nullable fields)

Use selection Option **4** from the SQLTABLE Utility menu to redefine a table's columns. See Figure 198 on page 413 which displays the current table definitions. New columns can be added by pressing PF4.

The panel which follows, Figure 199, will then allow the user to define the new column(s).

```
*****************      SQLTABLE TABLE COLUMN ADDITION       ****************
                        Source Table: OWNER.TABLE
 Colno   Column Name         Datatype               Nulls(Y/N) Default Value
   99   SAMPLE_COLUMN        CHAR(101)                     N  UNDEFINED
 -----  ------------------   ---------------------------   -  ---------------
   10   NEW_FIELD            CHAR(10)                      N  xxxxxx
   11
   13
   14
   15




      Add columns and press PF10 when ready to PROCESS column adds

 PF:  1 Help     3 QUIT (additions)    10 PROCESS Additions
```

Figure 199. SQLTABLE Column Addition Panel

The rows shown that begin with 10, 11, 12, allow inputs from the user to define new columns for the existing table. The first row, with a COLNO value of 10, indicates that the table currently contains 9 columns, so the first column added will be column number 10. The COLNO value can be changed on this screen to add a new column elsewhere in the table, but the provided COLNO value must not be duplicated on a prior panel (you will need to change COLNO values of existing columns before coming to this screen).

To add a new column, enter the column name and datatype in the entry fields under the appropriate heading, and specify Y or N to indicate whether the new column should allow NULLS (Y) or be defined as NOT NULL (N). If Y is specified, there are 2 choices available regarding defaults: 1) If no default value is entered on the panel, the new column will be added to the table and any rows of data which currently exist in the table will set the new column value to NULL; 2) If a default value is specified on the panel, the new column will be initialized to that value for any currently existing rows in the unloaded file. If N is specified, you MUST specify a default value for numeric data in the rightmost entry field of this panel. For CHAR type data, if no default value is specified, a default of blanks will be assumed. The new column will be added to the table as NOT NULL and any existing rows of data will have the default value entered for the new column.

After any new columns are defined, press PF10 to save your definitions and toggle back to the previous "CHANGE TABLE" panel. You can continue switching between the CHANGE TABLE panel and the ADD COLUMNS panel until you press PF10 or PF3 on the CHANGE TABLE panel to PROCESS the table redefinition or CANCEL the redefinition. Note, however, if you press PF3 from the ADD COLUMN panel, all previously entered new columns will disappear if you go back into this function with PF4 from the main table definition column. PF3 from the Table Column Addition panel resets this panel to blanks.

Special Restrictions: Note that SQLTABLE will always attempt a DATAUNLOAD and DATALOAD of the table(s) specified in the REDEFINE function. The DATALOAD will fail if the data stored in the source table does not match the data format specified by a redefined datatype in the object table. If you want a new table without the data from the source table, SQLTABLE should first be run using Option **2** (DDL only) to create a copy of the original table without data. The REDEFINE function (Option **4**) can then be used against the new table without concern about dataload errors.

The REDEFINE function will also attempt to reconstruct all DDL associated with the table, including index and view definitions and all grants. If the owner of the new table is the same as the owner of the source table, but the table name is different, the INDEX and VIEW create statements will *fail* due to duplication (non-uniqueness) within the database. The PAUSE option can be used to capture the INDEX and VIEW create statements within the DBSU format and allow the index and view names to be manually modified before completing the execution of the SQLTABLE function.

Special Considerations

SQLTABLE and LONG VARCHAR Fields

You cannot use SQLTABLE to unload/reload (reorg) the QMF table Q.OBJECT_DATA. The run will be stopped after a warning message is issued. The reason for this is that the database DBSU suppresses blanks at the end of VARCHAR and LONGVARCHAR fields when doing a DATALOAD (which is what SQLTABLE uses). This action renders QMF FORMS and some PROCS and QUERIES unusable resulting in loss of QMF objects. In order to prevent this situation, SQLTABLE will display a warning message and stop the processing. You can still use the SQLREORG tool to UNLOAD/RELOAD the DBSPACE in which this table resides (PUBLIC.DSQTSCT3).

Referential Integrity Considerations

- If the table being reorganized has a deactivated primary key, any dependent tables (that is, those with foreign keys) will not be able to have their foreign keys created.
- If the table being reorganized has an **implicitly** deactivated foreign key, it cannot be created.
- If the table being reorganized has an **explicitly** deactivated foreign key, it can be created, then deactivated, **IF** the parent table's primary key is active.
- If the table being reorganized has an active primary key, any dependent table with an explicitly deactivated foreign key **may** be able to have its foreign key created **AFTER** the primary key table is loaded. The create will fail if the dependent table's data is not correct. If the create is successful, the key will then be deactivated.

Unloading to Tape

When using the TAPE option, you must issue the appropriate FILEDEF, LABELDEF, and TAPE MOUNT commands prior to invoking SQLTABLE. The DDNAME for the unloaded data *must* be DBSFILE. SQLTABLE executes the DATAUNLOAD utility. DATAUNLOAD supports all record format (RECFM) values with the exception of undefined (U). For instance, an example of a FILEDEF and LABELDEF command are given below:

```
FILEDEF DBSFILE TAP1 SL (RECFM VBS BLKSIZE 28672
LABELDEF DBSFILE VOLID SCRATCH
```

If the unloaded data will fit on a single tape, then SQLTABLE can be executed without the PAUSE option, and the tape will automatically be rewound between the DATAUNLOAD and DATALOAD steps.

If the unloaded data will span multiple tapes, then the proper LABELDEF command must be issued to handle the multiple volume tape and the PAUSE option must be specified so that the appropriate sequence of DETACH and MOUNT commands can be issued between the DATAUNLOAD and DATALOAD processing steps. After the PAUSE, the last tape must be detached and the first tape must again be mounted before the DATALOAD process can begin.

Failure Restart

SQLTABLE will display status information at your terminal to indicate what step is being performed. If SQLTABLE fails at any time prior to the "Reloading..." message, the table has not been changed and SQLTABLE should be restarted from the beginning. If the failure occurs during the "Unloading..." step and you are using a temporary disk, you can detach the temporary disk by issuing the CMS RELEASE command with the DET option.

If SQLTABLE fails at any time after the "Reloading..." message has been displayed, the table must be recovered using the created REXX exec and related files. This exec will begin the dataload process by using the DBSU files and the unloaded data file. If the data was unloaded to TAPE, then the proper FILEDEF, LABELDEF, and tape MOUNT commands must be used before the created exec can be invoked.

Note:

If SQLTABLE unloads data to a temporary disk and SQLTABLE fails, you should be aware that the temporary disk (and unloaded data) will be purged if you log off. Therefore, the created exec *must* be run successfully before logging off.

As a special precaution to prevent loss of data, SQLTABLE sends the unloaded data file to your reader if a temporary disk is used. If the temporary disk is lost, the reader copy can be received onto a new disk and the created exec can then be executed. Under this scenario, you should verify that all file mode references in the created exec correctly point to the new location of the data file.

Pause Option

The same cautions discussed in “Failure Restart” also apply to the use of the SQLTABLE PAUSE option. Each time that SQLTABLE is run, a new exec and related files will be created. Any previous copy of these files will be renamed by reversing the filename and filetype and reversing all letters in each. This implies that if a second SQLTABLE is executed prior to completion of the first SQLTABLE, the first one can be used again if the files are renamed back again.

The PAUSE option can be very useful to an experienced user of SQLTABLE and very dangerous to an inexperienced user. If you have a good working knowledge of the Database Services Utility (SQLDBSU) and of SQLTABLE, the PAUSE option will allow you a great deal of flexibility and control in the SQLTABLE functions. The SQLDBSU file can be examined and modified prior to running the created exec in order to change or remove certain characteristics of the table. Once again, care should be exercised to prevent loss or contamination within the database.

Unloaded Data File

SQLTABLE will unload the database data within a table to an OS format file on a CMS minidisk or tape using the DBS Utility DATAUNLOAD command. This unloaded format file can be modified using XEDIT or other file editors to modify the data prior to the reload. Caution should be exercised to ensure that this data file is not modified in such a way that will cause the DATALOAD to fail. The default XEDIT setting of NULLS ON will cause automatic truncation of lines within the unloaded file, which will result in the file being corrupted.

Failure Analysis

SQLTABLE uses a DBS Utility command file to execute the DATAUNLOAD portion of the reorganization. Detailed output from the DATAUNLOAD portion is captured in a file on your 191 A-disk with a file type of UNLIST. This file can be examined to determine the reason for failure.

During the "Reloading..." portion of SQLTABLE processing, the SQLDBSU command file is executed using the DBS Utility. Detailed output of this processing is captured within a file on the target disk with a filetype of LOADLIST or DDLLIST and a filename equal to the first eight alphanumeric characters of the table name. If a failure occurs during this step of processing, this file can be examined to determine the cause of failure.

One fundamental difference between SQLREORG, the DBSPACE Reorganization tool, and SQLTABLE is that SQLREORG uses the UNLOAD/RELOAD commands of DBSU, while SQLTABLE uses DATAUNLOAD/DATALOAD. This provides several advantages to the SQLTABLE user. The RELOAD command performs the entire data load function as a single Logical Unit of Work (LUW). If the table or DBSPACE is very large, the entire LUW can be too large to be recorded within the database log file, forcing an archive in the middle of the RELOAD. The archive cannot take place until a checkpoint is taken, which requires a breakpoint without any active LUWs. The RELOAD LUW must therefore be rolled back by the database before the archive can complete. By using the DATALOAD command with the COMMITCOUNT option, SQLTABLE can successfully perform a table-level reorganization with an archive during the process.

Another SQLTABLE advantage is the use of a simple CMS file format by DATAUNLOAD that allows modification prior to the DATALOAD. The disadvantages of SQLTABLE mostly relate to the pure reorganization function. UNLOAD/RELOAD requires less external disk space than DATAUNLOAD/DATALOAD and will also execute faster. The DBSPACE level processing of SQLREORG also provides a considerably more efficient reorganization than the table-level processing of SQLTABLE, allowing empty pages to be returned to the database for use elsewhere and providing much better clustering properties to be maintained.

Chapter 33. Index Reorganization Tool

Overview

A database index is a set of pointers, logically ordered on a column or set of columns in a table, that exist to provide faster access and force uniqueness of rows within a table. Indexes are stored on index pages in the same DBSPACE as their associated tables. Control information about indexes is stored on header pages in the DBSPACE.

Extensive modifications to a table can fragment its indexes. This can lead to increased I/O and execution time whenever those indexes are used.

The Index Reorganization tool automates index maintenance by analyzing the indexes in your databases and/or reorganizing those that can benefit from it. You can select the indexes to be examined by:

1. Owner
2. DBSPACE name
3. Number of pages

To invoke the Index Reorganization tool through the panel interface,

1. Enter Option **U** on the Main Menu and press ENTER.
2. Enter Option **RI** on the Database Utility Functions screen and press ENTER.

To invoke the Index Reorganization tool directly from the CMS READY prompt:

1. Type SQLRINDX and press ENTER.

Note:

1. The user must own the indexes or have DBA authority in the target database.
2. The user must have at least level 3 (Administrator) authority on the service machine for that database.

Indexes are reorganized using these methods:

1. Primary key indexes and unique indexes are reorganized using the ALTER TABLE statement.
2. Other active, valid indexes are reorganized using the database DBSU REORGANIZE INDEX command.

Note:

System catalog table indexes are analyzed BUT NOT REORGANIZED. Use Control Center's system catalog index reorganization utility to reorganize system catalog indexes. (Option **RC** on the Database Utility Functions menu).

The product index reorganizations run in multiple user mode (with the database up). They may execute on your own virtual machine or on a support machine. Remember that you or the support machine must be either the owner of the index or have DBA authority in the target database.

Features

The Index Reorganization tool:

- Runs in multiple user mode
- Lets you select indexes by:
  - Owner
  - DBSPACE name {with wildcard}
  - Number of DBSPACE pages
- Generates a list of indexes requiring reorganization
- Reorganizes indexes based on a previous candidate list
- Analyzes and performs the reorganizations automatically
- Allows scheduling of analysis and/or reorganization
- Provides an index analysis report (LISTINDEX)
- Provides an index reorganization report (INDEXPRIOR or RUNINDEX)

How the Index Reorganization Tool Works

The Index Reorganization tool uses the system catalogs to access information about an index. It calculates the number of pages that each index should occupy according to the guidelines given in Appendix "A" of the DB2 Database Administration Manual. Only valid, active indexes, primary keys, and unique constraints are considered in these calculations. The total of all calculated index pages is then compared to the actual number of pages the indexes occupy in the DBSPACE (from SHOW DBSPACE). If indexes occupy more than the calculated number of pages, the indexes in that DBSPACE are considered candidates for reorganization.

Two ranking factors are used to assign a relative need of reorganization for the indexes in a DBSPACE. They are:

1. Percent of index pages to be reclaimed by index reorganization:

   (Actual index pages - calculated index pages) / total index pages

2. Number of index pages to be reclaimed by index reorganization:

   Actual index pages - calculated index pages

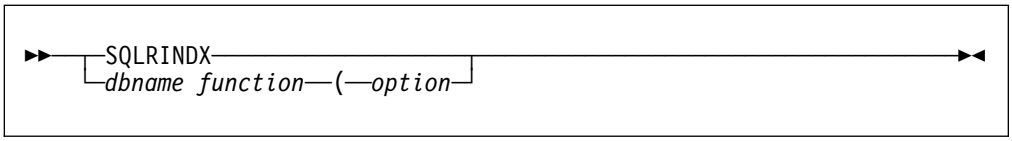
Indexes in DBSPACES with higher ranking factors are reorganized first.

Note:

1. Page calculations for indexes on variable length columns are over estimated in accordance with the database guidelines. Over estimation keeps an index from being reorganized when it is not needed.
2. Only valid and active indexes are reorganized.
3. Up-to-date Statistics will help ensure accurate LISTINDEX calculations.
4. PCTFREE retains its previous value.

Command Mode Invocation

To execute the Index Reorganization tool in command mode, the syntax is:



where:

Parameter	Description
dbname	is the machine name (not the database name) of the database where the maintenance will be performed.
function	LISTINDEX, INDEXPRIOR, or RUNINDEX (See “Functions” on page 422).
options	(See the list of the valid parameters as described in “Options” on page 422).

If no parameters are specified, the Index Maintenance Utility Screen is displayed.

Index Maintenance Utility Screen

The Control Center Index Maintenance Utility Screen is shown in Figure 200 on page 422. The first two parameters, FUNCTION and DATABASE, are required. They identify the function you want to perform and the database against which you want to perform it. If you are initialized to a database, that name will appear in the DATABASE field.

Depending on the function selected, the parameters below the line labeled "Options" will allow you to limit the scope of candidate selection and set time or quantity based execution limits.

```
*************    SQLRINDX - DATABASE INDEX MAINTENANCE UTILITY  **************

FUNCTION ==>  LI       LI - List Index Reorg Candidates
              --       IP - Reorg Indexes using Prior List
                        RI - Evaluate and Reorg Indexes

DATABASE ==>  SQLDBA
              --------
    ----------------------------  Options  -----------------------------
DBSPACE OWNER ==> SQLDBA             ( ALL, PUBLIC, PRIVATE, ownername )
                  ------------------
DBSPACE NAME  ==>                    ( blank for ALL, use % for wildcard )
                  ------------------


TIME LIMIT  ==>  180    (Minutes)        QTY LIMIT ==> 10    (Max. Run Qty)
                 -----                                 -----


DBSPACE PAGES:  MINIMUM ==> 0          MAXIMUM     ==> 999999
                            -------                    -------
 PF1 HELP      PF3 QUIT       PF6 Schedule        PF10 Process Immediately
```

Figure 200. Index Maintenance Utility Screen

Functions

Each Index Reorganization function generates report and output files which are stored on the service machine for that database. Refer to “Files and Reports” on page 423 for a description of these files. The function field can have these values:

Function	Description
LI (List Index)	Analyze indexes in the selected DBSPACES and list the DBSPACES with indexes requiring reorganization.
IP (Index Previous)	Reorganize indexes in DBSPACES identified as candidates in a previous evaluation by LI (List Index) or RI (Run Index). The previous candidate list (dbname INDXMAIN) must exist on the A disk of the service machine. The options selected will determine which DBSPACES in the INDXMAIN file will be selected. Reorganization proceeds until the TIME or QTY LIMIT is reached.
RI (Reorganize Indexes)	Analyze indexes in the selected DBSPACES and create a list of candidate indexes. Reorganize indexes based on the Options selected until the TIME or QTY LIMIT is reached.

Options

These options apply in panel interface as well as command mode.

Option	Description
DBSPACE OWNER	Used with the LI, IP, and RI functions. Specifies whether ALL, PUBLIC, PRIVATE, or specific DBSPACE owner(s) are selected for analysis or reorganization. The percent sign (%) can be used at the beginning or end of a specified DBSPACE owner to select LIKE DBSPACE owners. Default is ALL. Command mode syntax: OWNer=
DBSPACE NAME	Used with the LI, IP and RI functions. Specifies a DBSPACE name or group of names to be selected for analysis. The percent sign (%) can be used as at the beginning or end of a DBSPACE name to select LIKE DBSPACES. Default is ALL. Command mode syntax: DBSPname=
TIME LIMIT	Used with the IP and RI functions. Limits reorganization to a specific amount of time in minutes. Once this limit is reached, no more index reorganizations are started. Default is 180 minutes. Command mode syntax: TIME=
QTY LIMIT	Used with the IP and RI functions. Limits index reorganization to the specified number of DBSPACES. Default is 10. Command mode syntax: QTY=
DBSPACE PAGES	Used with the LI, IP and RI functions. Limits index analysis (LISTINDEX) or reorganization (INDEXPRIOR, RUNINDEX) to DBSPACES that fall within the specified range of pages. Default minimum is 0, default maximum is 999999. Command mode syntax: DBMIN=, DBMAX=

Files and Reports

Each index reorganization function generates report and output files which are created on the virtual machine where the Index Reorganization is run and sent to the service machine for that database.

The index reorganization files remain on the "A" disk of the user or Control Center support machine and the Control Center service machine. The files on the service machine are used for future index maintenance and to ensure that:

- Index analysis and reorganizations will not be needlessly duplicated by other DBA's
- All DBA's for a database will have easy access to the same index reorganization information.

Each file used in the Index Reorganization process is described below:

File	Description
dbname INDXPICK	Created during the LISTINDEX and RUNINDEX functions. Contains a list of DBSPACES and indexes selected for reorganization. This is kept as a record of LISTINDEX selections and not updated by INDEXPRIOR or RUNINDEX functions. Renamed by reversing the filetype when another LISTINDEX is performed against the same database. Overwritten on the service machine.
dbname INDXMAIN	Created by both the LISTINDEX and RUNINDEX functions. Contains a list of DBSPACES and indexes selected for reorganization. Sent to the service machine and the user machine executing INDEXPRIOR. Returned to the service machine when processing is completed. Updated by removing entries reorganized by INDEXPRIOR and RUNINDEX. Deleted when all entries have been removed. Renamed by reversing the filetype when another LISTINDEX is performed against the same database. Overwritten on the service machine.
dbname XXXXINDX	Created by LISTINDEX function where XXXX = the first 4 characters of OWNER. Summarizes the DBSPACE index analysis and reports whether a DBSPACE has been chosen as a candidate for reorganization. Replaced when another LISTINDEX is performed on the same database with the same OWNER. Backed-up on the user machine's A disk before being overwritten.
dbname RUNINDX	Created by the INDEXPRIOR and RUNINDEX functions. Shows the result of index reorganization on a DBSPACE's index pages.
SQLRINDX DBSUERR	Created during INDEXPRIOR and RUNINDEX. Contains a listing of any DBSU errors encountered. Backed up by reversing the filetype before being over written. Remains on the user's A disk and is not kept on the service machine.

Sample Index Analysis Report

The figure which follows is an example of a DBSPACE Index Reorganization Candidates Report.

```
               DBSPACE INDEX REORGANIZATION CANDIDATES REPORT
       Dbname: WMAVM1.SQL34DBA                    CNTRLID: SQMSTTS1
                            03/13/97 08:43:58

               OPTIONS: Dbspaces: ALL.ALL, min=0, max=999999,
                                qty=3, time=180

  DBSPACES EXAMINED = 23
  ************************************************************

  CRITERIA 1 = Percent of Index Pages to be Reclaimed by Reorg

  CRITERIA 2 = Number of Index Pages to be Reclaimed by Reorg


*******************************************************************************
ANDYS.ANDYS                    DBSPACENO: 9

                            NBR PAGES  OCCUPIED PAGES   %FREE  EMPTY
        SHOW DBSPACE INDEX:     200    100 (    50%)     98%    0
          CALCULATED INDEX:              50 (    25%)
                   RECLAIM:              50 (    25%)


          Index reorganization candidate
*******************************************************************************
RAYM.RAYM                      DBSPACENO: 12

                            NBR PAGES  OCCUPIED PAGES   %FREE  EMPTY
        SHOW DBSPACE INDEX:      84     17 (    20%)     45%    0
          CALCULATED INDEX:             17 (     0%)
                   RECLAIM:              0 (     0%)

          Actual occupied index pages are less than/equal
          calculated index pages.  Indexes will NOT be reorganized
*******************************************************************************

PUBLIC.SYS0001                 DBSPACENO: 1

                            NBR PAGES  OCCUPIED PAGES   %FREE  EMPTY
        SHOW DBSPACE INDEX:    7680   4433 (    57%)     61%    0
          CALCULATED INDEX:            113 (     0%)
                   RECLAIM:           4320 (    56%)


          WARNING: This DBSPACE contains system indexes.
          Only the non-system indexes in this dbspace will
          be reorged by SQLRINDX. Use CTRLCTR  Reorganize
          Catalog Index tool to reorganize System catalog indexes.
          Index reorganization candidate.

*******  THERE ARE 2 CANDIDATES FOR REORGANIZATION  *****
```

Figure 201. DBSPACE Index Reorganization Candidates Report

In the example above, all 23 DBSPACES in the database were included. QTY= 3 was specified so analysis ceased after the third DBSPACE. Two DBSPACES were identified as candidates for index reorganization.

Sample Index Reorganization Report

The figure which follows is an example of a DBSPACE Index Reorganization Report.

```
                    DBSPACE INDEX REORGANIZATION REPORT
                         Database: WMAVM1.SQL34DBA            03/13/97
11:33:50


                                  ELAPSE              CALC
OWNER    DBSPACENAME        SPNO  TIME  ST   W1   W2 IPGS INDEX STATISTICS
_____    __________________ ____ ______ __ ____ ____ ____ _________________

ANDYS    ANDYS                 9   0.01 00   0%    0   50 200 100(50%) 98%
                                     ****** After REORG: 200  50(25%) 98%



Number of dbspaces successfully reorged =  1

Total elapsed time for all reorgs = 1.00 minute


                End of Report for  SQL34DBA  INDEX REORG JOB


NOTE : SPNO  - dbspace number
       ELAPSE- elapsed time, in minutes, for the REORG
       W1    - Percent of index pages reclaimed by reorg
       W2    - Number of index pages reclaimed by reorg
       CALC IPGS - Number of pages indexes should occupy
       ST    - status of the REORG
               '00' Good Reorg
               'CC' Good Reorg but calculated index pages
                    were under estimated
               'SY' Good Reorg but System Catalog Indexes
                    in this Dbspace were not Reorged
               'TT' Not enough time left to reorg this space
               'XX' Bad Index Reorgs occurred in this dbspace
       DATA/INDEX BEFORE - the results of a SHOW DBSPACE
                command before the REORG was done, format:
                    tttt uu(pp%) ff%
                where:  tttt is total pages,
                        uu   is pages used,
                        pp   is percent of pages used,
                        ff   is avg. percent free per page.
       DATA/INDEX AFTER  - the results of a SHOW DBSPACE
                command after the REORG was done
```

Figure 202. DBSPACE Index Reorganization Candidates Report

Chapter 34. Automated DBSPACE Maintenance Tools

Overview

Within the database, the OPTIMIZER uses catalog statistics to choose the best path to the data within the database. Due to the overhead involved, the database does not automatically update this statistical data during normal database update activity. Without reasonably accurate statistics, the optimizer can choose a less efficient method of retrieving data, reducing the overall database performance. It is therefore the responsibility of the DBA to assure that statistics are updated on a regular basis so that database performance will be maximized.

Another common database problem which impacts performance is the gradual fragmentation and poor organization of data over a period of time. After many inserts, updates, and deletes, the data within a database will become spread over many physical pages, with many gaps in between and with the physical sequence of data different than the logical sequence of data. Periodic reorganization is required to pack the data on the minimum number of physical pages and in a logical sequence that optimizes data retrieval.

The Automated DBSPACE Maintenance tools are known collectively as the SQLMAINT program with its associated functions. They are part of the Database Administration tools. The purpose of the Automated DBSPACE Maintenance tools is to provide a means to automatically monitor DBSPACES and perform maintenance upon them to improve performance. These tools are designed for automatic execution on a scheduled basis on a Control Center support machine, performed at a routine interval with a selected set of parameters.

There are two types of DBSPACE maintenance tools: Statistics Maintenance and DBSPACE Reorganization Maintenance. This chapter explains the steps needed to install SQLMAINT, then explains the function and use of each of the two types of tools.

SQLMAINT jobs should *not* be scheduled to execute on the service machine, since all automated operations functions with *all* databases will be disabled while the maintenance functions execute.

Before You Begin

Use SQLMAINT during non-peak hours to prevent locking contention with other users of the database. Due to the intensive updating of the database system catalogs, use of this tool during periods of high database use can lead to lock contention.

The parameters used when SQLMAINT is executed will determine the number of DBSPACES that will have maintenance applied. These parameters should be carefully selected to assure that maintenance does not run into peak database use periods.

How the Automated DBSPACE Maintenance Tools Work

SQLMAINT automates the UPDATE STATISTICS and the REORGANIZATION processes for a database by keeping track of these activities at the DBSPACE-level and executing these functions where required based on specified execution parameters.

SQLMAINT performs a detailed analysis of each DBSPACE to determine the degree of need for reorganization. The results of this analysis are provided in a hardcopy report for study by the DBA. SQLMAINT can be optionally instructed to automatically reorganize one or more DBSPACES based on the degree of need determined by the analysis. The number of DBSPACES reorganized can be limited by specified quantities or elapsed time limits.

By setting up SQLMAINT to run on a regular schedule, database maintenance can be performed automatically during periods of low database use (e.g., at night or on weekends). SQLMAINT uses a database table to maintain use information for each DBSPACE in the database. When SQLMAINT executes, each DBSPACE is considered for maintenance based on parameters specified by you for that particular execution. After execution, the information within the SQLMAINT table is updated to reflect the changes which have occurred.

When a maintenance function of SQLMAINT is invoked, these steps are performed:

1. Link and establish database communications to the target database.
2. Verify that the user has DBA authority.
3. Refresh the SQLMAINT table with new data from SYSTEM.SYSDBSPACES.
4. Select a list of DBSPACES based on selection parameters.
5. Create the report or perform the maintenance as specified.
6. Update the SQLMAINT table for each maintained DBSPACE.

Who Can Use the Automated DBSPACE Maintenance Tools

Use of these tools requires Control Center database Administration-level or greater authorization.

Installing the Automated DBSPACE Maintenance Tool for Operation

You must have database DBA connect authorization to install the SQLMAINT tool into each target database using the install function of the SQLMAINT panel interface. These steps are required to install the SQLMAINT too. They must be completed for each database where SQLMAINT will run. A detailed description of each step follows.

1. Make SQLMAINT's files available to subsequent users.
2. Acquire a public DBSPACE in which the tool is to be installed.
3. Run the Install option from the Control Center panel interface.

Step 1. Make SQLMAINT's Files Available to Subsequent Users

The SQLMAINT tool consists of these files, which must be placed on a CMS minidisk available to SQLMAINT users:

Note: These files are part of the Control Center code and reside on the Control Center service machine's code disk, which is typically accessible to users.

Filename	Description
SQLMAINT EXEC	Master tool which performs the maintenance activities.
SQLMAIN$ XEDIT	XEDIT macro which generates a parameter entry panel.
SQLMAINX XEDIT	XEDIT macro used in generating panels.
SQLMAINT HELPCMS	CMS help file for SQLMAINT.
SQLREORG EXEC	Actual execution of the reorganization function will require all files belonging to the DBSPACE Reorganization tool, including the DBINIT CONTROL file.

Step 2. Acquire a Public DBSPACE in which the Tool is to be Installed

The installation procedure will create a public table within the database (SQLMAINT.SQLMAINT_TABLE). A public DBSPACE must be acquired and available prior to invoking the install process. The DBSPACE name can be any valid database name.

The required size of the public DBSPACE will vary depending on the number of DBSPACES within the target database. The SQLMAINT_TABLE will contain a row for each acquired DBSPACE in the database, which will be used to maintain information about reorganizations done against those DBSPACES using the SQLMAINT tool.

Step 3. Run the Install Option from the Control Center Panel Interface

Refer below to “DBSPACE Maintenance Utility Panel” on page 431.

Invoke the panel interface (SQM) of the Control Center; choose Option **U** (Database Utilities); then Option **M** (SQLMAINT). At the ensuing panel, select Option **I** (Install); fill in the database name and then press PF10 to initiate the installation process.

The SQLMAINT installation process will create a public table in the target database as shown in Figure 203 on page 430. In addition, a row for each non-system-owned DBSPACE will be entered into this table during the install process.

```
CREATE TABLE "SQLMAINT"."SQLMAINT_TABLE" (
  "OWNER"          CHAR(8)        NOT NULL
 ,"DBSPACENAME"    VARCHAR(18)    NOT NULL
 ,"DBSPACENO"      SMALLINT       NOT NULL
 ,"FREEPCT"        SMALLINT       NOT NULL
 ,"PCTINDX"        SMALLINT       NOT NULL
 ,"UPSTAT_DATE"    CHAR(7)        NOT NULL
 ,"UPSTAT_TIME"    CHAR(8)        NOT NULL
 ,"UPSTAT_ELAPSED" SMALLINT       NOT NULL
 ,"REORG_DATE"     CHAR(7)        NOT NULL
 ,"REORG_TIME"     CHAR(8)        NOT NULL
 ,"REORG_ELAPSED"  SMALLINT       NOT NULL
 ,"REORG_FREEPCT"  SMALLINT       NOT NULL
 ,"REORG_PCTINDX"  SMALLINT       NOT NULL
 ,"REORG_STATUS"   CHAR(2)        NOT NULL
 ,"REORG_WEIGHT"   SMALLINT       NOT NULL
 ,"NPAGES"         INTEGER        NOT NULL
 ) IN "PUBLIC"."SQLMAINT";
```

Figure 203. SQLMAINT Table Definition

SQLMAINT Control Table

When SQLMAINT is installed on a database, a control table will be created in a public DBSPACE to store information about maintenance activities. This control table can be used as a basis for creating maintenance reports and can also be updated manually to further control the maintenance process (such as changing the reorganization dates to prevent a large DBSPACE from being reorganized automatically).

The name of the control table defaults to SQLMAINT.SQLMAINT_TABLE unless the SQLOWNER and SQLTNAME variables at the beginning of the SQLMAINT EXEC are changed at your installation. The columns of this table are:

Column Name	Description
OWNER	Owner of the DBSPACE (from SYSTEM.SYSDBSPACES)
DBSPACENAME	Name of the DBSPACE (from SYSTEM.SYSDBSPACES)
DBSPACENO	DBSPACE number (from SYSTEM.SYSDBSPACES)
FREEPCT	FREEPCT value (from SYSTEM.SYSDBSPACES)
PCTINDX	PCTINDX value (from SYSTEM.SYSDBSPACES)
UPSTAT_DATE	Date of the last UPDATE STATISTICS
UPSTAT_TIME	Time of the last UPDATE STATISTICS
UPSTAT_ELAPSED	Elapsed time of the last UPDATE STATISTICS
REORG_DATE	Date of the last SQLREORG
REORG_TIME	Time of the last SQLREORG
REORG_ELAPSED	Elapsed time of the last SQLREORG in minutes
REORG_FREEPCT	FREEPCT value of the last SQLREORG
REORG_PCTINDX	PCTINDX value of the last SQLREORG
REORG_STATUS	A two-character indicator of SQLREORG success
REORG_WEIGHT	An integer weight of SQLREORG need
NPAGES	Size of the DBSPACE

The SQLMAINT.SQLMAINT_TABLE can be viewed at any time to examine the results of SQLMAINT activities. For a discussion of how to use this table, see

"Using the SQLMAINT Control Table" on page 446. The SQLMAINT_TABLE is refreshed each time the SQLMAINT function is executed. If you run SQLREORG, the SQLMAINT_TABLE will also be updated. For example:

Column Name	DBSPACE Attribute
FREEPCT	ALTER PCTFREE
PCTINDX	PCTINDEX (current)
REORG_FREEPCT	PCTFREE used to create the DBSPACE
REORG_PCTINDX	PCTINDEX (current)

SQLMAINT Installation Output Example

Shown below is example output from the SQLMAINT installation process.

```
SQLMAINT version V6.1 03/16/98 13:28:56
INSTALL function executing...
Enter a public DBSPACEname for the installation
(Or just press ENTER to cancel)
tools
03/16/97 13:29:04 Dropping old SQLMAINT.SQLMAINT_TABLE...
03/16/97 13:29:05 Creating new SQLMAINT.SQLMAINT_TABLE...
13:29:06 Refreshing SQLMAINT.SQLMAINT_TABLE...
13:29:06 Deleting SQLMAINT.SQLMAINT_TABLE names not on SYSTEM.SYSDBSPACES...
13:29:06 Refreshing data from SYSTEM.SYSDBSPACES...
13:29:06 0 SQLMAINT.SQLMAINT_TABLE rows updated from SYSTEM.SYSDBSPACES
13:29:06 Get DBSPACE names not in SQLMAINT.SQLMAINT_TABLE...
13:29:57 85 rows from SYSTEM.SYSDBSPACES added to SQLMAINT.SQLMAINT_TABLE
  SQLMAINT INSTALL completed
SQLMAINT ended 03/16/97 13:29:58
Ready; T=3.01/4.97 13:29:58
```

Figure 204. SQLMAINT Installation Output Example

The SQLMAINT install function primes the SQLMAINT_TABLE with entries of all private and non-system-owned public DBSPACES. All date and time columns are set to zeroes. The table is now ready for use by the other functions of SQLMAINT.

DBSPACE Maintenance Utility Panel

The entry panel of the Automated DBSPACE Maintenance tools is shown in Figure 205 on page 432.

```
**********************   DBSPACE MAINTENANCE UTILITY    ************** *********

  FUNCTION ==>  LS      ( I, LS, US, LR, RP, RR ) - (Install, List Statistics,
                --       Update Statistics, List Reorg Candidates,
                          Reorg using Prior List, Run Reorgs - Re-evaluate)
  DATABASE ==>  SQMDBA
                --------
  ----------------------------  Options  --------------------------------
  DBSPACE OWNER ==> ALL                 (ALL, PUBLIC, PRIVATE, ownername)
                    ------------------
  DBSPACE NAME  ==>                     (blank for ALL, use % for wildcard)
                    ------------------
  SEQUENCE      ==> WEIGHT (AGE, WEIGHT)   DAYS SINCE  ==> 14
                    ------                              ---
  TIME LIMIT    ==> 180    (Minutes)       QTY LIMIT   ==> 10     (Max. Run Qty)
                    -----                               -----
  STOP ON FAIL  ==> NO     (YES or NO)     REPREP PKGS ==> YES    (YES/NO)
                    ---                                 ---
  PAGES: MIN    ==> 0      MAX   ==> 999999    COMMITCOUNT ===> 0
                                   -------                    -------
  FILEMODE    ==>    (Reorg work disk)      ALLSTATS  ==> NO   (YES/NO)

  PF1 HELP  PF3 QUIT   PF6 SCHED   PF8 SEL CRITERIA   PF9 VIEW TAB   PF10 PROCESS
```

Figure 205. Automated DBSPACE Maintenance Tools Entry Screen

Most fields are initialized to some values. The first two parameters (FUNCTION and DATABASE) are required. They will identify what function you want to perform and what database the function will be performed upon. These fields are initialized to LS and SQLDBA, respectively. However, if you are currently initialized to a database, that name will appear in the DATABASE field.

Display Fields

Functions

The function field can have these values:

Function	Description
Install (I)	This should only be used once, when SQLMAINT needs to be installed into a new database. If this function is used again, all previous maintenance data will be erased.
List Statistics (LS)	This function lists DBSPACES which need to have statistics updated. It will produce a report but will not perform any maintenance.
Update Statistics (US)	This function will perform UPDATE STATISTICS on DBSPACES that meet the specified criteria.
List Reorg Candidates (LR)	This function will list candidate DBSPACES that require reorganization based on specified criteria. It will produce a report but will not perform any maintenance.
Reorg using Prior List (RP)	This function will reorganize DBSPACES based on candidates chosen by a previous evaluation list (generated either by the LS or RR function). The

	previous candidate list *must* be available on the A-disk in file database SQLMAINT. The essential purpose of this option is to allow one execution of the candidate evaluation process using either LS or RR, with subsequent multiple reorganization runs based on the single prior evaluation. This can save considerable DBSPACE examination time, but should be used with *caution* since the evaluation data can quickly become invalid.
Run Reorgs - Reevaluate (RR)	This function will reorganize candidate DBSPACES that meet the specified criteria. It will search all DBSPACES designated by the criteria looking for candidates, generate a list of the candidates and perform the reorganizations.

Options

The parameters below the line labeled "Options" allow you to control the maintenance functions by restricting the number and type of DBSPACES that will be selected.

Parameter	Description
DBSPACE OWNER	Can be used with the LS, US, LR, and RR functions to specify whether PUBLIC, PRIVATE, ALL, or specific DBSPACE owner(s) will be selected for analysis. The database wildcard character (%), can be used at the beginning or end of a specified DBSPACE OWNER to select like DBSPACE OWNER names.
DBSPACE NAME	Can be used with the LS, US, LR, and RR functions to specify a DBSPACE name or group of similar DBSPACE names that should be selected for analysis. The database wildcard character (%), can be used at the beginning or end of a specified DBSPACE name to select like DBSPACE names. This parameter defaults to ALL if not specified.
SEQUENCE	Is only applicable to the RR function and controls the order in which the selected DBSPACES will be reorganized. AGE will perform reorganization on DBSPACES sequenced according to the last date that each DBSPACE was reorganized (oldest to newest). WEIGHT will perform reorganization according to a calculated weight which indicates how much each DBSPACE needs the maintenance. WEIGHT is the default if this parameter is not specified.
DAYS SINCE	Applicable to the LS, US, LR, and RR functions. It is the primary method of controlling the automated execution of maintenance activities. It specifies the number of days since the previous UPDATE STATISTICS or REORGANIZATION that must have passed before the same maintenance will be performed again on a DBSPACE.

TIME LIMIT	Applicable to the US, RR, and RP functions. This limit the number of minutes during which maintenance will be performed. After this time limit has passed, SQLMAINT will not begin maintenance on any new DBSPACE candidates but will complete maintenance on any DBSPACE that is currently in process. The default is 180 minutes (3 hours).
QTY LIMIT	Applicable to the US, RR, and RP functions. Will limit the number of DBSPACES that will be maintained.
STOP ON FAILURE	Applicable to the RR and RP functions. Enter YES if SQLMAINT should STOP if a DBSPACE reorganization fails; enter NO if SQLMAINT should continue with subsequent DBSPACE reorganizations if a failure occurs.
REPREP PKGS	Applicable to the RR and RP functions. Enter NO if SQLREORG should *not* reprep (or rebind) the access modules (packages) when doing a DBSPACE reorganization. The default is YES, which means reprepping (rebinding) will occur.
DBSPACE PAGES	Applicable to the LS, US, LR, and RR functions, the MINIMUM and MAXIMUM DBSPACE sizes (expressed in pages) can be specified to limit the DBSPACES selected for analysis.
COMMITCOUNT	Applicable to the RR and RP functions. Allows you to specify a value from 1 to 2147483647 to be used in the RELOAD processing. A value of 0 means to disregard the parameter.
FILEMODE	Applicable to the RR and RP functions. This optional parameter is used to specify a minidisk, linked in WRITE mode, that should be used as work space for the unloaded DBSPACE data for reorganizations. The default is to acquire and use a temporary disk for the DBSPACE and to use the 191 A-disk for the DBSU file and listing file.
ALLSTATS	Applicable to the RR and RP functions. When doing DBSPACE reorganization maintenance (SQLREORG), you can choose to have it generate the UPDATE ALL STATISTICS DBSU command if this parameter is set to YES. Otherwise, the default UPDATE STATISTICS command is used.

PF Keys

The PF Keys offer these options:

PF Key	Description
SEL CRITERIA (PF8)	Use this key to view the selection criteria which will be used for determining whether the DBSPACES selected by the main panel options also meet the criteria for being selected as a reorganization candidate. You can also change any of the criteria options to affect the selection process.
VIEW TAB (PF9)	Choose this key to view the SQLMAINT table entries.

Statistics Maintenance Tool

The Statistics Maintenance tool consists of the List Statistics and Update Statistics functions.

How the Statistics Maintenance Tool Works

The main objective of this tool is to choose DBSPACES which need to have UPDATE STATISTICS run based upon the criteria supplied to the tool.

A list of candidates is supplied with the **LS** option. The **US** option actually runs the UPDATE STATISTICS utility (SQLDBSU) and maintains the history in the SQLMAINT_TABLE. By using the Job Scheduling tool, the Statistics Maintenance tool can be scheduled to run on a regular basis to cycle through the list of DBSPACES on some determined schedule. Optionally, the job can be scheduled to run on selected DBSPACES after analyzing the report.

List Statistics (LS)

This function is chosen by specifying the **LS** option on the SQLMAINT entry panel. The purpose of this function is to create a list of DBSPACES, based upon the input parameters specified, which need to have UPDATE STATISTICS run.

The output of this run is file "database USTAT*nnn* A", where *nnn* depends on the DBSPACE OWNER value entered. If ALL was chosen, then *nnn*=ALL. If PUBLIC was entered, then *nnn*=PUB. If PRIVATE was entered, then *nnn*=PRI. If another value was entered, then *nnn* equals the first three characters of the value. This file is in the format of an SQLDBSU input file with commands to update statistics for the DBSPACES chosen by SQLMAINT. The file can therefore be used as input to a stand-alone DBSU run if desired. Refer to the *DB2 Server for VSE & VM Database Services Utility* manual for more information on this utility.

Update Statistics (US)

This function is chosen by specifying the **US** option on the SQLMAINT entry panel. The purpose of this function is to run the UPDATE STATISTICS on the DBSPACES chosen according to the parameters supplied.

The output of this run is file "database USTAT*nnn* A" (see LS function above), which is the input to the SQLDBSU to perform the UPDATE STATISTICS on the selected DBSPACES. Another file, SQMINSRT USTATLOG A, will contain the results of the run. The function is either performed at your terminal **(PF10)** or can be scheduled for later execution **(PF6)**.

Examples of the two files are shown in Figure 206 on page 436.

```
COMMENT '************************************************************'
COMMENT '******** Update Statistics for DBSPACEs on SQLDBA  *********'
COMMENT '********** OPTIONS: DBSPACEs: PRIVATE.ALL, min=0, **********'
COMMENT '**********     max=4097, days=14, qty=5, time=30 ***********'
COMMENT '************************************************************'
UPDATE ALL STATISTICS FOR DBSPACE "MSTRUSR1"."DBSPNAM1";
COMMIT WORK;
UPDATE SQLMAINT.SQLMAINT_TABLE SET UPSTAT_DATE = '97110',
UPSTAT_TIME = '10:47:25'
     WHERE OWNER = 'MSTRUSR1' AND DBSPACENAME = 'DBSPNAM1';
COMMIT WORK;
COMMENT '******************************************'
UPDATE ALL STATISTICS FOR DBSPACE "MSTRUSR2"."DBSPNAM2";
COMMIT WORK;
UPDATE SQLMAINT.SQLMAINT_TABLE SET UPSTAT_DATE = '97110',
UPSTAT_TIME = '10:47:25'
     WHERE OWNER = 'MSTRUSR2' AND DBSPACENAME = 'DBSPNAM2';
COMMIT WORK;
COMMENT '******************************************'
UPDATE ALL STATISTICS FOR DBSPACE "MSTRUSR3"."DBSPNAM3";
COMMIT WORK;
UPDATE SQLMAINT.SQLMAINT_TABLE SET UPSTAT_DATE = '97110',
UPSTAT_TIME = '10:47:25'
     WHERE OWNER = 'MSTRUSR3' AND DBSPACENAME = 'DBSPNAM3';
COMMIT WORK;
COMMENT '******************************************'
UPDATE ALL STATISTICS FOR DBSPACE "MSTRUSR4"."DBSPNAM4";
COMMIT WORK;
UPDATE SQLMAINT.SQLMAINT_TABLE SET UPSTAT_DATE = '97110',
UPSTAT_TIME = '10:47:25'
     WHERE OWNER = 'MSTRUSR4' AND DBSPACENAME = 'DBSPNAM4';
COMMIT WORK;
COMMENT '******************************************'
UPDATE ALL STATISTICS FOR DBSPACE "MSTRUSR5"."DBSPNAM5";
COMMIT WORK;
UPDATE SQLMAINT.SQLMAINT_TABLE SET UPSTAT_DATE = '97110',
UPSTAT_TIME = '10:47:25'
     WHERE OWNER = 'MSTRUSR5' AND DBSPACENAME = 'DBSPNAM5';
COMMIT WORK;
COMMENT '******************************************'
COMMENT '******************************************'
COMMENT '*** DBSPACES needing update = 10 ****'
COMMENT '******************************************'
COMMENT '*** DBSPACES to update this run = 5 ****'
COMMENT '******************************************'
```

Figure 206. Update Statistics DBSU File

In the preceding figure, Figure 206, ten DBSPACES were eligible for maintenance; however, only five were done since a QTY LIMIT of five was specified.

```
                Update Statistics for DBSPACEs on SQLDBA
                           03/16/97 10:47:29

                     DBSPACEs: PRIVATE.ALL, min=0,
                    max=4097, days=14, qty=5, time=30

10:47:52 MSTRUSR1.DBSPNAM1 UPDATE STATISTICS completed successfully
10:48:16 MSTRUSR2.DBSPNAM2 UPDATE STATISTICS completed successfully
10:48:18 MSTRUSR3.DBSPNAM3 UPDATE STATISTICS completed successfully
10:48:36 MSTRUSR4.DBSPNAM4 UPDATE STATISTICS completed successfully
10:48:38 MSTRUSR5.DBSPNAM5 UPDATE STATISTICS completed successfully
10:48:38 SQLMAINT UPDATE STATISTICS function completed
```

Figure 207. Update Statistics Report File

Note that the panel input values are listed in the report. Here, all private DBSPACES were to be considered, but only five were chosen. A time limit of 30 minutes was allocated to the job, and only DBSPACES up to 4097 pages were considered. Only those DBSPACES which had had statistics updated at least 14 days ago were considered.

DBSPACE Reorganization Maintenance Tool

The DBSPACE Reorganization Maintenance tool consists of the List Reorg Candidates function **(LR)**, Run Reorg function **(RR)** and the Run Reorg Prior **(RP)** function.

How the DBSPACE Reorganization Maintenance Tool Works

The main objective of this tool is reorganization candidate selection. The selection function is a two step process. First, a list of potential DBSPACES is selected from the SQLMAINT table based upon the options entered on the Automated DBSPACE Maintenance entry panel. Second, this list is checked against the selection criteria in effect that determine whether a particular DBSPACE needs reorganizing. The DBSPACE selection options have been covered above. The selection criteria are described next.

DBSPACE Maintenance Criteria Selection Screen

Shown in Figure 208 is the Criteria Selection screen for the Automated DBSPACE Maintenance tool.

```
**********************   DBSPACE MAINTENANCE UTILITY    ************** *********

 DATABASE => SQMDBA          CRITERIA  SELECTION  SCREEN

 -------------- CRITERIA DERIVED FROM SYSTEM CATALOG: (QUICK RETRIEVAL) -------

 CRITERIA 1: Y UNCLUSTERED INDEX   OR    CLUSTER RATIO LESS THAN
             -                         -                          ----

 CRITERIA 2: Y OVERFLOW ROWS >  10 %        (Specify Y or N)
             -                  --

 -------------- CRITERIA DERIVED FROM SHOW DBSPACE: (LONGER RETRIEVAL) --------
 Y USE CRITERIA 3 thru 6 (Y or N)
 -
 CRITERIA 3: DATA or INDEX PAGES > 70 % USED
                                   --
 CRITERIA 4: PROPORTION of USED DATA/INDEX PAGES
 ipg used > 40 % AND dpg used < 40 %  OR dpg used > 40 % AND ipg used < 40 %
           --                 --                --                --
 CRITERIA 5: DATA PAGES FREE SPACE > 35 % AND DATA PAGES USED > 25 %
                                    --                      --
 CRITERIA 6: EMPTY DATA or INDEX PAGES  >  5 % of TOTAL DATA or INDEX PAGES
                                   --
 PF1 HELP PF3 QUIT PF6 SET TO DEFAULTS PF7 RET TO PRIOR SCREEN  PF10 PROCESS
```

Figure 208. Automated DBSPACE Maintenance Selection Criteria Screen

The Automated DBSPACE Maintenance tool provides for user selected maintenance criteria parameters. With this panel, you can choose which criteria to use, as well as adjust any of the values of a specific criterion.

There are currently six criteria which are evaluated by SQLMAINT, any one of which can cause a DBSPACE to be selected for reorganization:

1. The existence of unclustered primary indexes on tables within the DBSPACE or the CLUSTERRATIO field value reaching a designated value for any index within the DBSPACE. This will typically occur when many inserts and deletes are performed over time, and it will greatly reduce the performance of the database.
2. The existence of over 10% of overflow pages for any table within the DBSPACE. This condition indicates a great deal of fragmentation caused by inserts and will considerably increase the number of I/O operations required to return selected rows.
3. The condition of over 70% of available data pages or index pages being occupied. The reorganization process can better pack data on each occupied page and reduce the number of DBSPACE pages required.
4. An imbalance between the fullness of the data pages and index pages. Each DBSPACE reserves a certain percentage of its allocated pages for indexes which can or can not provide the proper proportion of data and index pages after actual DBSPACE use. SQLMAINT will detect any imbalance and will calculate the correct proportion to be used during the reorganization process.
5. The existence of a lot of free space on each DBSPACE page and a large number of pages used. The DBSPACE will normally reserve a certain percentage of each data page for later insert activity, allowing the data to remain clustered. If the number of pages increases considerably and the reserved space within each page is not being used for inserts, then SQLMAINT will request a reorganization which will reduce the percentage of each page that is reserved for inserts.
6. The existence of empty data or index pages in excess of 5% of total DBSPACE data pages or index pages.

The Search Criteria are divided into 2 groups:

A. Those obtained from the database System Catalogs
B. Those obtained with the SHOW DBSPACE command

Group A criteria afford a quicker determination of a DBSPACE's candidacy and result in a faster SQLMAINT execution. Group B criteria use the SHOW DBSPACE command for each selected DBSPACE to obtain relevant data against which to match the criteria values to determine DBSPACE candidacy.

Each Group's criteria are discussed below. Note that any criterion can be used in conjunction with any other criterion. Criterion **1** offers a choice between 2 index type tests. If no criteria are modified or specified directly by the DBA, the defaults will be used. The defaults are set originally by the menu. Subsequent modification to any values and choices will be retained the next time SQLMAINT is invoked. To reset to the default values, bring up the CRITERIA SELECTION panel and press PF6 to restore to default values.

```
        ***************  GROUP A CRITERIA    ***************

CRITERION 1:  UNCLUSTERED INDEX    or    CLUSTER RATIO LESS THAN nnnn

              This criterion affords a choice of either using
unclustered index occurrences within the DBSPACE OR specifying a
CLUSTERRATIO value (0001-9999) to determine candidates for reorganization.
The 2 choices are mutually exclusive - one or the other can be specified.
Either can be used in conjunction with other criteria.
              CLUSTER RATIO LESS THAN nnnn is the value which the
CLUSTERRATIO field in the catalog must be below in order to have the
DBSPACE be picked as a candidate for reorganization regardless of the
value of the CLUSTER field.  If this field is blank, then the UNCLUSTERED
INDEX criterion is used and a weight of 1 is assigned if a table has an
index which is W (no longer clustered).  If the CLUSTER RATIO field has
a value entered, a weight of 1 is assigned if any table's CLUSTERRATIO
field value is below the value specified in this parameter, regardless
of CLUSTER field value.

              The DEFAULT criterion is UNCLUSTERED INDEX.

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

CRITERION 2:  OVERFLOW ROWS > nn %  (10% is default)

              Specifying this criterion will cause SQLMAINT to check
the number of rows in overflow and determine whether the percentage
specified in the selection criterion is equal to or greater than the
actual percentage calculated from the catalog.

              The DEFAULT is 10%.

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

        ***************  GROUP B CRITERIA    ***************

  Specifying a Y in the USE CRITERIA 3 through 6 field will cause
SQLMAINT to employ all GROUP B criteria when determining candidate
selection.  Specifying an N will cause SQLMAINT to use only those
criteria which have values specified in their parameter fields.

              The DEFAULT is Y (use all Group B criteria)


CRITERION 3:  DATA or INDEX PAGES > nn % USED   (70% is default)

              This criterion checks the percentage of currently occupied
data and index pages.  If the percentage equals or exceeds the value
specified, the DBSPACE is considered a candidate for reorg.

              The DEFAULT is 70%.
```

Figure 209 (Part 1 of 2). SQLMAINT DBSPACE Selection Criteria

```
------------------------------------------------------------------------

CRITERION 4:  PROPORTION of USED DATA/INDEX PAGES   nn % (40% is default)

              Specifying a value for this criterion will cause
SQLMAINT to do these checks to determine candidacy:

   a. If index pages used is GREATER THAN nn %      AND
          data pages used is LESS    THAN nn %
   b. If data  pages used is GREATER THAN nn %       AND
         index pages used is LESS    THAN nn %

   A match on either of the above results in candidacy.

              The DEFAULT is 40%.

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

CRITERION 5:  DATA PAGES FREE SPACE > nn %  AND  (nn default is 35%)
              DATA PAGES USED       > xx %       (xx default is 25%)

              This criterion checks the percentage of data pages free
space for greater than nn%.  If met or exceeded, it checks the percentage
specified (xx %) of data pages used.  If both match or exceed the values
specified, the DBSPACE is considered a candidate.

              The DEFAULT is 35% for nn, and 25% for xx.

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

CRITERION 6:  EMPTY DATA or INDEX PAGES > nn % of TOTAL DATA or INDEX
              PAGES.

              This criterion checks the percentage of empty data or
index pages for nn %.  If met or exceeded, the DBSPACE is considered a
candidate.

              The DEFAULT is 5% for nn.
```

Figure 209 (Part 2 of 2). SQLMAINT DBSPACE Selection Criteria

The PF Keys offer these options:

PF Key	Description
Help Screen (PF1)	Displays the HELP SCREEN which explains the selection criteria.
Quit (PF3)	Is the QUIT or EXIT key which ends SQLMAINT processing.
Set to Defaults (PF6)	Resets the criteria's default values.
Ret to Prior Screen (PF7)	Returns you to the main SQLMAINT entry panel.
Process (PF10)	This key processes any changes made to the panel criteria selections and values and begins the process of DBSPACE candidate selection based upon the main panel choices and the criteria selected.

SQLMAINT TABLE List Screen

Shown in Figure 210 is the SQLMAINT Table list screen for the Automated DBSPACE Maintenance tool.

```
 mm/dd/yyyy                    CONTROL CENTER                        hh:mm:ss
*----------------------  DBSPACE MAINTENANCE UTILITY   ----------------------*
|  Database => SQMDBA                                                        |
|                                 UPSTATS      REORGANIZATION                |
| OWNER     NAME                LAST   ELAP  LAST   ELAP  STAT  WGHT  NPAGES |
| --------- ------------------  -----  ----  -----  ----  ----  ----  ------- |
| C370451   C370451             97073     0  97073     1    00     1     1024 |
| M356959   M356959A            97094     0  97094     2    00     1     1024 |
| M760595   M760595B            00000     0  00000     0           1     1024 |
| M997990   M997990A            97074     0  97074     0    00     1     1024 |
| PUBLIC    DBEDB00             97016     1  00000     0           2     2048 |
| PUBLIC    DSQTSCT1            97035     0  97035    11           1     4097 |
| PUBLIC    DSQTSCT2            95334     0  95336     7           1     4097 |
| PUBLIC    DSQTSCT3            97041     0  97037    27           1     8064 |
| PUBLIC    DSQTSDEF            97071     0  00000     0           1     4097 |
|                                                                            |
|                      The fields in RED are modifiable                      |
|                                                                            |
|                                                                            |
|                              Page 1     of 3                               |
*----------------------------------------------------------------SQMMAIN3----*
PF:  1 Help        3 EXIT          4 Maint Panel    7 Bkwd         8 Fwd
     9 Sort/Wght  10 Sort/Udate   11 Sort/Rdate  12 Sort/Rstat   ENTER UPDATE
```

Figure 210. SQLMAINT Table List Screen

Pressing **PF9** from the main SQLMAINT entry panel will provide you with the rows from the SQLMAINT_TABLE which are retrieved according to the options present on the screen. For the LS and US options, the rows which meet the UPSTAT_DATE parameter (DAYS SINCE) will be presented (the rows must also meet the other parameters selected, such as MIN-MAX pages). For the **LR, RR,** and **RP** options, the rows which meet the REORG_DATE parameter (DAYS SINCE) will be presented.

From the screen which is presented, you can sort the report by various fields. You can also update certain fields which would then subsequently alter the candidate list chosen when you finally begin the candidate selection process.

For a full explanation of the new SQLMAINT Table Display screen, enter HELP SQLMNT3 at a CMS command prompt. This is the same information which is presented when you press **PF1** from the Table Display List screen.

List Reorg Candidates (LR)

This function is chosen by specifying the **LR** option on the SQLMAINT entry panel. The purpose of this function is to create a list of DBSPACES, based upon the input parameters specified, which need to be reorganized.

The output of this run is three files:

database REOPICKS A — Contains a list of DBSPACES picked for reorganization. It is used internally by SQLMAINT.

database SQLMAINT A	Contains a list of DBSPACES picked for reorganization. It is used internally by SQLMAINT when running the Reorg using Prior List function.
database *nnn*REOLS A	The *nnn* is dependent upon the DBSPACE type picked (see “List Statistics (LS)” above). This file is the actual report generated showing the status of the candidate DBSPACES which were examined for reorganization potential.

Shown in Figure 211 on page 443 is an example of the reorganization analysis report created by SQLMAINT. This report is saved in a file on the user's 191 A-disk. The filename of the report will be the database name and the filetype will be descriptive of the subset of the database analyzed (the filetype of the example is PRIREOLS, indicating that private DBSPACES were considered for analysis).

In this report, nine private DBSPACES were analyzed and three were selected for reorganization, while one was discounted since it only met criterion 5 (space USER2.DBSPACE2). Note that the others had never been reorganized, since their date/time fields were 0. They happened to be new entries in the SQLMAINT_TABLE. The report was sequenced by DBSPACE name since this was only a report run.

Based on the analysis, DBSPACE MSTRUSR1.DBSPNAM1 will be reorganized first because it has a weight of 2. Weight is calculated by adding the number of unclustered indexes to the number of other reorganization criteria which caused this DBSPACE to be chosen.

The next reorganization will be performed on the DBSPACE with the next highest WEIGHT, and so on.

```
                DBSPACE REORGANIZATION CANDIDATES REPORT
                        Database: VMSYSTM1.SQLDBA          03/16/97 13:57:00

            OPTIONS: DBSPACEs: PRIVATE.ALL, seq=DBSPACE, min=0,
                    max=999999, days=14, qty=10, time=180

  DBSPACES EXAMINED = 9
  ************************************************************
  CRITERIA 1 = DBSPACES with Tables with index type W

  CRITERIA 2 = DBSPACES with Tables with NOVERFLOW rows >= 10%

  CRITERIA 3 = DBSPACES with data or index pages > 70% used

  CRITERIA 4 = DBSPACES with index pages used > 40% and data pages < 40%
               -- OR -- with data pages used > 40% and index pages < 40%

  CRITERIA 5 = DBSPACES with data pages free space > 35%
               -- AND --  data pages used > 25%

  CRITERIA 6 = DBSPACES with empty data or index pages > 5%
                        of DBSPACE data pages or index pages
*******************************************************************************
MSTRUSR1.DBSPNAM1              DBSPACENO: 59       LAST REORG: 00000 00:00:00
       Criteria 1:  2 indexes unclustered
                           NBR PAGES  OCCUPIED PAGES   %FREE  EMPTY
                    HEADER:      8      1 (    12%)     81%
                    DATA:      679    190 (    27%)     22%    0
                    INDEX:     337     24 (     7%)     48%    2
            WEIGHT = 2,    DBSPACE PAGES = 1024
*******************************************************************************
MSTRUSR2.DBSPNAM2              DBSPACENO: 71       LAST REORG: 00000 00:00:00
       Criteria 4:  PCTINDX: Old=20, Suggested=20
                           NBR PAGES  OCCUPIED PAGES   %FREE  EMPTY
                    HEADER:      8      1 (    12%)     83%
                    DATA:      812    437 (    53%)     12%    0
                    INDEX:     204      0 (     0%)      0%    0
            WEIGHT = 1,    DBSPACE PAGES = 1024
   This DBSPACE will NOT be REORGED, the evaluation indicates
   that a reorganization is not currently needed.
*******************************************************************************
MSTRUSR3.DBSPNAM3              DBSPACENO: 179      LAST REORG: 00000 00:00:00
       Criteria 1:  1 index unclustered
                           NBR PAGES  OCCUPIED PAGES   %FREE  EMPTY
                    HEADER:      8      1 (    12%)     97%
                    DATA:     1733      3 (     0%)     13%    0
                    INDEX:     307      2 (     0%)     67%    0
            WEIGHT = 1,    DBSPACE PAGES = 2048
*******************************************************************************
MSTRUSR4.DBSPNAM4              DBSPACENO: 180      LAST REORG: 00000 00:00:00
       Criteria 1:  1 index unclustered
                           NBR PAGES  OCCUPIED PAGES   %FREE  EMPTY
                    HEADER:      8      1 (    12%)     97%
                    DATA:     1733      3 (     0%)     13%    0
                    INDEX:     307      2 (     0%)     67%    0
            WEIGHT = 1,    DBSPACE PAGES = 2048
```

Figure 211. SQLMAINT List Reorg Candidates Report

Run Reorg Prior Function (RP)

This function is chosen by specifying the **RP** option on the SQLMAINT entry panel. The purpose of this function is to run reorganizations using a previously created list of reorganization candidates in a file named dbname SQLMAINT A).

This option is identical to the **RR** option, except that DBSPACES are not evaluated; the file containing previously evaluated DBSPACES is used as input to DBSPACE selection.

The output of this run is two files:

database RUNREOLS A	This file is a report of the reorganized DBSPACES.
database SQLMAINT A	This file is edited by removing the entries which were reorganized in this run. When all entries are deleted, the file is also erased.

Shown in Figure 212 is an example of the reorganization execution report generated by SQLMAINT. This report is saved in a file on the user's 191 A-disk. The filename of the report will be the database name and the filetype will be RUNREOLS.

The reorganization report shows the before and after statistics for the DBSPACE and the results of the reorganization (provided as a status code defined at the end of the report).

```
                        DBSPACE REORGANIZATION REPORT
                           Database: VMSYSTM1.SQLDBA            03/16/97 15:10:01

                                ELAPSE
OWNER    DBSPACENAME        SPNO TIME   ST W DATA STATISTICS      INDEX STATISTICS
________ __________________ ____ ______ __ _ _________________    _________________

MSTRUSR1 DBSPNAM1             59 2.38   00 2 679 190(27%) 22%     337 24(7%) 48%
                        ****** After REORG: 679 190(27%) 22%     337 24(7%) 48%

MSTRUSR3 DBSPNAM3            179 1.66   00 1 1733 3(0%) 13%       307 2(0%) 67%
                        ****** After REORG: 1733 3(0%) 13%       307 2(0%) 67%



Number of DBSPACES successfully reorged =  2

Total elapsed time for all reorgs = 4.04 minutes


                     End of Report for  PDEVDBA  REORG JOB

NOTE : SPNO  - DBSPACE number
       ELAPSE- elapsed time, in minutes, for the REORG
       W     - weight factor of the DBSPACE reorged
       ST    - status of the REORG
               '00' Good Reorg
               'XX' Good Reorg but space should be increased
               'CC' Good Reorg but selection criteria did not change
               'NN' Reorg not done because space should be increased
               'SS' Not enough unload disk space for reorg
               'TT' Not enough time left to reorg this space
               '99' Bad Reorg
       DATA/INDEX BEFORE - the results of a SHOW DBSPACE
                command before the REORG was done, format:
                    tttt uu(pp%) ff%
                where:  tttt is total pages,
                        uu   is pages used,
                        pp   is percent of pages used,
                        ff   is avg. percent free per page.
       DATA/INDEX AFTER  - the results of a SHOW DBSPACE
                command after the REORG was done
```

Figure 212. Example SQLMAINT DBSPACE Reorganized Report

Run Reorg Function (RR)

This function is chosen by specifying the **RR** option on the SQLMAINT entry panel. The purpose of this function is to evaluate the DBSPACES to create a list of reorganization candidates, then to run reorganization on them according to the specified input parameters.

The output of this run is three files:

database REOPICKS A	This file is a list of DBSPACES picked for a reorganization. It is used internally by SQLMAINT.
database *nnn*REOLS A	The *nnn* is dependent upon the DBSPACE type picked (see "List Statistics (LS)" above). This file is the actual report generated showing the status of the candidate DBSPACES which were examined for reorganization potential. It is identical to the report produced by the RS function, except that SEQ will either be WEIGHT or AGE, depending on the option chosen on the entry panel. See Figure 211 on page 443.
database RUNREOLS A	This file is a report of the reorganized DBSPACES. It is identical to the report generated by the RP function. See Figure 212 on page 444.

Additional Topics

Automated Execution

SQLMAINT has been designed for repeated execution on a scheduled basis using information in the SQLMAINT table that is created in each database. Since this table will initially have no history of maintenance on each DBSPACE, all DBSPACES will be considered when SQLMAINT is executed the first time. For the List Reorg Candidates (LR) and Run Reorgs (RR) functions, this can take a considerable amount of time (each DBSPACE will be scanned). To reduce the number of DBSPACES scanned, update the SQLMAINT table manually.

Examples

To update statistics on the SQLDBA database for a maximum of 50 private DBSPACES per night, with a maximum run time of one hour, and selecting DBSPACES that have not had maintenance for three weeks, this statement could be executed each night from a disconnected virtual machine:

```
SQLMAINT SQLDBA UPSTATS ( PRIVATE QTY=50 TIME=60 DAYS=21
```

To reorganize DBSPACES on the TESTDBA database for a maximum of 10 private DBSPACES per night, with a maximum run time of three hours, selecting DBSPACES in sequence according to the most need, and only considering DBSPACES that have not been reorganized within the last month, this statement could be executed each night from a disconnected virtual machine:

```
SQLMAINT TESTDBA RUNREORG ( PRIVATE WEIGHT QTY=10 TIME=180 DAYS=30
```

Reorganization Work Space

The reorganization process will require disk space to hold the unloaded DBSPACE data until each DBSPACE is dropped and reacquired. It is your responsibility to assure that enough disk space is available to hold all DBSPACES that will be reorganized during a single SQLMAINT execution. If the default temporary disk option is selected, each DBSPACE reorganization will acquire a temporary disk to hold the unloaded data and will drop the temporary disk when the reorganization has completed successfully. This option uses the 191 A-disk to hold the DBSU command file and the DBSU output listing file. These files can be large, so enough 191 disk space should be provided.

When a linked disk is specified for SQLMAINT use (with the FILEMODE entry), the disk will be used for all reorganization activity. This includes the unloaded DBSPACE, the DBSU command file, and the DBSU output listing file. These files will not be erased after the reorganization completes, so this disk must be large enough to hold files for all DBSPACE reorganizations that will be performed during a single execution of SQLMAINT.

Using the SQLMAINT Control Table

The SQLMAINT Control Table can be altered by the Control Center Administrator to control the selection process of the SQLMAINT tool.

If a database has a large number of DBSPACES, the REORG_DATE and UPSTAT_DATE columns in the SQLMAINT table can initially be set to different values so that a single SQLMAINT run will only scan a portion of the DBSPACES. For instance, you can execute a database command such as:

```
UPDATE SQLMAINT.SQLMAINT_TABLE SET UPSTAT_DATE = '99001', REORG_DATE
= '99001' WHERE OWNER < 'M'
```

This can change about half of the DBSPACES to a different maintenance date. You could then invoke SQLMAINT with an appropriate DAYS SINCE parameter so that only those DBSPACES would be selected. By using the TIME LIMIT and/or QTY LIMIT parameters, you could further restrict the number of DBSPACES that would be maintained until the SQLMAINT table developed a maintenance history.

The REORG_STATUS parameter can also be changed to NN to prevent a DBSPACE from being selected for reorganization.

Summary Comments

- Many flexible execution options provide you with a high degree of control over the maintenance of the database. Limits can be placed on the maintenance activities, such as a maximum quantity of DBSPACES to reorganize or the maximum elapsed time for the maintenance activity.
- The analysis and subsequent maintenance can also be restricted to a defined subset of the database during each execution by using one or more of the available selection options.

- The UPDATE STATISTICS function can be automated for an entire database by scheduling SQLMAINT to execute on a regular basis. SQLMAINT maintains a history of UPDATE STATISTICS maintenance performed for each DBSPACE, allowing automatic rotation through the database. Each DBSPACE will therefore obtain maintenance regularly as SQLMAINT selects candidates based on length of time since UPDATE STATISTICS was performed.
- SQLMAINT will examine DBSPACE characteristics within the database to select candidates for reorganization. A weight is assigned to each DBSPACE based on how seriously the DBSPACE requires reorganization. For example, a DBSPACE with 10 unclustered indexes requires reorganization more than a DBSPACE with only 1 unclustered index.
- The maintenance can also be restricted to a defined subset of the database during each execution by using one or more of the available selection options. A single execution can be specified to consider only public or private DBSPACES. You can also specify that only DBSPACES with a specific owner or DBSPACENAME should be considered. The OWNER and DBSPACENAME values can also be specified with the database wildcard selection characters to specify a subset of DBSPACES to be analyzed. You can also limit the selected DBSPACES to those within certain specified size limits (minimum and maximum). You can also exclude DBSPACES from the evaluation if maintenance had previously been performed on them within a specified number of days.
- The sequence in which reorganizations are performed on the selected DBSPACES can be ordered by criticality of need or by the elapsed time since reorganization was last performed on the DBSPACES. Reorganizing based on criticality of need will assure that the DBSPACES that need reorganization the most will be reorganized first. Reorganizing based on elapsed time since the previous reorganization will assure that DBSPACES are always reorganized within a reasonable time period.
- The SQLMAINT panel provides a single screen which allows you to enter selectable and optional parameters when invoking the DBSPACE maintenance function.
- Processing options for SQLMAINT include many parameters which will limit the DBSPACES within the database that will be considered for reorganization or statistics updating. This is important for large databases, since the evaluation process itself can be time-consuming if all DBSPACES in the database are examined.
- The DBSPACE OWNER parameter can be used to limit the DBSPACES considered for maintenance to PUBLIC only, PRIVATE only, or a specific DBSPACE OWNER. The % (percent sign) can also be used as a leading or trailing character in the value provided for DBSPACE OWNER as a wildcard indicator, allowing the selection of all DBSPACES which begin or end with specified characters.
- The DBSPACE NAME parameter can also be specified to restrict the SQLMAINT operation to a single DBSPACE. The % (percent sign) can also be used in this parameter to select a range of DBSPACE names.
- The SEQUENCE in which the candidate DBSPACES are processed can be specified by WEIGHT (those DBSPACES which are most in need will be processed first) or AGE (those DBSPACES which have the longest elapsed time since the process was performed will be processed first).

- Processing limits can be placed on the SQLMAINT execution by specifying a TIME LIMIT (in minutes) or a QUANTITY LIMIT (the maximum number of DBSPACES to process), or both limits can be used. If SQLMAINT does not process all DBSPACES due to a defined processing limit, the remaining DBSPACES will automatically be reconsidered again when SQLMAINT next executes.
- The STOP ON FAILURE option allows you to specify whether SQLMAINT should continue with maintenance on subsequent DBSPACES if a failure occurs on any previous DBSPACE. A value of YES would cause SQLMAINT to immediately terminate whenever maintenance on a specific DBSPACE fails. A value of NO would allow SQLMAINT to continue with maintenance on subsequent DBSPACES when maintenance on a previous DBSPACE fails.

Chapter 35. Rebind Package Tool

Overview

Using the Rebind Package tool, you can selectively rebind packages within the database. Processing may be initiated for immediate execution (PF key 10) or scheduled execution (PF key 6) using the Job Scheduling tool.

The Rebind Package tool uses the Database Services (DBS) Utility command REBIND PACKAGE described in the *DB2 Server for VSE & VM Database Services Utility* manual.

Package Rebind Utility Screen

Below is an example of the Package Rebind Utility Screen.

```
******* SQLRBIND V6.1 **** PACKAGE REBIND UTILITY   *************

DATABASE ==>  SQLDBA
              --------


    ----------------------------  Options  ------------------------------

PACKAGE CREATOR ==> ALL                ( ALL, creator, use % for wildca
                    ------------------
PACKAGE NAME    ==>                    ( blank for ALL, use % for wildc
                    ------------------
INVALID         ==> YES ( YES or NO, rebind only if INVALID )
                    ---
PAUSE           ==> NO  ( YES or NO, pause after DBSU creation )
                    ---




PF:  1 HELP   3 QUIT   4 EXIT   5 Main Menu   6 SCHEDULE  10 PROCESS NOW
```

Figure 213. Rebind Package Tool Entry Panel

Entry Fields

Database

The name of the target database where maintenance will be performed. Enter either the machine name or dbname.

Package Creator

Only packages having a specified PACKAGE CREATOR will be considered for rebinding. You may use a percent sign (%) at the beginning or end of your entry to represent zero or more characters. This allows you to select packages when only knowing a portion of the value or to select packages having similar values. Enter 'ALL' to rebind packages regardless of PACKAGE CREATOR.

Package Name

Only packages having a specified PACKAGE NAME will be considered for rebinding. You may use a percent sign (%) at the beginning or end of your entry to represent zero or more characters, allowing you to select packages when only knowing a portion of the name or to select packages having similar names. Enter 'ALL' (or leave blank) to rebind packages regardless of PACKAGE NAME.

Invalid

Enter 'YES' to rebind only those packages that are currently invalid. Enter 'NO' to rebind packages regardless of whether they are currently invalid.

Pause

Enter 'YES' to have the Rebind Package tool generate and capture all REBIND PACKAGE commands as specified without executing. The output is saved in Database Services Utility (DBSU) command file '*creator* SQLDBSU A' (see Figure 214) where *creator* equals the PACKAGE CREATOR entry without the optional leading and trailing percent signs (%). You may execute the DBSU file at a later date and time using the DBS Utility external to the product.

Note: If file '*creator* SQLDBSU A' already exists, then a backup copy is made before the new file is created. The name of the backup file is generated by taking the original name, switching the filename and filetype, then reversing the order of the characters. For example, if the original file is named ALL SQLDBSU A, then the backup file will be named USBDLQS LLA A. If the backup file already exists, it will be overwritten without warning.

```
SET AUTOCOMMIT (ON)
SET ERRORMODE (CONTINUE)
REBIND PACKAGE ("SQLDBA1"."ARIDSQL");
REBIND PACKAGE ("SQLDBA1"."ARIFCRD");
REBIND PACKAGE ("SQLDBA1"."ARIISQL");
REBIND PACKAGE ("SQLDBA2"."ETCATPHD");
REBIND PACKAGE ("SQLDBA3"."ETCATPHI");
REBIND PACKAGE ("SQLDBA4"."ETCATPHM");
```

Figure 214. Example SQLDBSU File Generated by Rebind Package Tool

Sample Output

```
ARI0801I DBS Utility started: 05/13/97 15:37:21.
         AUTOCOMMIT = OFF ERRORMODE = OFF
         ISOLATION LEVEL = REPEATABLE READ
------> SET AUTOCOMMIT (ON)
ARI0815I ...AUTOCOMMIT Processing Mode = ON
ARI8997I ...Begin COMMIT processing.
ARI0811I ...COMMIT of any database changes successful.
------> SET ERRORMODE (CONTINUE)
ARI0827I ...Begin command execution: ERRORMODE = CONTINUE
------> REBIND PACKAGE ("SQLDBA1"."ARIDSQL");
ARI0852I REBIND PACKAGE processing started.
ARI0855I REBIND PACKAGE processing successful.
ARI8997I ...Begin COMMIT processing.
ARI0811I ...COMMIT of any database changes successful.
------> REBIND PACKAGE ("SQLDBA1"."ARIFCRD");
ARI0852I REBIND PACKAGE processing started.
ARI0855I REBIND PACKAGE processing successful.
ARI8997I ...Begin COMMIT processing.
ARI0811I ...COMMIT of any database changes successful.
------> REBIND PACKAGE ("SQLDBA1"."ARIISQL");
ARI0852I REBIND PACKAGE processing started.
ARI0855I REBIND PACKAGE processing successful.
ARI8997I ...Begin COMMIT processing.
ARI0811I ...COMMIT of any database changes successful.
------> REBIND PACKAGE ("SQLDBA2"."ETCATPHD");
ARI0852I REBIND PACKAGE processing started.
ARI0855I REBIND PACKAGE processing successful.
ARI8997I ...Begin COMMIT processing.
ARI0811I ...COMMIT of any database changes successful.
------> REBIND PACKAGE ("SQLDBA3"."ETCATPHI");
ARI0852I REBIND PACKAGE processing started.
ARI0855I REBIND PACKAGE processing successful.
ARI8997I ...Begin COMMIT processing.
ARI0811I ...COMMIT of any database changes successful.
------> REBIND PACKAGE ("SQLDBA3"."ETCATPHM");
ARI0852I REBIND PACKAGE processing started.
ARI0855I REBIND PACKAGE processing successful.
ARI8997I ...Begin COMMIT processing.
ARI0811I ...COMMIT of any database changes successful.
ARI0802I End of command file input.
ARI0809I ...No errors occurred during command processing.
ARI0808I DBS processing completed: 05/13/97 15:37:26.
```

Figure 215. Example DBSU Message File Output Generated During Processing

Chapter 36. Object Search and List Tools

Overview

The Object Search and List tools are a set of integrated applications that search the system catalog tables in your database and display information about these objects:

1. DBSPACES
2. Tables
3. Packages
4. Indexes
5. Views
6. Columns

For every object type except columns, you can "drill-down" to a lower level of information. For example, you might ask for a list of public DBSPACES whose names contain the string "ORDER". To do this, you would make the entries shown in Figure 216 and press ENTER.

```
 mm/dd/yyyy                    CONTROL CENTER                        hh:mm:ss
*------------------------- DBSPACE List Utility -----------------------------*
|                                                              CTRLID: MSTRSRV1 |
|  Database => SQLDBA                                            NODE: VMSYSTM1 |
|                                                                               |
|  DBSPACE OWNER ==> ALL                 (ALL, PUBLIC, PRIVATE, ownername)      |
|                                                                               |
|  DBSPACE NAME  ==> %ORDER%             (blank for ALL, use % for wildcard)    |
|                                                                               |
|  DBSPACE PAGES:  MINIMUM ==> 0        MAXIMUM     ==> 9999999                 |
|                                                                               |
|  DBSPACE TYPE ==> A                    (A=Acquired, U=Unacquired, B=Both)     |
|                                                                               |
|  STORPOOL     ==> _____                (blank for ALL)                        |
|                                                                               |
|       The % SQL wildcard character can be used when specifying the            |
|       Dbspace Owner or Dbspace Name values for selection.                     |
|                                                                               |
|                                                                               |
|                                                                               |
*-------------------------------------------------------------SQMDLIST-----*
  PF1  HELP      PF3 QUIT    PF4 EXIT    PF5 MAIN MENU     ENTER Process
  PF10 TABLE Search panel    PF11 PACKAGE Search panel
```

Figure 216. DBSPACE Object Search and List screen

The Object Search and List tools search the database system catalog tables using the parameters you supply and displays the output as shown in Figure 217 on page 454.

```
 mm/dd/yyyy                    CONTROL CENTER                        hh:mm:ss
*------------------------- DBSPACE List Utility ----------------------------*
|  Database => SQLDBA                                                       |
|                                                                 %  %      |
| SEL SPNO  OWNER    NAME              POOL  PAGES  NACTIVE NH IX FR NTAB LK |
| --- ---- -------- ------------------ ---- ------- ------- -- -- -- ---- -- |
|  _   17  PUBLIC   ORDER_DETAIL          4    5120     908  8 33 15    0  S |
|  _    8  PUBLIC   CUSTOMER_ORDER        3    1206      52  8 20  5    1  S |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|  Select: S = SHOW DBSPACE           L = SHOW LOCK           T = Table List |
|          R = REORG   A = ACQUIRE    U = UPDATE STATISTICS   D = DROP       |
|                                                                           |
|                              Page 1     of 1                              |
*---------------------------------------------------------------SQMDLST2-----*
PF:  1 Help         3 QUIT         4 StatHistory
     9 Sort/Name   10 Sort/Spno   11 Sort/Pages    12 Sort/Pool
```

Figure 217. DBSPACE List Utility Screen

Commonly requested information is displayed for each instance of the object type you are listing. For DBSPACES, this includes:

- DBSPACE number
- Owner
- Name
- Storage pool
- Number of pages
- Number of active pages
- Number of header pages
- Percent index
- Percent free
- Number of tables
- Lock mode

At the bottom of each list screen, the commonly used DBA commands and utilities which can be invoked directly from the list screen are displayed. For DBSPACES, this includes:

- Show DBSPACE
- Show Lock
- Table List
- Reorganize DBSPACE
- Acquire DBSPACE
- Update Statistics
- Drop DBSPACE

The PF key selections offer further function such as the ability to access help or exit to the next higher level screen. Other PF keys allow you to alter the display of information itself. For example, the DBSPACE List screen offers these PF key functions:

- PF01 Help
- PF03 Quit
- PF04 Display Reorganization/Update Statistics Information
- PF09 Sort by DBSPACE Name
- PF10 Sort by DBSPACE Number
- PF11 Sort by Pages
- PF12 Sort by Storage Pool

Invoking the Object Search and List

There are two ways to invoke the Object Search and List:

1. Through the product panel interface.
2. By executing the CMS SQMUTIL exec.

To reach the Object Search and List using the panel interface:

1. Enter Option **U** on the Main Menu and press ENTER.
2. Enter Option **SL** on the Control Center Database Utility Functions screen and press ENTER.

> NOTE:
>
> You can also enter Option **SL** on the Main Menu and press ENTER to go directly to the Object Search and List, bypassing the Database Utility Functions screen.

To reach the Object Search and List directly from CMS, you can:

1. Type "**SQMUTIL**" at the CMS READY prompt and press ENTER.

Navigating the Object Search and List Panels

Once you enter the Object Search and List tool, you have many options which are best described by referring to Figure 218 on page 456.

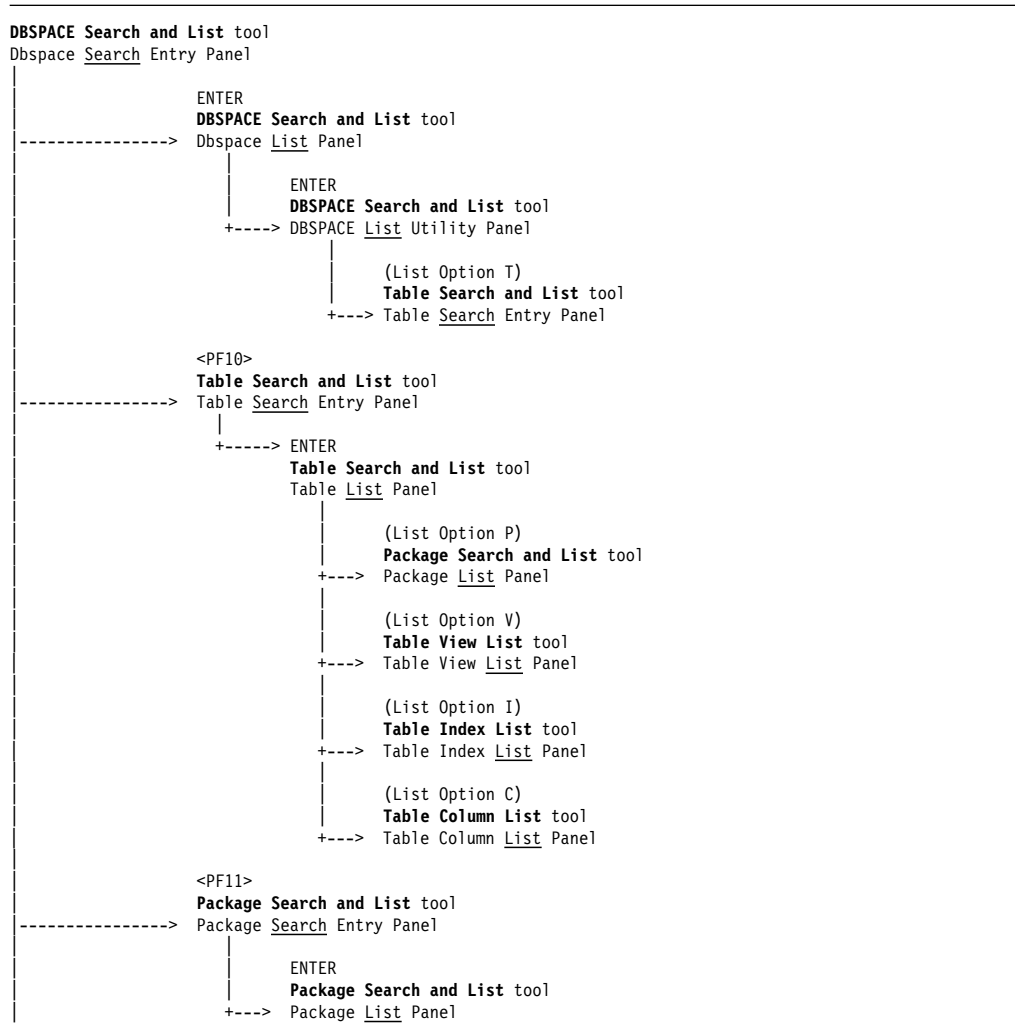


Figure 218. Navigating the Object Search and List Tools

For example, the first panel displayed is the DBSPACE List Utility Screen. From this panel, you can:

1. Press **ENTER** to list DBSPACES
2. Press **PF10** to display the Table List Utility screen
3. Press **PF11** to display the Package List Utility screen

If you want to view package information and you know the package creator and name, you can go directly to the Package List Utility screen, key in the creator and name, and see the information you want. You can also find the same information by displaying DBSPACES and "drilling-down" through the Table List and Package List screens.

Who Can Use the Object Search and List Tools

Using the Object Search and List tools requires database DBA authority or greater.

DBSPACE Search and List Tool

Using the DBSPACE Search and List tool, you can search a database for a specific DBSPACE, several DBSPACES with common or similar attributes, or all DBSPACES. DBSPACES meeting your search criteria are displayed in a list panel. Actions such as performing a DBSPACE reorganization can be executed against specific DBSPACES in the list.

DBSPACE Search Entry Panel

```
 mm/dd/yyyy                    CONTROL CENTER                          hh:mm:ss
*------------------------- DBSPACE List Utility -----------------------------*
|                                                                 CTRLID: MSTRSRV1 |
|  Database => SQLDBA                                               NODE: VMSYSTM1 |
|                                                                                  |
|  DBSPACE OWNER ==> ANDYS               (ALL, PUBLIC, PRIVATE, ownername)         |
|                                                                                  |
|  DBSPACE NAME  ==> ____________________  (blank for ALL, use % for wildcard)     |
|                                                                                  |
|  DBSPACE PAGES:  MINIMUM ==> 0        MAXIMUM     ==> 9999999                   |
|                                                                                  |
|  DBSPACE TYPE ==> A                    (A=Acquired, U=Unacquired, B=Both)        |
|                                                                                  |
|  STORPOOL     ==> _____                (blank for ALL)                           |
|                                                                                  |
|       The % SQL wildcard character can be used when specifying the               |
|       Dbspace Owner or Dbspace Name values for selection.                        |
|                                                                                  |
|                                                                                  |
|                                                                                  |
*----------------------------------------------------------------SQMDLIST-----*
  PF1  HELP      PF3 QUIT    PF4 EXIT    PF5 MAIN MENU     ENTER Process
  PF10 TABLE Search panel    PF11 PACKAGE Search panel
```

Figure 219. DBSPACE Search Entry Panel

Shown in Figure 219 is the search entry panel of the DBSPACE Search and List tool. The database you are currently working with is identified as is the ID of the service machine that controls that database and the node it resides upon. You can change the database name on this panel. To specify another service machine and node, use Option **C** on the Control Center Main Menu. Control Center Communication Path and Database Settings are discussed on page 143.

From this panel, enter your DBSPACE search criteria and press ENTER to process, or go directly to the search entry panel of the Table Search and List tool using program function key 10 (PF10), or to the Package Search and List tool using PF11.

DBSPACE Search Criteria

One or more of these attributes can be specified when searching for a specific DBSPACE or group of DBSPACES:

Entry Field	Description
DBSPACE OWNER	Enter PUBLIC or PRIVATE to search for DBSPACES of a specific type, or ALL for both types. To search by DBSPACE owner, enter a search string. You can use the percent sign (%) to represent a string of zero or more characters in the same way you would with a LIKE predicate in an SQL statement.
DBSPACE NAME	Leave blank to search for DBSPACES regardless of DBSPACE name or enter a search string. You can use the percent sign (%) to represent a string of zero or more characters in the same way you would with a LIKE predicate in an SQL statement.
DBSPACE PAGES	Change the minimum and/or maximum page size values to limit a search based on DBSPACE size.
DBSPACE TYPE	Enter A to search for acquired DBSPACES, U for unacquired, or B for both.
STORPOOL	Enter a storage pool number to search for DBSPACES within a specific storage pool; otherwise, leave blank. You can enter a negative value to search a nonrecoverable storage pool.

DBSPACE List Panel

DBSPACES which satisfy your search criteria are displayed as shown in Figure 220.

```
 mm/dd/yyyy                      CONTROL CENTER                          hh:mm:ss
*-------------------------- DBSPACE List Utility -----------------------------*
|  Database => SQLDBA                                                        |
| SEL SPNO  OWNER    NAME               POOL  PAGES  NACTIVE NH IX FR NTAB LK |
| --- ---- -------- ------------------ ---- ------- ------- -- -- -- ---- -- |
|  _   22  DUKEWE   DUKE_IX               3    1024       1     0 15    1  S |
|  _   21  D128980  D128980A              2    1024       1    20  0    1  S |
|  _   26  D128980  SUSAN_DENNIS_DBS     -4    1024       1    20  0    1  S |
|  _   14  L002653  L002653A              3    1024       1    10  0    1  S |
|  _   12  M356959  M356959A              3    1024       2    31  0    1  S |
|  _   13  M760595  M760595A              3    1024       1    33  0    1  S |
|  _   15  MILBURN  MILBURNJ              3    1024       1    33  0    1  S |
|  _    6  PUBLIC   ANDYS                 1    1024       1    33 15    2  P |
|  _    7  PUBLIC   ETCTPHD_DEV           3    5120       2    25  0    2  T |
|  _    3  PUBLIC   HELPTEXT              1    8192       0    33 10   19  P |
|                                                                            |
|  Select: S = SHOW DBSPACE           L = SHOW LOCK           T = Table List |
|          R = REORG   A = ACQUIRE    U = UPDATE STATISTICS   D = DROP       |
|                                                                            |
|                              Page 1     of 3                               |
*----------------------------------------------------------------SQMDLST2-----*
PF:  1 Help         3 QUIT         4 StatHistory
     9 Sort/Name   10 Sort/Spno   11 Sort/Pages    12 Sort/Pool
```

Figure 220. DBSPACE List Panel

Display Fields

Column Heading Description

SEL	User's processing selection
SPNO	DBSPACE number
OWNER	DBSPACE owner
NAME	DBSPACE name
POOL	Storage pool into which the database places pages that belong to this DBSPACE
PAGES	Number of pages
NACTIVE	Number of active pages
NH	Number of header pages
% IX	Percentage of pages to be used for indexes
% FR	Percentage of space on each page to be kept free when rows are inserted
NTAB	Number of tables in the DBSPACE
LK	Possible values are: **S** if the entire DBSPACE is to be locked. **P** if page locking is to be done in this DBSPACE. **T** if row locking is to be done in this DBSPACE.

Select Options

Select options can be invoked against listed DBSPACES.

Option	Description
Show DBSPACE (S)	Executes the database operator command SHOW DBSPACE.
Show Lock (L)	Executes database operator command SHOW LOCK DBSPACE.
Table List (T)	Invokes the Table Search and List tool to list all tables in the DBSPACE.
Reorg DBSPACE (R)	Executes Multiple User Mode DBSPACE Reorganization tool.
Acquire DBSPACE (A)	Displays the Acquire DBSPACE entry panel shown in Figure 221 on page 461. Executes the database statement ACQUIRE DBSPACE using entered parameters.
Update Statistics (U)	Executes database statement UPDATE STATISTICS FOR DBSPACE.
Drop DBSPACE (D)	Executes a DBSU job to drop the DBSPACE. You will be given the opportunity to verify or cancel the Update Statistics and Drop DBSPACE options before they are executed.

PF Key Selections

PF Key	Description
StatHistory (PF4)	Displays the following columns which contain additional information kept by the Automated DBSPACE Maintenance tools for each DBSPACE: **Heading** — **Description** LAST REORG — Date (*yyddd*) and time (*hh:mm:ss*) of last DBSPACE reorganization. ELAPSED — Time (in minutes) it took to perform DBSPACE reorganization. LAST UPSTAT — Date (*yyddd*) and time (*hh:mm:ss*) of last UPDATE STATISTIC FOR DBSPACE. **Operational Notes:** 1. This function is available *only* if you have previously installed the Automated DBSPACE Maintenance tools. Refer to Chapter 34, “Automated DBSPACE Maintenance Tools” on page 427. 2. Information relating to activities prior to the installation of the Automated DBSPACE Maintenance tool are not shown.
Sort/Name (PF9)	Sorts list by DBSPACE name.
Sort/Spno (PF10)	Sorts list by DBSPACE number.
Sort/Pages (PF11)	Sorts list by DBSPACE page size.
Sort/Pool (PF12)	Sorts list by storage pool number.

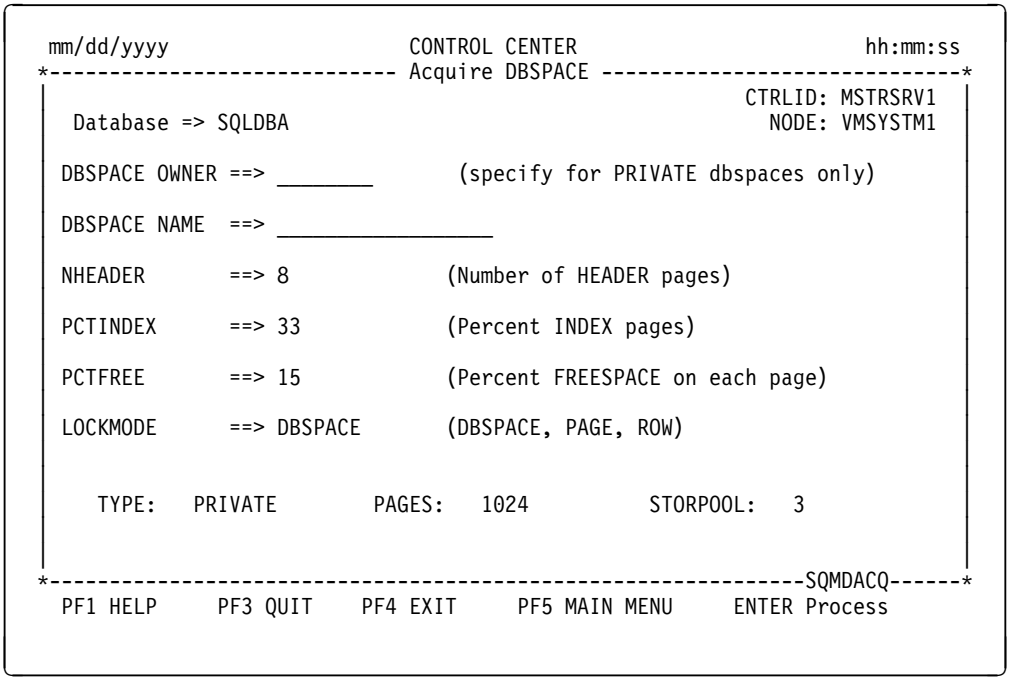


Figure 221. Acquire DBSPACE Entry Panel

Package Search and List Tool

Using the Package Search and List tool, you can search an entire database for a specific package, several packages with common or similar attributes, or all packages. Packages meeting your search criteria are displayed in a list panel. Actions such as performing a package rebind can be executed against specific packages in the list.

Packages can also be listed for a specific table selected using the Table Search and List tool.

Package Search Entry Panel

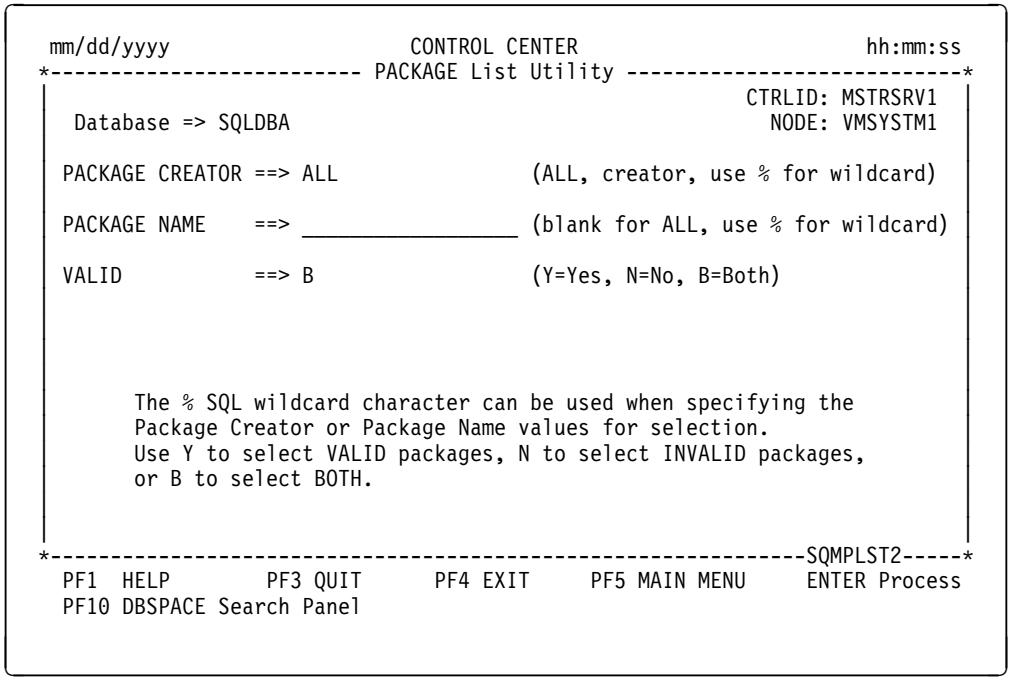


Figure 222. Package Search Entry Panel

Shown in Figure 222 is the search entry panel of the Package Search and List tool. The database you're currently working with is identified by your communication path and database settings shown near the top of the panel. The communication path can be changed using Option **C** from the Control Center Main Menu, whereas the database setting can be changed on this panel. Control Center Communication Path and Database Settings are discussed on page 143.

From this panel you can enter your package search criteria and press ENTER to process, or return to the DBSPACE Search and List tool using program function key 10 (PF10).

Package Search Criteria

One or more of these attributes can be specified when searching for a specific package or group of packages:

Entry Field	Description
PACKAGE CREATOR	Enter search string to search by package creator or ALL. You can use the percent sign (%) to represent a string of zero or more characters in the same way you would with a LIKE predicate in an SQL statement.
PACKAGE NAME	Leave blank to search for packages regardless of package name or enter a search string. You can use the percent sign (%) to represent a string of zero or more characters in the same way you would with a LIKE predicate in an SQL statement.
VALID	Enter Y to search for valid packages, N for invalid, or B for both.

Package List Panel

Packages which satisfy your search criteria are displayed as shown in Figure 223. If you invoked the Package List (**P**) select option from the list panel of the Table Search and List tool, then only packages which reference the selected table are listed.

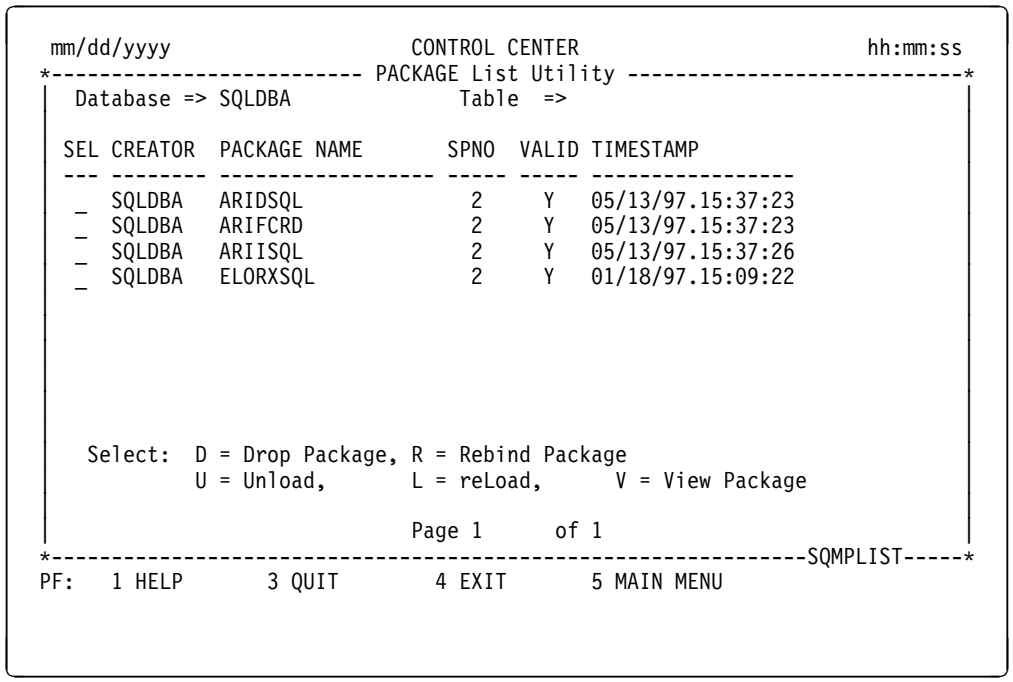


Figure 223. Package List Panel

Display Fields

Column Heading	Description
CREATOR	Package owner.
PACKAGE NAME	Package name.
SPNO	The number of the DBSPACE that contains this package.
VALID	Y if package is valid; N if invalid.
TIMESTAMP	Date and time when package was created. The field has the format *mm/dd/yy.hh:mm:ss*.

Select Options

Select options can be invoked against listed packages.

Option	Description
Drop Package (D)	Executes the database DROP statement against package.
Rebind Package (R)	Invokes the Rebind Package tool to rebind the package.

Unload (U)	Executes Database Services Utility UNLOAD PACKAGE command. Package name is used as output filename. Package creator is used as filetype.
ReLoad (L)	Executes Database Services Utility RELOAD PACKAGE command. **Usage Considerations:** 1. Requires output file generated by Unload (**U**) option. 2. To be used in conjunction with the Unload (**U**) option to rebind a package.
View Package (V)	Determines package attributes and displays using XEDIT. See example output in Figure 224. Output file is given a filename equal to the first eight characters of the package name and a filetype of PACKDATA.

```
 Package:  MSTRUSR1.EMPLOYEP
 Database: SQLDBA

 Package create release:  Unknown
 Last prepped in release: 3.4
 Character Set:  ENGLISH
 Sections:       1

 Preprocessing Characteristics:
 ------------------------------
   Package built using Extended Dynamic
   Database DBCS option OFF at package creation
   Block option:  NOBLOCK
   Package is modifiable
   Describe option: ON

 Options Specified at Prep time:
 -------------------------------
     BLock
     ISOLation(USER)

 DEFAULT Options at Prep time:
 -----------------------------
     RELease(COMMIT), EXPLAIN(NO), NOBLock, KEEP, REPLACE, NOEXIST
     NOCHECK, TIME(default), DATE(default), PERiod, APOST, ISOLation(CS)
     PREPname=EMPLOYEP
     LABEL(default=spaces)
     CHARSUB(Default), CTOKEN(NO)

 Static SQL Statements in package:
 ---------------------------------
UPDATE MSTRUSR1.EMPLOYEE SET TPH_RECTYPE = ? WHERE  WBSID = ? AND
TPH_RECTYPE = ?;
```

Figure 224. Example Output from View Package Option

Table Search and List Tool

Using the Table Search and List tool you can search an entire database for a specific table, several tables with common or similar attributes, or all tables. Tables meeting your search criteria are displayed in a list panel. Actions such as performing a table reorganization can be executed against specific tables in the list.

Tables can also be listed for a specific DBSPACE selected using the DBSPACE Search and List tool.

Table Search Entry Panel

```
 mm/dd/yyyy                    CONTROL CENTER                          hh:mm:ss
*---------------------------- TABLE List Utility -----------------------------*
|                                                              CTRLID: MSTRSRV1 |
|  Database => SQLDBA                                            NODE: VMSYSTM1 |
|                                                                               |
|  TABLE CREATOR ==> M760595             (ALL, creator, use % for wildcard)     |
|                                                                               |
|  TABLE NAME    ==> ____________________  (blank for ALL, use % for wildcard)  |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
|       The % SQL wildcard character can be used when specifying the            |
|       Table Creator or Table Name values for selection.                       |
|                                                                               |
|                                                                               |
|                                                                               |
*-------------------------------------------------------------------SQMTLST2-----*
  PF1  HELP        PF3 QUIT     PF4 EXIT     PF5 MAIN MENU     ENTER Process
  PF10 DBSPACE Search Panel
```

Figure 225. Table Search Entry Panel

Shown in Figure 225 is the search entry panel of the Table Search and List tool. The database you are currently working with is identified as well as the server machine ID and node. To change server machine ID and node, use Option **C** on the Control Center Main Menu. You can specify another database name on this panel. Control Center Communication Path and Database Settings are discussed on page 143.

From this panel, enter your table search criteria and press ENTER to process. To return to the DBSPACE Search and List tool, use program function key 10 (PF10).

Table Search Criteria

One or more of these attributes can be specified when searching for a specific table or group of tables:

Entry Field	Description
TABLE CREATOR	Enter search string to search by table creator or ALL. You can use the percent sign (%) to represent a string of zero or more characters in the same way you would with a LIKE predicate in an SQL statement.
TABLE NAME	Leave blank to search for tables regardless of table name or enter a search string. You can use the percent sign (%) to represent a string of zero or more characters in the same way you would with a LIKE predicate in an SQL statement.

Table List Panel

The Table List tool can be reached in one of two ways.

1. Listing tables in a specified DBSPACE:
   a. On the DBSPACE List Utility panel (SQMDLIST), type values for the parameters and press ENTER to display the DBSPACE List Utility panel SQMDLST2.
   b. On panel SQMDLST2, select the DBSPACE name in which you are interested, type a T in the SEL column, and press ENTER.
   c. The TABLE List Utility panel, SQMTLIST is now displayed showing only the tables in the specified DBSPACE.
2. Listing tables by specifying a search argument for the name.
   a. On the DBSPACE List Utility panel (SQMDLIST), press PF 10 (display Table Search panel).
   b. The Table Search panel, SQMTLST2, is displayed.
   c. On SQMTLST2, enter a TABLE CREATOR and TABLE NAME, optionally using the SQL wildcard character,%. Press ENTER.
   d. The TABLE List Utility panel, SQMTLIST is now displayed showing only the tables that match your search criteria.

The Table List panel is shown in Figure 226 on page 467.

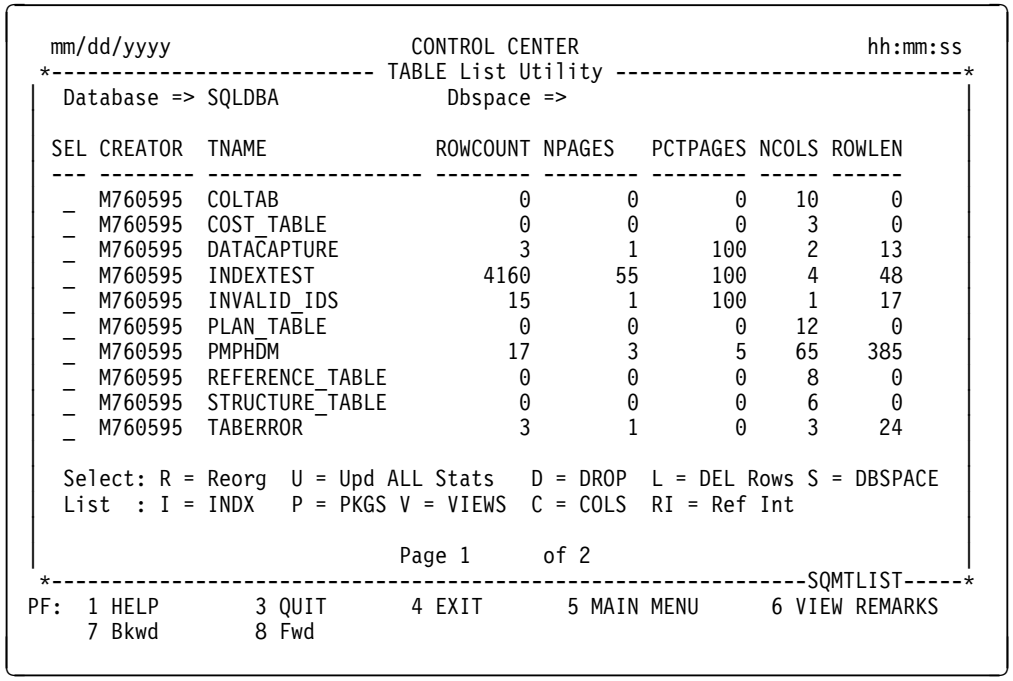


Figure 226. Table List Panel

Tables which satisfy your search criteria are displayed as shown in Figure 226. If you invoked the Table List (**T**) select option from the list panel of the DBSPACE Search and List tool, then only tables in the selected DBSPACE are listed.

Display Fields

Column Head	Description
CREATOR	Table owner.
TNAME	Table name.
ROWCOUNT	The total number of rows in this table.
NPAGES	The number of pages on which rows of this table appear.
PCTPAGES	The approximate percentage of the total active pages in the DBSPACE that have rows from this table on them.
NCOLS	The number of columns in this table.
ROWLEN	The average length of the rows in this table.

Select Options

Select options can be invoked against listed tables. Enter a selection in the SEL field.

Option	Description
Reorg/redefine (R)	Invokes the Table Reorganization and Redefinition tool.
Update ALL Statistics (U)	Executes an UPDATE ALL STATISTICS for the table.
Drop Table (D)	Executes the database DROP statement against the table.

Delete Rows (L)	Deletes all rows in the table.
DBSPACE (S)	Displays DBSPACE attributes.

List Options

List options can be invoked against listed tables.

Option	Description
INDEX List (I)	Invokes the Table Index List tool to list all related indexes.
PACKAGE List (P)	Invokes the Package Search and List tool to list all packages that reference the table.
VIEW List (V)	Invokes the Table View List tool to list all table views.
COLUMN List (C)	Invokes the Column List tool to list all columns in the table.
RefInt (RI)	Presents a summary panel of a table's Referential characteristics.

View Remarks

PF key 6 allows viewing, adding, or changing the remarks column of a table in the catalog. Pressing PF6 causes panel SQMTLST3 to be displayed; see Figure 227.

```
mm/dd/yyyy                    CONTROL CENTER                        hh:mm:ss
*--------------------------- TABLE List Utility ------------------------------*
|  Database => SQLDBA              Dbspace =>                                 |
|                                                                             |
| SEL CREATOR  TNAME               REMARKS                                    |
|     -------- ------------------ ------------------------------------------  |
|  _  M760595  COLTAB              Test Column table                          |
|  _  M760595  COST_TABLE                                                     |
|  _  M760595  DATACAPTURE                                                    |
|  _  M760595  INDEXTEST           Index table for testing                    |
|  _  M760595  INVALID_IDS                                                    |
|  _  M760595  PLAN_TABLE                                                     |
|  _  M760595  PMPHDM                                                         |
|  _  M760595  REFERENCE_TABLE                                                |
|  _  M760595  STRUCTURE_TABLE                                                |
|  _  M760595  TABERROR                                                       |
|                                                                             |
|  Select: R = Reorg  U = Upd ALL Stats  D = DROP  L = DEL  A = ADD Com       |
|  List  : I = INDX   P = PKGS V = VIEWS C = COLS  RI = REF INT               |
|                                                                             |
|                              Page 1      of 2                               |
 *-------------------------------------------------------------SQMTLST3-----*
PF:  1 HELP        3 QUIT       4 EXIT       5 MAIN MENU      6 TABLE DATA
     7 Bkwd        8 Fwd
```

Figure 227. Table Remarks Panel

Display Fields

Column Heading	Description
CREATOR	Table owner.
TNAME	Table name.
REMARKS	The first 42 characters of the REMARKS field from the SYSCATALOG table.

Select/List Options

The Select and List options are the same as those for the preceding Table list screen, with the additional select choice, (A), for adding or changing a table's REMARKS field.

Note: Since only 42 characters of a comment can be displayed, changing the text of a pre-existing, longer comment will result in truncation after the 42nd character.

Index List Tool

You can use the Index List Tool to list all indexes of a table selected using the Table List Tool. You can DROP or reorganize indexes displayed in the list by entering the appropriate selection code.

Index List Panel

```
 mm/dd/yyyy                    CONTROL CENTER                          hh:mm:ss
*--------------------------- INDEX List Utility -----------------------------*
|  Database => SQLDBA              Table  => M760595.INDEXTEST                |
|                                                                             |
| SEL ICREATOR INAME              TYPE RATIO CLUST COLNAMES                   |
| --- -------- ------------------ ---- ----- ----- ------------------------   |
|  _  M760595  IXT1               I-D  10000   F   +NAME,+OWNER,              |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|  Select: D = Drop Index/Key/Constraint  R= Reorganize Index/Key/Constraint  |
|                                                                             |
|                              Page 1      of 1                               |
*-----------------------------------------------------------------SQMILIST-----*
PF:   1 HELP     3 QUIT     4 EXIT     5 MAIN MENU     6 INDEX DATA
```

Figure 228. Index List Panel

Column Heading	Description
SEL	Field for entering a selection option (see below).
ICREATOR	The user ID of the person who created the index.
INAME	The name of the index.

TYPE Indicates the type of index. It is a 3-character value, the first of which denotes the type of index ("regular" index, primary key, or unique constraint); the second of which is always a dash; and the last of which denotes, in the case of a "regular" index, whether duplicates are allowed or whether it is a unique index; and in the case of primary keys and unique constraints, whether they are active or inactive.

Possible values are:

I-D	for regular index, duplicates allowed
I-U	for unique regular index
P-A	for active primary key
P-I	for inactive primary key
U-A	for active unique constraint
U-I	for inactive unique constraint

RATIO Value between 0 and 10000, where 10000 represents a totally clustered index and 0 represents a totally unclustered index.

CLUST Possible values are:

C	if the index is clustered,
N	if the index is not clustered,
F	if the index was the first index created and is now clustered, and
W	if the index was the first index created and is now not clustered.

COLNAMES First 25 characters of the names of the columns on which the index is defined.

Select Options: Select options can be invoked against listed indexes.

Option	Description
D (Drop Index/Key/Constraint)	Executes the database DROP statement against the index or key.
R (Reorganize Index/Key/Constraint)	Executes a Database Services Utility REORGANIZE INDEX statement for the index, and an ALTER TABLE ACTIVATE statement for a key or constraint.

PF key 6 presents the Index List screen with more detailed information about the indexes obtained from the catalog. The new screen is shown in Figure 229 on page 471.

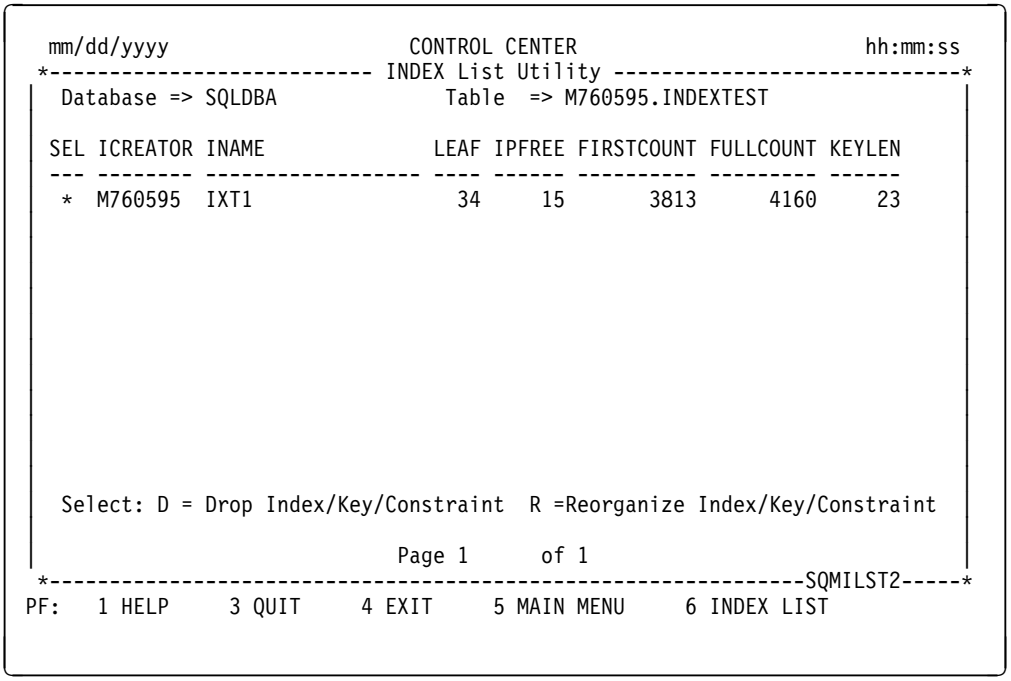


Figure 229. Index Data Panel

Display Fields

Column Heading	Description
ICREATOR	The user ID of the person who created the index.
INAME	The name of the index.
LEAF	Number of lowest level pages in the index.
IPFREE	The amount of free space reserved in the index for later insertions and updates (specified via the PCTFREE parameter in the CREATE INDEX statement).
FIRSTCOUNT	The FIRSTKEYCOUNT value from the catalog. It gives the number of distinct values for the index, considering the first column only.
FULLCOUNT	The FULLKEYCOUNT value from the catalog. It gives the number of distinct values for the index, considering all key columns.
KEYLEN	Indicates the average length of the key field.

The Select options are the same as those of the Index List panel. Pressing F6 on this panel returns you to the Index List panel, Figure 228 on page 469.

View List Tool

Use the View List Tool to list the views on a table selected from the Table List Tool. Any view displayed can be dropped or displayed by entering the appropriate selection code.

View List Panel

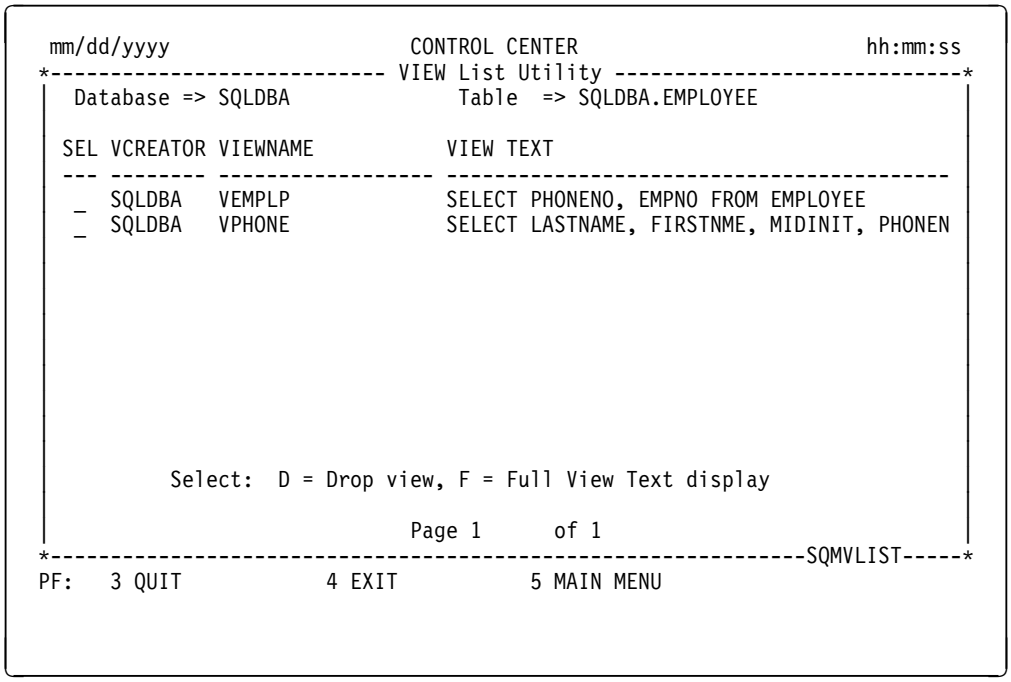


Figure 230. View List Panel

Column Heading	Description
VCREATOR	Owner of the view.
VIEWNAME	View name.
VIEW TEXT	Partial display of the database statement that defined the view. Use Select Option **F** to view the full statement.

Select Options

Select options can be invoked against listed views.

Option	Description
Drop View (D)	Executes the database DROP statement against the table view.
Full View Text display (F)	Displays full view text using XEDIT. See example output in Figure 231. Output file is given file name SQMMENUL VIEW.

```
CREATE VIEW VPHONE (LASTNAME, FIRSTNAME, MIDINITL, PHNUMBER, EMNUMBER,
 DPNUMBER, DEPTNAME) AS SELECT LASTNAME, FIRSTNME, MIDINIT, PHONENO,
 EMPNO, DEPTNO, DEPTNAME FROM EMPLOYEE,
 DEPARTMENT WHERE WORKDEPT=DEPTNO
```

Figure 231. Example Output from Full View Text Display Option

Column List Tool

The Column List tool displays information about all of the columns in a table. It is reached by selecting **C** from the Table Search and List screen.

Column List Panel

```
 mm/dd/yyyy                    CONTROL CENTER                          hh:mm:ss
*------------------------- COLUMNS List Utility ----------------------------*
|  Database => SQLDBA              Table  => SQLDBA.CUSTOMER_PRICE           |
|                                                                            |
|     NAME                TYPE         LENGTH    NULLS  COLNO              |
|     ------------------ ----------- --------- ------ -----                  |
|     ADDED_BY            CHAR                3     N      6                |
|     ADDED_DATE          DATE                4     N      5                |
|     CHANGED_BY          CHAR                3     N      8                |
|     CHANGED_DATE        DATE                4     N      7                |
|     CUSTOMER_NO         INTEGER             4     N      1                |
|     MODEL               SMALLINT            2     N      3                |
|     RETAIL_PRICE_AMT    DECIMAL          1794     N      4                |
|     SELL_LOT            INTEGER             4     N      2                |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
|                              Page 1      of 1                              |
*----------------------------------------------------------------SQMCLIST-----*
PF:  1 Help         3 QUIT
```

Figure 232. Column List Panel

Column Heading	Description
NAME	Column name.
TYPE	Column type.
LENGTH	Internal length of the column as specified in the CREATE TABLE statement.
NULLS	Whether null values are allowed in this column (Y=yes/N=no).
COLNO	The number of the column in the table. Corresponds to the sequence specified in the CREATE TABLE statement.

Table Referential Integrity Tool

Description

The Table Referential Integrity (RI) tool is a comprehensive tool, which is invoked from the Table List panel (Figure 226 on page 467) of the Search List tool.

1. It first presents a Summary panel.
   - The Summary panel (SQMREFLST), shown in Figure 233 on page 475), lists the primary key (if there is one), lists the number of Unique Constraints (if any), lists the tables which have foreign keys defined on this primary key (if existent - these are the dependent tables of the current table), and then lists the parent tables of the current table if there are any foreign keys defined for this table.
   - PF keys from this panel allow you to view more detailed information about the primary key, the foreign keys, the dependent keys, the unique constraints, and a full RI report.
2. The Primary Key Panel, SQMPKLIST, is shown in Figure 234 on page 477; the help panel is SQMHPKLST.
   - This panel lists the key name, its status, the names of up to 8 columns which comprise the key and the columns' TYPE and LENGTH.
   - Various data from the SYSINDEXES table are also displayed (for example, CLUSTER, CLUSTERRATIO).
   - PF keys allow DROP, DEACTIVATE, DEACTIVATE ALL, ACTIVATE, and ACTIVATE ALL functions to be performed on the primary key.
3. The Foreign Key Panel, SQMFKLIST, is shown in Figure 235 on page 478; the help panel is SQMHFLST.
   - There is one panel for each foreign key defined. The foreign key and its status are displayed, and the names of up to 6 columns which comprise the key and the columns' TYPE and LENGTH. In addition the DELETE RULE for that key is displayed.
   - The name of the PARENT TABLE and its key status is displayed.
   - PF keys allow for DROP, DEACTIVATE, and ACTIVATE functions to be performed on the foreign key.
4. The Dependent Key Panel, SQMDKLIST, is shown in Figure 236 on page 479; the help panel is SQMHDKLST.
   - There is one panel for each dependent key defined. The dependent table name, its keyname and status, and the names of up to 6 columns which comprise the key and the columns' TYPE and LENGTH. In addition the DELETE RULE for that key is displayed.
   - PF keys allow DROP, DEACTIVATE, and ACTIVATE functions to be performed on the dependent key.
5. The Unique Constraint Panel, SQMUQLIST, is shown in Figure 237 on page 480; the help panel is SQMHUQLST.
   - There is one panel for each Unique Constraint defined for the Table. The name, status, and the names of up to 6 columns with TYPE and LENGTH are displayed, along with other data from the SYSINDEXES table.
   - PF keys allow DROP, DEACTIVATE, and ACTIVATE functions to be performed on the Unique Constraint.
6. The Full Report Option
   - A full report (sample shown in Figure 238 on page 481) is generated and displayed on the screen in XEDIT mode. The report starts with the current

table, that is, the table which is chosen by the user from the Table List (Figure 226 on page 467). It lists the primary key (if any), the parent tables (if there are foreign keys) and the Dependent tables (if there are any dependent on this table) in tabular format. The report then proceeds down the chain of tables found, doing the same for each table involved in the chain. The report looks somewhat like the example in the SQL Reference manual for the Sample Application in Appendix D, except in words rather than arrows. It gives a comprehensive listing of all tables involved in the Referential Integrity (RI) schema and their relationships to one another, and data about the primary and foreign keys.

Invocation

The Referential Integrity tool is reached through the Search List tool (from the Table List tool RI option), which can be invoked either from the SL option of the Utility Menu or from the stand alone exec SQMUTIL.

The Referential Integrity Summary panel is shown in Figure 233.

```
 mm/dd/yyyy                    CONTROL CENTER                          hh:mm:ss
*---------------------- Referential Integrity Summary -----------------------*
|  Database => SQLDBA              Table  => SQLDBA.DEPARTMENT                |
|                                                                             |
|                Relationship of this table to Other Tables:                  |
|                                                                             |
|   PARENT OF                     STATUS  DEPENDENT OF                STATUS  |
|   ---------------------------- ------  ------------------------- ------    |
|   SQLDBA.EMPLOYEE                A      SQLDBA.EMPLOYEE             A      |
|   SQLDBA.PROJECT                 A                                         |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|   PRIMARY KEY STATUS: ACTIVE                                                |
|   UNIQUE CONSTRAINTS: 1                                                     |
|                                                                             |
*-----------------------------------------------------------------SQMREFLST-----*
PF: 1 HELP   3 QUIT    4 EXIT   5 MAIN MENU   6 PRIMARY KEY   7 FOREIGN KEY
    8 DEPENDENT KEYS   9 UNIQUE CONSTRAINTS  10 FULL REPORT
```

Figure 233. Table RI Summary

Display Fields

Column Heading	Description
Table	Lists the subject table creator and name.
PARENT OF	Lists the tables of which the subject table is defined as a parent.
STATUS	Lists the status of the table's key. Values are: • A - Active • I - Inactive • D - Implicitly Inactive

DEPENDENT OF Lists the tables of which the subject table is a dependent.

STATUS The same as STATUS above.

PRIMARY KEY STATUS Lists the status of the primary key of the subject table (if one exists). Values are:

- A - Active
- I - Inactive

UNIQUE CONSTRAINTS Specifies the number of unique constraints defined for the subject table.

Select Options

Select options are chosen by use of the PF keys. These are described below:

PF Key	Description
PF 1	Invokes Help panel
PF 3	Returns to the Table List panel (Figure 226 on page 467)
PF 4	Fast path exit from Control Center interface to CMS
PF 5	Fast Path return to Control Center Main Menu
PF 6	Display Primary Key panel (if one exists)
PF 7	Display Foreign Key panel(s)
PF 8	Display Dependent Key panel(s)
PF 9	Display Unique Constraints panel
PF 10	Display a Full Report of the subject table and all its referential relations to and with other tables.

The remaining panels are discussed below.

Primary Key Panel

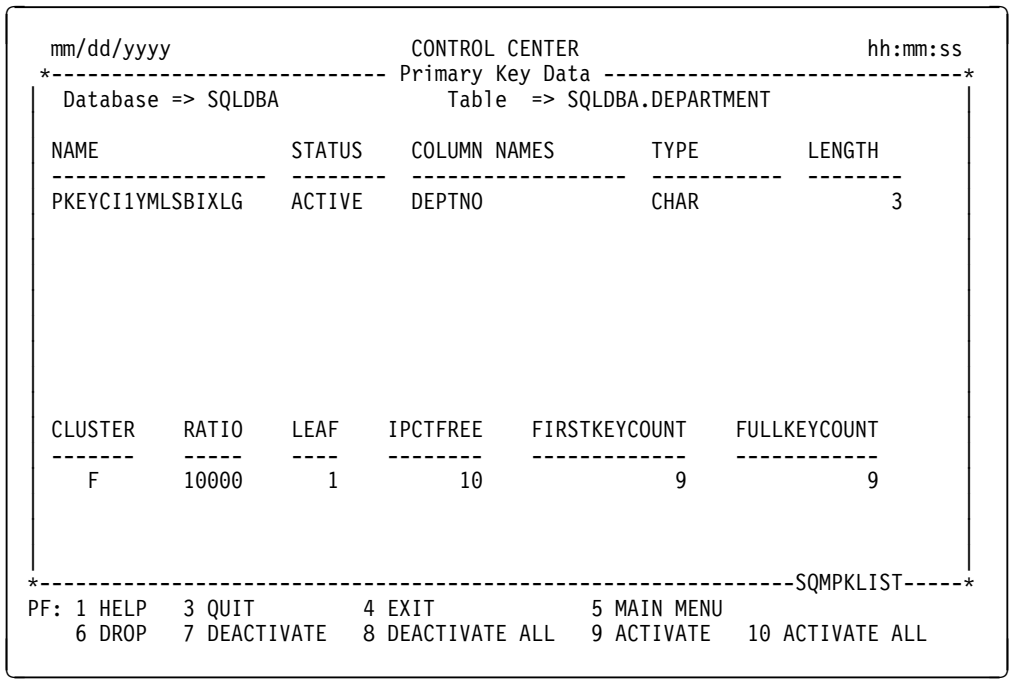


Figure 234. Primary Key Panel

Display Fields

Column Heading	Description
NAME	Primary KEYNAME field from SYSTEM.SYSKEYS.
STATUS	Status of primary key
COLUMN NAMES	The names of up to 8 columns which comprising the primary key.
TYPE	The column type.
LENGTH	The length of the column.
CLUSTER	Values are those from the SYSTEM.SYSINDEXES table.
RATIO	This is the value from the CLUSTERRATIO field from SYSTEM.SYSINDEXES.
LEAF	Values are those from the SYSTEM.SYSINDEXES table.
IPCTFREE	Value is that from the SYSTEM.SYSINDEXES table.
FIRSTKEYCOUNT	This is the value from the SYSTEM.SYSINDEXES table.
FULLKEYCOUNT	This is the value from the SYSTEM.SYSINDEXES table.

Select Options: Select options are chosen by use of the PF keys:

PF Key	Description
PF 1	Invokes Help panel
PF 3	Returns to the REFERENTIAL INTEGRITY SUMMARY panel, Figure 234.
PF 4	Fast path exit from Control Center interface to CMS

PF 5	Fast Path return to Control Center Main Menu
PF 6	DROP primary key
PF 7	DEACTIVATE primary key
PF 8	DEACTIVATE ALL (deactivate primary key, all active foreign keys in the subject table, and all unique constraints defined).
PF 9	ACTIVATE the primary key if inactive.
PF 10	ACTIVATE ALL referential constraints for a primary key, that is activate primary key, and all explicitly inactive foreign keys and unique constraints.

Foreign Keys Panel

```
 mm/dd/yyyy                    CONTROL CENTER                          hh:mm:ss
*--------------------------- Foreign Key Data -------------------------------*
| Database => SQLDBA                Table  => SQLDBA.DEPARTMENT              |
|                                                                            |
|   OPTION:             FOREIGN KEYS OF THIS TABLE:                          |
|          --                                                                |
|   KEY NAME    : R_EMPLY1                      STATUS: A                    |
|   PARENT TABLE: SQLDBA.EMPLOYEE               STATUS: A                    |
|   DELETE RULE : SET NULL                                                   |
|   COLUMN NAMES             TYPE               LENGTH                       |
|   ------------------       -----------        --------                     |
|   MGRNO                    CHAR                      6                   |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
| Select:  D = Drop          DE = Deactivate          A = Activate           |
|                                                                            |
|                             Page 1         of 1                            |
*-----------------------------------------------------------------SQMFKLIST- ---*
PF:   1 HELP         3 QUIT         4 EXIT         5 MAIN MENU
```

Figure 235. Foreign Keys Panel

There is one such screen displayed for each foreign key of the subject table.

Display Fields

Column Heading	**Description**
KEY NAME	Name of the foreign key.
STATUS	Status of foreign key
PARENT TABLE	Name of the parent table (the table which has the primary key which this foreign key references).
STATUS	Status of the parent's primary key.
DELETE RULE	The delete rule for this foreign key.
COLUMN NAMES	The names of up to 6 columns comprising the foreign key.
TYPE	The column type.

LENGTH The length of the column.

Select Options: Select options can be invoked for the foreign key. These are described below:

Option	Description
D	Drop foreign key
DE	Deactivate foreign key (explicit deactivation)
A	Activate foreign key

Dependent Keys Panel

```
 mm/dd/yyyy                    CONTROL CENTER                          hh:mm:ss
*---------------------------- Dependent Key Data -----------------------------*
|  Database => SQLDBA              Table  => SQLDBA.DEPARTMENT                |
|                                                                             |
|    OPTION:             DEPENDENTS OF THIS TABLE:                            |
|           --                                                                |
|    TABLE NAME : SQLDBA.EMPLOYEE                                             |
|    KEY NAME   : R_DEPT1                       STATUS: A                     |
|    DELETE RULE: SET NULL                                                    |
|    COLUMN NAMES             TYPE              LENGTH                        |
|    ------------------       -----------       --------                      |
|    WORKDEPT                 CHAR                     3                      |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|  Select:  D = Drop          DE = Deactivate          A = Activate           |
|                                                                             |
|                              Page 1         of 2                            |
*-----------------------------------------------------------------SQMDKLIST-----*
PF:   1 HELP         3 QUIT         4 EXIT         5 MAIN MENU
```

Figure 236. Dependent Keys Panel

There is one such screen displayed for each dependent of the subject table (that is, each table which has a foreign key referencing the subject table's primary key). The display fields and the Select options are identical to those of the foreign keys panel shown in Figure 235 on page 478.

Unique Constraints Panel

```
mm/dd/yyyy                    CONTROL CENTER                          hh:mm:ss
*------------------------ Unique Constraint Data ---------------------------*
|  Database => SQLDBA              Table  => SQLDBA.DEPARTMENT                |
|                                                                             |
| NAME                STATUS    COLUMN NAMES        TYPE         LENGTH       |
| ------------------  --------  ------------------  -----------  --------     |
| DEPNAME             ACTIVE    DEPTNAME            VARCHAR            36     |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
| CLUSTER    RATIO    LEAF    IPCTFREE    FIRSTKEYCOUNT    FULLKEYCOUNT     |
| -------    -----    ----    --------    -------------    ------------     |
|     C      10000       1          10                9               9     |
|                                                                             |
|                                                                             |
|                              Page 1         of 1                            |
*-----------------------------------------------------------------SQMUQLIST-----*
PF: 1 HELP      3 QUIT      4 EXIT    5 MAIN MENU
    9 DROP     10 DEACTIVATE   11 ACTIVATE
```

Figure 237. Unique Constraint Panel

There is one such screen displayed for each unique constraint defined for the subject table.

The display fields are identical to those of the Primary Key panel (see Figure 234 on page 477).

Select Options: Select options are chosen by use of the PF keys. These are described below:

PF Key	Description
PF 1	Invokes Help panel
PF 3	Returns to the REFERENTIAL INTEGRITY SUMMARY panel, Figure 234 on page 477.
PF 4	Fast path exit from Control Center interface to CMS
PF 5	Fast Path return to Control Center Main Menu
PF 9	Drop the Unique Constraint
PF 10	Deactivate the Unique Constraint
PF 11	Activate the Unique Constraint

Full Referential Integrity Report

Displays a full report of the subject table and all its referential relations to and with other tables. An example of the report for the subject table is:

```
                    REFERENTIAL INTEGRITY REPORT

                      Database: WMAVM1.PSNLDBA
                      08/19/1998      09:08:39


                 Table: SQLDBA.DEPARTMENT
                 -----------------------------------
                |       PRIMARY KEY IS ACTIVE       |
                |  ---> PRIMARY KEY COLUMNS ARE:    |
                |            DEPTNO                 |
                 -----------------------------------

                     Relationship to Other Tables:

PARENT OF:     SQLDBA.EMPLOYEE             STATUS: ACTIVE
               SQLDBA.PROJECT              STATUS: ACTIVE

DEPENDENT OF: SQLDBA.EMPLOYEE              STATUS: ACTIVE
 foreign key: R_EMPLY1               delete rule: SET NULL
     columns:   MGRNO
```

```
                 Table: SQLDBA.EMPLOYEE
                 -----------------------------------
                |       PRIMARY KEY IS ACTIVE       |
                |  ---> PRIMARY KEY COLUMNS ARE:    |
                |            EMPNO                  |
                 -----------------------------------

                     Relationship to Other Tables:

PARENT OF:     SQLDBA.DEPARTMENT           STATUS: ACTIVE
               SQLDBA.EMP_ACT              STATUS: ACTIVE
               SQLDBA.PROJECT              STATUS: ACTIVE

DEPENDENT OF: SQLDBA.DEPARTMENT            STATUS: ACTIVE
 foreign key: R_DEPT1                delete rule: SET NULL
     columns:   WORKDEPT
```

Figure 238 (Part 1 of 3). Full Referential Integrity Report

```
                Table: SQLDBA.PROJECT
                ---------------------------------
               |       PRIMARY KEY IS ACTIVE     |
               |  ---> PRIMARY KEY COLUMNS ARE:  |
               |            PROJNO               |
                ---------------------------------

                    Relationship to Other Tables:

PARENT OF:    SQLDBA.PROJ_ACT             STATUS: ACTIVE

DEPENDENT OF: SQLDBA.DEPARTMENT           STATUS: ACTIVE
 foreign key: R_DEPT2                delete rule: RESTRICT
     columns:   DEPTNO
                            ********

DEPENDENT OF: SQLDBA.EMPLOYEE             STATUS: ACTIVE
 foreign key: R_EMPLY2               delete rule: SET NULL
     columns:   RESPEMP
```

```
                Table: SQLDBA.EMP_ACT
                ---------------------------------
               |       NO PRIMARY KEY DEFINED    |
                ---------------------------------

                    Relationship to Other Tables:


DEPENDENT OF: SQLDBA.EMPLOYEE             STATUS: ACTIVE
 foreign key: R_EMPLY3               delete rule: CASCADE
     columns:   EMPNO
                            ********

DEPENDENT OF: SQLDBA.PROJ_ACT             STATUS: ACTIVE
 foreign key: R_PROACT               delete rule: RESTRICT
     columns:   PROJNO
     columns:   ACTNO
     columns:   EMSTDATE
```

Figure 238 (Part 2 of 3). Full Referential Integrity Report

```
                Table: SQLDBA.PROJ_ACT
                ----------------------------------
               |       PRIMARY KEY IS ACTIVE      |
               |  ---> PRIMARY KEY COLUMNS ARE:   |
               |            PROJNO                |
               |            ACTNO                 |
               |            ACSTDATE              |
                ----------------------------------

                    Relationship to Other Tables:

   PARENT OF:    SQLDBA.EMP_ACT              STATUS: ACTIVE

   DEPENDENT OF: SQLDBA.ACTIVITY             STATUS: ACTIVE
    foreign key: R_ACTIV                delete rule: RESTRICT
        columns:   ACTNO
                              ********

   DEPENDENT OF: SQLDBA.PROJECT              STATUS: ACTIVE
    foreign key: R_PROJ2                delete rule: RESTRICT
        columns:   PROJNO
```

```
                Table: SQLDBA.ACTIVITY
                ----------------------------------
               |       PRIMARY KEY IS ACTIVE      |
               |  ---> PRIMARY KEY COLUMNS ARE:   |
               |            ACTNO                 |
                ----------------------------------

                    Relationship to Other Tables:

   PARENT OF:    SQLDBA.PROJ_ACT             STATUS: ACTIVE


****************        END OF REPORT       **********************
```

Figure 238 (Part 3 of 3). Full Referential Integrity Report

The report starts with the subject table. It lists the table name and whether a primary key has been defined for the table. If so, its status is shown. Next, it lists those tables the subject table is a parent of and their status. Then it lists those tables of which the subject table is a dependent and their status.

The report then proceeds to list the same information for each table involved in a referential relationship with the subject table, printing the same information as for the subject table. If any new tables are found, the same information on them will be printed. This proceeds until all the tables involved have had their information printed.

The report is presented as an XEDIT screen and filed under the name table name REFIREPT after viewing. Only the first 8 characters of the table name are used.

Chapter 37. Control Center Administration Tools

Overview

Use these tools to manage database console-related information and activities. Console management tools include capabilities that will allow you to view what is happening on a database machine, as well as feed input directly to the database console, if required.

Full understanding of the capabilities of the tools included in this chapter is essential knowledge for those persons designated as Control Center database administrators or Control Center administrators.

View Message Log Tool

Use the View Message Log tool to display database console messages that are recorded and maintained by the product. Console messages for all database console activities such as archiving, adding/deleting dbextents, and cold logging are logged by the product for all databases. In addition, any communication messages the product issued to the databases are logged.

This tool will allow you to display console messages for a given period of time for a single database machine. Use this tool when you need to review database console messages for problem determination, database activity (archive, recovery) verification, monitoring current console activities, and so on.

Who Can Use the View Message Log Tool

Use of the View Log tool requires Control Center database user, operator, or greater level of authorization.

How the View Message Log Tool Works

All communications between a Control Center support machine and a database virtual machine are captured through the Single Console Image Facility (SCIF) and are recorded by the product, in a log file on the machine's 191 A-disk. Refer to Chapter 2, “Architecture” on page 13. A new log file is created each day and is used to record a day's worth of database console messages. The product will keep as many log files as specified during the product installation process.

Each daily log file has a filename of SQLMSTR and a filetype in the format LOG*yyddd*, where *yy* is the current year and *ddd* is the day of the year (1 - 365). Entries are made to the SQLMSTR LOG*yyddd* file in a chronological order as the database console messages occur. Each entry in the log will be preceded with the date and time that the entry was made. The type of message will be indicated for each entry, such as RDR (reader file received by the product machine), MSG (message received by the product machine), or SCIF (Single Console Image Facility message received from a database machine).

When multiple database machines are managed by a single product virtual machine, the SQLMSTR LOG file will contain entries in chronological order regardless of which database is being communicated with. This causes messages

from different databases to be interspersed throughout the log file, making it difficult to view all communications between the product and a single database. The View Message Log tool will filter the log file and display all messages relating to a single database machine.

View Message Log Panel

Figure 239 shows the View Message Log panel, which is reached using Option **V** on the Control Center Main Menu.

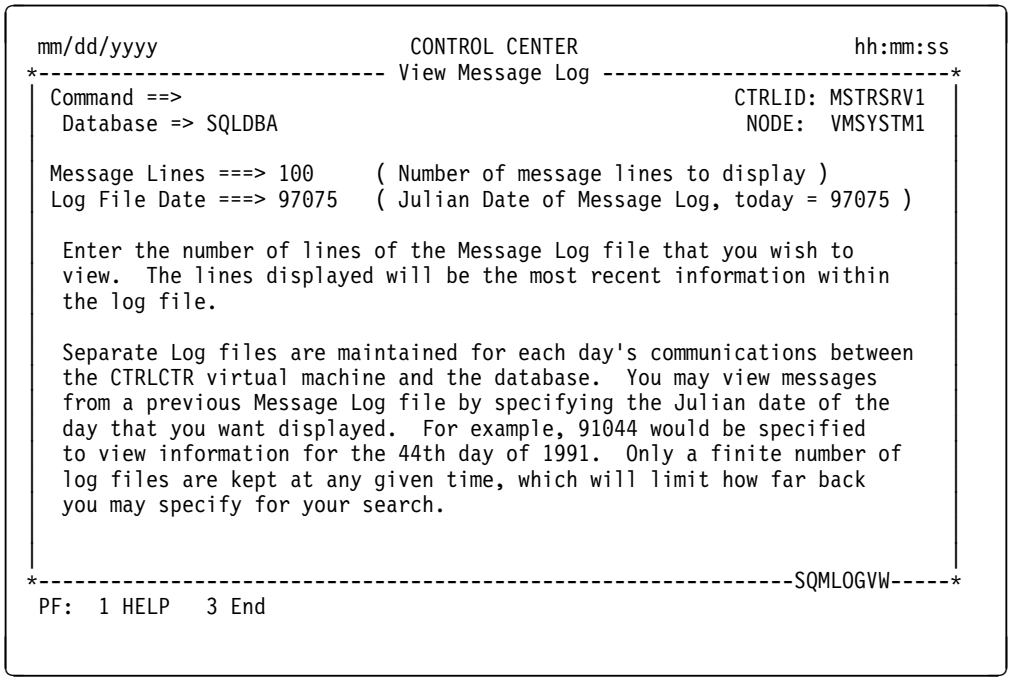


Figure 239. View Message Log Panel

Entry Field	Description
Message Lines	The Message Lines entry field is used to specify how many lines from the SQLMSTR LOG*yyddd* file are to be displayed. The lines displayed will be the most recent available. For instance, if you specify 100 lines, the latest 100 lines from the log file that pertain to the specified database will be displayed. If the number of lines specified is greater than the number of entries available in the log file, all lines will be displayed.
Log File Date	The Log File Date entry field is used to specify which daily log file should be used for the display search. Control Center will default to the most recent Log file (current day). You can specify a previous day's log file by entering a date in Julian format (*yyddd*, where *yy* is a two-digit year and *ddd* is the day number of the year). The product will only maintain a limited number of old log files, limiting the available search. The number of log files maintained by the product is specified during the Control Center service machine installation process.

Sample View Message Log Display

Figure 240 on page 488 is an example of a View Message Log file. Each message in the file is preceded by the date and time that it was recorded by the product.

The example Message Log file covers the startup of the SQLDBA database. All database messages from the database are shown as message type SCIF. Other messages with message type MSG are sent to the product machine with the CP MSG command. The messages with message type CP are messages generated by VM/CP. The last message in this log listing is a product command sent from a user who requested to view the Message Log.

```
 Userid: MSTRSRV1                                    Date: 03/15/97
At Node: VMSYSTM1                                    Time: 13:54:18

     COMMAND EXECUTED:   QUERY LAST 50 MESSAGES of LOG 97075

  DISPLAYED: Last 50 messages between MSTRSRV and database SQLDBA on 97075

10:11:59 MSG  VMSYSTM1 MSTRSRV1 LABELDEF ISSUED FOR SQLDBA: 300
10:11:59 MSG  VMSYSTM1 SQLDBA   SQMSTATU C SQLDBA  STARTUP LOG LABELDEF
10:12:00 MSG  VMSYSTM1 SQLDBA   SQMSTATU C SQLDBA  STARTUP TRACE LABELDEF
10:12:00 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: DMSLBD441R ENTER VOLID INFORMATION:
10:12:02 CP   VMSYSTM1 MSTRSRV1 HCPQCS150A USER SQLDBA  HAS ISSUED A VM READ
10:12:02 MSG  VMSYSTM1 SQMTAPEU SQLDBA NEW_STATUS FILDEF 200 TRACE DATA A 191
10:12:04 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0717I START SQLSTART: 03/15/97 10:12 EST.
10:12:05 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0320I THE DEFAULT DATABASE NAME IS SQLDBA.
10:12:05 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0663I FILEDEFS IN EFFECT ARE:
10:12:05 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: Z        DISK     DMSNAM   LOADLIB  *
10:12:05 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARIARCH  TAP1  SL  00001  VOLID QU1405
10:12:05 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARILARC  TAP3  SL  00001
10:12:05 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARITRAC  DISK     TRACE    DATA     A1
10:12:06 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARISQLLD DISK     ARISQLLD LOADLIB  Q1
10:12:06 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: BDISK    DISK     300
10:12:06 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: LOGDSK1  DISK     201
10:12:06 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: LOGDSK2  DISK     202
10:12:06 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: DDSK1    DISK     200
10:12:06 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: DDSK2    DISK     204
10:12:06 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: DDSK3    DISK     220
10:12:07 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: DDSK4    DISK     221
10:12:07 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: DDSK5    DISK     222
10:12:07 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: DDSK6    DISK     223
10:12:07 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: DDSK7    DISK     224
10:12:08 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: DDSK8    DISK     225
10:12:08 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARIUSRDD DISK     USERLIB  LOADLIB  *
10:12:09 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0025I PROGRAM ARISQLDS IS LOADED AT 856000.
10:12:09 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0025I PROGRAM ARIXRDS IS LOADED AT 6FF000.
10:12:09 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0025I PROGRAM ARIXSXR IS LOADED AT 957000.
10:12:09 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0025I PROGRAM ARICMOD IS LOADED AT 95B000.
10:12:10 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0015I ACCOUNT PARAMETER VALUE IS N.
10:12:10 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0015I DUMPTYPE PARAMETER VALUE IS P.
10:12:10 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0015I LOGMODE PARAMETER VALUE IS L.
10:12:11 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0015I STARTUP PARAMETER VALUE IS W.
10:12:11 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0015I SYSMODE PARAMETER VALUE IS M.
10:12:11 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0015I EXTEND PARAMETER VALUE IS N.
10:12:12 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0015I CHARNAME PARAMETER VALUE IS ENGLISH.
10:12:12 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0015I DBNAME PARAMETER VALUE IS SQLDBA.
10:12:12 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0015I PARMID PARAMETER VALUE IS SQLDBA.
10:12:13 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0015I TRACDSC PARAMETER VALUE IS 00.
10:12:13 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0015I TRACRDS PARAMETER VALUE IS 0000000.
10:12:13 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0016I ARCHPCT PARAMETER VALUE IS 79.
10:12:14 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0016I CHKINTVL PARAMETER VALUE IS 10.
10:12:14 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0016I NCSCANS PARAMETER VALUE IS 30.
10:12:14 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0016I NCUSERS PARAMETER VALUE IS 5.
10:12:15 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0016I NDIRBUF PARAMETER VALUE IS 50.
10:12:15 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0016I NLRBS PARAMETER VALUE IS 2520.
10:12:15 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0016I NLRBU PARAMETER VALUE IS 1000.
10:12:16 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0016I NPAGBUF PARAMETER VALUE IS 100.
10:12:16 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0016I SLOGCUSH PARAMETER VALUE IS 90.
10:12:16 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0016I SOSLEVEL PARAMETER VALUE IS 10.
10:12:16 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0016I DISPBIAS PARAMETER VALUE IS 7.
10:12:17 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0283I LOG ANALYSIS IS COMPLETE.
10:12:17 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0282I LUW UNDO IS COMPLETED.
10:12:17 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0281I LUW REDO IS COMPLETED.
10:12:17 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0060I DATABASE INITIALIZATION COMPLETE.
10:12:17 SCIF VMSYSTM1 MSTRSRV1 SQLDBA: ARI0045I READY FOR OPERATOR COMMUNICATIONS.
13:54:17 MSG  MSTRUSR  SQMSFILE SQLDBA SQMSFILE SQLDBA MESSAGES 100 92009
```

Figure 240. Message Log File

Issue Database Commands Tool

The Issue Database Commands tool will allow you to enter any line command directly to the database machine's console, as if you were logged onto the database machine itself. Obviously, this is a very powerful and flexible capability providing *direct access* to your database machine's console, and should therefore be used with extreme care.

Use this tool when you need to invoke any CP or CMS commands on the database virtual machine. By using this tool, you will not need to log onto the database machine and therefore, the SCIF (Secondary Console Image Facility) link between the product and the database machine will not be broken.

Who Can Use the Issue Database Commands Tool

Only those persons with Control Center database Administration-level or greater authorization can use this tool.

How the Issue Database Commands Tool Works

The product manages database consoles by using the Secondary Console Image Facility (SCIF). Refer to Chapter 2, “Architecture” on page 13. This facility gives the product control of the database virtual machine's virtual console and virtual keyboard. The Issue Database Commands tool allows authorized users to utilize this relationship.

Issue Database Commands Entry Panel

The Issue Database Commands entry panel shown in Figure 241 is displayed when Option **CO** on the Database Utility Functions panel is selected. Authorized users can send any line command to the database machine through this panel.

```
 mm/dd/yyyy                    CONTROL CENTER                        hh:mm:ss
*----------------------------- Database Command -----------------------------*
| Option ===>                                                CTRLID: MSTRSRV1  |
|    Database ===> SQLDBA                                     NODE:  VMSYSTM1  |
|                                                                              |
|                                                                              |
|     I  IMMEDIATE COMMAND          Immediately issue command                  |
|                                                                              |
|     S  SCHEDULE COMMAND           Schedule later command                     |
|                                                                              |
|                                                                              |
|         Enter OPTION followed by the COMMAND to be executed, then            |
|         press ENTER to process                                               |
|                                                                              |
|         NOTE:    You may execute CP commands while the database is          |
|                  running if you begin the command with CP                    |
|                                                                              |
|         EXAMPLE:   I CP DETACH 181  would detach a tape, even while          |
|                                     the database is active                   |
|                                                                              |
|                                                                              |
*------------------------------------------------------------------SQMDBCMD-----*
 PF:   1 Help    3 End
```

Figure 241. Issue Database Commands Entry Panel

Entry Field	Description
Option	Enter line command preceded by either an `I` or an `S`. An `I` prefix will immediately send the command to the database. An `S` prefix will allow the command to be scheduled for execution at a later date and time. When the database is running the database, then CMS commands will *not* be allowed to process on the database virtual machine. CP commands can be executed on the database machine while the database is running if they are preceded with `CP`. For example, to immediately execute a detach command for the archive tape drive while the database is running, this command could be entered on the Database Command panel: `I CP DETACH 181`. The `I` instructs the product to execute the command immediately and the `CP` instructs the product to submit the command to the database as a CP command.

Scheduling Commands

Commands can also be scheduled, which will allow dependencies to be established, such as requiring the target database to be in an up or down state before the command is to be issued. For example, this command schedules a message to be sent to user ID: `OPERATOR` on node `VMSYSTM2` when the database is down:

```
S TELL OPERATOR AT VMSYSTM2 Please start database SQLDBA at VMSYSTM1
```

List Control Center VM Database Files Tool

Use this tool to list all the product's database files for a specified database. All files maintained by it for a given database will be listed. From this list you are able to browse and update any of the files.

Who Can Use the List Control Center VM Database Files Tool

Use of the tool display option (view only) requires Control Center database User, Operator, or greater authorization. Use of the tool update options (modify or delete) requires database Administration or greater level of authorization.

How the List Control Center VM Database Files Tool Works

The List Control Center VM Database Files tool is invoked by selecting Option **L** on the Database Utility Functions panel. All files with filenames matching the database machine name will be listed. Files in this list may then be displayed, modified, or deleted.

List Control Center VM Database Files Panel

Figure 242 on page 491 shows the List Files panel, with files listed for database machine SQLDBA.

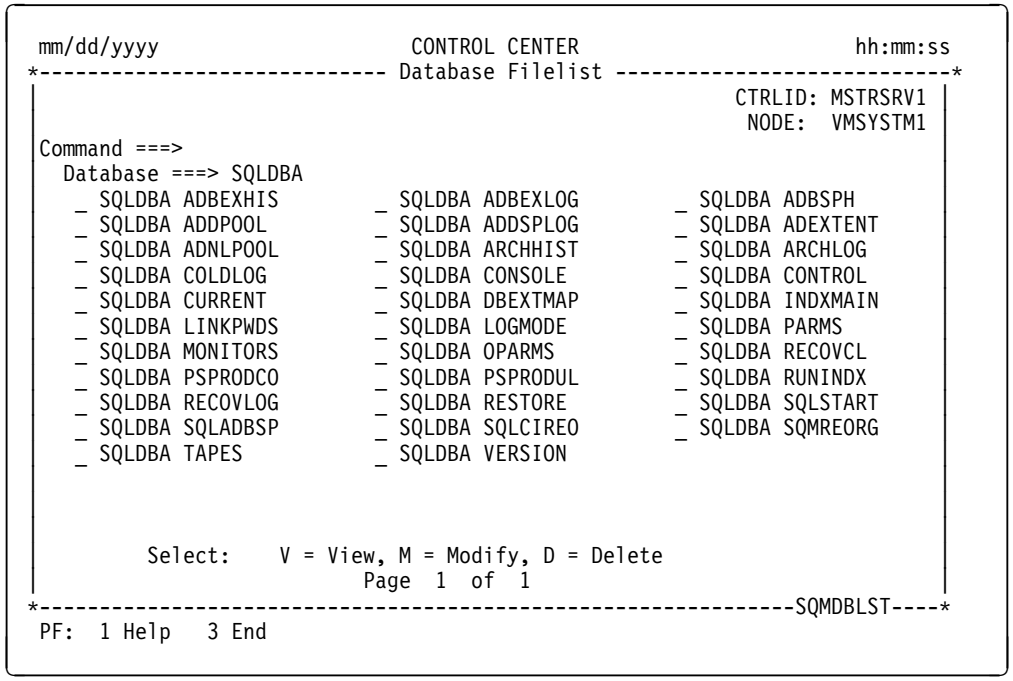


Figure 242. Database Filelist Screen

Select Options

Select options can be invoked against listed files.

Select Option	Description
View (V)	Displays the selected file. The file is sent to your virtual reader from the Control Center service machine and is displayed using CMS PEEK. When you exit the file, it is purged from your reader.
Modify (M)	Allows the selected file, which resides on the Control Center service machine, to be modified. The file is sent to your virtual reader and subsequently received to your 191 A-disk. You are then placed into edit (XEDIT) mode. When you have completed your changes, exit the file using PF3 or FILE. You are then given the opportunity to migrate the modified file to the Control Center service machine. After migration or cancel migration has been selected, the file is deleted from your A-disk.
Delete (D)	Deletes the selected file from the Control Center service machine's 191 A-disk.

Note:

Updating database files can be dangerous! Files such as the PARMS file and TAPES file have separate update functions within the product panel interface that perform validation on all changes. Using the Modify option to update files should be avoided whenever possible to prevent typing mistakes which can cause the product functions to fail.

Note:

Use extreme caution when deleting files to avoid the possible deletion of a file required for the product operation.

Types of Database Files

The files listed in Figure 242 on page 491 are typical files kept by the managing Control Center service machine for any given database. A brief description of each file listed in this example is given below.

Filetype	Description
ADBEXHIS	History file of dbextents that have been added and deleted.
ADBEXLOG	Log file of the last execution of the Add DBEXTENTS (SQLADBEX) tool.
ADBSPHIS	History file of DBSPACEs that have been added to the database.
ADDPOOL	History file of recoverable storage pools that have been added to the database.
ADDSPLOG	Log file of the last execution of the Add DBSPACE (SQLADBSP) tool.
ADEXTENT	Command file used by the Add DBEXTENTS tool which specifies the pool number and virtual address of each dbextent to add.
ADNLPOOL	Command file used by the Add DBEXTENTS tool which specifies the pool number of each new non-logging storage pool to be added.
ARCHHIST	History file of archive activity needed by the product to automate the recovery process.
ARCHLOG	Log file of the last archive performed on the specified database.
COLDLOG	Log file of the last execution of the Database COLDLOG tool.
CONSOLE	Log file of the last product console captured.
CONTROL	Log file that contains the product database authorization list, the the database archive parameters and the database startup parameters.
CURRENT	Database startup parameters used when the database was last started.
DBDELEXT	Command file used by the Delete DBEXTENTS tool which specifies the dbextents to be deleted.
DBEXTMAP	DBEXTENT to Storage Pool mapping (displayed using the Query DBEXTENTS/STORPOOL Mapping tool).
DDBEXLOG	Log file of the last execution of the Delete DBEXTENTS tool run on the database.
LOGMODE	Logmode switch message file containing the database message which indicates the old and new logmode.
MONITORS	Master file containing all monitor routines for the database.
PARMS	Master file containing all parameters associated with the database.

RECOVCTL	Recovery control file which contains all information for the restore set requested by the user when selecting a database recovery.
RECOVLOG	Log file of the last database recovery for this database.
RESTORE	A report of available restore sets as selected by the user when initiating a database recovery.
RUNINDX	A report generated by the Index Reorganization tool.
SQLADBSP	Control file used by the Add DBSPACE tool which specifies the DBSPACEs to be added to the database.
SQLCIREO	Log file of the last execution of the Reorganize Catalog Indexes tool.
SQLSTART	Console log from the database machine which contains the database startup process. The console is closed and transferred to the Control Center service machine each time the database starts.
SQMREORG	Log file of the last single user mode reorganization.
TAPES	Master file containing information on all tapes or disk files that are used for archiving, log archiving, and tracing on the database.
VERSION	Log file of the version, release and modification level of the database database.

Query DBEXTENTS/STORPOOL Mapping Tool

Use this tool to display the DBEXTENT/STORPOOL mapping report for a specified database. The information contained in the report provides valuable information relating to a database's dbextents.

Who Can Use the Query DBEXTENTS/STORPOOL Mapping Tool

To use this tool you will need Control Center database User, Operator, or greater authorization.

How the Query DBEXTENTS/STORPOOL Mapping Tool Works

Mapping information is gathered each time the database is started and is kept in a file on the Control Center service machine's 191 A-disk. The filetype is DBEXTMAP and the filename matches the database machine name.

The product first determines all virtual addresses for each database minidisk by reviewing the database's SQLFDEF (refer to the *DB2 Server for VM System Administration* manual). After gathering virtual addresses, the product then reads the database's directory minidisk (BDISK) to determine the mapping of dbextents to database storage pools.

Query DBEXTENTS/STORPOOL Mapping Report

You can display the DBEXTENTS/STORPOOL Mapping report by selecting option QE on the Database Utility Functions panel. Figure 243 on page 494 shows an example of the DBEXTENT/STORPOOL Mapping report. Information in the report is presented in both dbextent and storage pool sequences.

```
                    DBEXTENT/STORPOOL Mapping
                            07/24/1998

                ******** Dbextent Sequence ********

                    Virtual           Blocks/    DASD            Real
Dbextent  Storpool  Address   Pages  Cylinders   Type   Volid    Address
--------  --------  -------   -----  ---------  ------  ------   -------
  BDISK                200             100 CYL   3380   USM0M8   113E
 LOGDSK1               201              75 CYL   3380   USM0M8   113E
 LOGDSK2               220              75 CYL   3380   USM0HG   1125
    1          1      202     14934    100 CYL   3380   USM0M8   113E
    2          8      209     14934    100 CYL   3380   USM0M9   113F
    3          1      20A     14934    100 CYL   3380   USM0M9   113F
    4          1      20F     15561    104 CYL   3380   USM0M4   113A
    5          6      20E     20064    134 CYL   3380   USM0ME   1144
    6          2      210      7467     50 CYL   3380   USM0OA   1156
    7          6      212      3534     20 CYL   3390   USM0L0   11D2
    8          6      215      3534     20 CYL   3390   USM0L1   11D3
    9         -3      204      4446     25 CYL   3390   USM0JG   11C5


                 ********* Pool Sequence *********

                    Virtual           Blocks/    DASD            Real
Storpool  Dbextent  Address   Pages  Cylinders   Type   Volid    Address
--------  --------  -------   -----  ---------  ------  ------   -------

    1          1      202     14934    100 CYL   3380   USM0M8   113E
               3      20A     14934    100 CYL   3380   USM0M9   113F
               4      20F     15561    104 CYL   3380   USM0M4   113A

    2          6      210      7467     50 CYL   3380   USM0OA   1156

   -3          9      204      4446     25 CYL   3390   USM0JG   11C5

    6          5      20E     20064    134 CYL   3380   USM0ME   1144
               7      212      3534     20 CYL   3390   USM0L0   11D2
               8      215      3534     20 CYL   3390   USM0L1   11D3

    8          2      209     14934    100 CYL   3380   USM0M9   113F
```

Figure 243. Example DBEXTENT/STORPOOL Mapping Report

Column Heading	Description
DBEXTENT	A dbextent number assigned to the extent during execution of the Add DBEXTENTS tool.
Storpool	The database storage pool number that a given dbextent resides in. DBEXTENTS are placed in specific storage pools during execution of the Add DBEXTENTS tool.
Virtual Address	The database machine's virtual address location for the dbextent listed.
Pages	The number of 4K pages allocated (used and unused) to this dbextent.
Blocks/Cylinders	Either (depending on DASD type) the number of blocks or cylinders allocated (used and unused) to the minidisk at the specified virtual address.
DASD Type	Indicates the type of DASD device (3330, 3350, 3380, 3390, ...).
Volid	The unique system identifier for the device located at the listed real address.

Real Address	The unique system address for the listed volid.

Appendix A. Control Center Messages

The majority of the product messages are derived from CP, CMS, and the database messages, which are intercepted by the Control Center service machine, acted on in some way, and possibly passed on to Control Center Administrators and users.

The DATABASE MESSAGES file, located on the product code disk, contains a list of messages that are expected from database machines, along with the module that is invoked whenever the message occurs. Typically, the filename of the executable module and the identifier of the message that gets passed on to the user are derived from the original message identifier.

All product messages sent to users will begin with the three characters SQM, followed by:

- the first character of the original message,
- the numeric portion of the original message, and
- the severity indicator of the original message.

  Severity indicators include:
  E Error
  I Information
  W Warning

For example, when message ARI0043I is received, Control Center module SQMA043I is executed on the service machine. This module will in turn send a message with identifier SQMA043I to appropriate users. The message text associated with the message may differ from that of the original message (ARI0043I), since the product may have additional information to supply.

Control Center Message/Action Report

Provided with the product is the *Control Center Message/Action Report* (file DATABASE MSGLIST on the code disk). It contains a list of messages that are expected from database machines, along with the module that is invoked whenever the message occurs and a description of the module. It can be viewed using XEDIT by entering this command in the option input field of the product panel interface:

```
CMS X DATABASE MSGLIST
```

A partial listing of the *Control Center Message/Action Report* (DATABASE MSGLIST file) is shown in Figure 244 on page 498.

```
1            Control Center - Message/Action Report                PAGE: 1

 Message #   Action    Description of Message/Action
 ----------  --------  ------------------------------------------------
 ARI0015I    SQMA015I  This module is executed by CTRLCTR when message
                       ARI015I is received (Database startup, parameter
                       value messages).

 ARI0016I    SQMA015I  This module is executed by CTRLCTR when message
                       ARI015I is received (Database startup, parameter
                       value messages).

 ARI0020I    SQMA015I  This module is executed by CTRLCTR when message
                       ARI015I is received (Database startup, parameter
                       value messages).
```

Figure 244. Partial Listing of the Control Center Message/Action Report

Appendix B. Changing Database Name

If you need to change the name of a database (dbname, not machine name) that is managed by Control Center, you must make changes to database system files and Control Center control files.

For the sake of this discussion, OLDNAME refers to the name of the database that is being changed, and NEWNAME is the new name it is to be given.

Database Changes

1. Log onto the OLDNAME virtual machine.
2. Access the database 195 production code disk in WRITE mode with an available access mode.
3. Use the CMS RENAME command to rename the following files to reflect the new database name:

```
     Old Name              New Name
------------------   ------------------
OLDNAME  SQLDBGEN Q   NEWNAME  SQLDBGEN Q
OLDNAME  SQLDBN   Q   NEWNAME  SQLDBN   Q
OLDNAME  SQLFDEF  Q   NEWNAME  SQLFDEF  Q
```

4. Use XEDIT to edit file NEWNAME SQLDBN Q.

   Locate the following statement and replace OLDNAME with NEWNAME (save your changes):

```
DBMACHID=dbmachid,DCSSID=Dcssid,DBNAME=OLDNAME
```

NOTE: If your database name differs from the VM machine ID, please make the appropriate entry in the Control Center DBINIT Control file. See “Maintaining the DBINIT CONTROL File” on page 123. Also be sure to include the database name in the VM directory IUCV statement, as the example below shows:

```
IUCV ALLOW
IUCV *IDENT SQLDBA LOCAL
```

Control Center Changes

1. Log onto the Control Center service machine that manages the database.
2. Use XEDIT to edit the database parameters file which resides on the service machine's A-disk. The database parameters file is named '*database* PARMS', where *database* is the database machine ID.

   Locate the following statement and replace OLDNAME with NEWNAME (save your changes):

```
:Dbname.OLDNAME
```

Note:

Tokens in this file are case-sensitive; do not alter.

3. Use XEDIT to edit the following two files; replace references to OLDNAME with NEWNAME (save your changes):

   ```
   DBINIT CONTROL
   SQLMSTR DIRECTRY
   ```

   These files exist on the Control Center code disk.

Appendix C. Password Support

Control Center supports the use of minidisk passwords for READ and MULTIPLE-WRITE links that a database machine may require during processing; (at startup, when running single user mode tools). Additionally, if READ passwords are required for the database production disk, password support can be granted to selected users and to the Control Center support machine.

If Data Restore has been enabled on your Control Center service machine, then a password file is required to provide the SQLDBA password to Data Restore and, if necessary, to support the Data Restore machine"s links to the database"s directory, log, data and archive disks.

If Data Restore will not be managed by Control Center then password support is an optional feature that is activated simply by establishing a special database password file on the managing Control Center service machine, and on the user and support machines if needed. No additional installation steps or changes in your system configuration are required. Support is deactivated by removing the password files from each userid/machine.

Note: If you are *not* using Data Restore, and if you are not using minidisk read/write passwords in your environment, or are using a security access product such as RACF, you should *not* activate the password support feature of Control Center.

Links Performed by Control Center

Control Center code executing on the database machine, support machine, and user/DBA machine only performs the dynamic links described below. In general, the database will only link to the database's production and service disks, and tape manager code.

1. Multiple-write (M) links to the database production disk are ONLY performed by the database and ONLY when executing SUM functions ADD, DELETE and COPY DBEXTENTS. NO WRITE links are performed.
2. Only the database links to the service disk (193). The service disk is ONLY linked in READ "RR" mode.
3. Only the database links to the tape manager code. The tape manager code disk is ONLY linked in READ (RR) mode.
4. The database will link "RR" to permanent disks owned by Support machines and user machines that are used for SUM dbspace reloads.

   Note: This link is eliminated with the Single User Mode DBSPACE Reorganization tool. It performs the entire reorganization on the database and requires the permanent disk to already be linked/accessed if used.
5. Control Center Service and Support machines, and users only link to the database production disk in READ (RR) mode. This only occurs when DBINIT is executed to initialize to a new database.

6. Control Center and database functions do NOT perform links to the Control Center code disk. Links to the Control Center code disk are performed in the VM Directory or the PROFILE EXEC.

The database will use userid and virtual address information contained in the database parms files and the SQLMSTR control file when linking to its production and service disks, and to the tape manager. Userids and addresses in the LINKPWDS file should match these values.

Password File Setup

To use the password link facility with a given database, you must create a file containing the minidisk owner ID (user ID), minidisk device address, and read and multi-write passwords of all minidisks that the database machine will potentially link to during processing (Control Center code disk, DB2 Server for VM production disk, or tape manager code disk). This file must be named *dbmachid LINKPWDS*, where dbmachid is the database machine ID, and be formatted as shown below.

```
userid   cuu       readpw   multi-writepw
userid   cuu       readpw   multi-writepw
userid   cuu       readpw   multi-writepw
```

*Figure 245. Example SQLMACH LINKPWDS File. Columns 1-8 **must** contain the user ID of the minidisk owner. The other columns are free form with only a blank necessary between values. The values must be specified in the order shown.*

```
DYNAMT   187       readpw   multi-writepw
userid   cuu       readpw   multi-writepw
userid   cuu       readpw   multi-writepw
```

Figure 246. Example dbname LINKPWDS File

This file must then be sent to the managing Control Center service machine, where it will automatically be received with filemode A0 for security reasons. A Control Center Administrator can later view the file by requesting the service machine to send a copy to their user ID. The commands for doing this are given below:

```
SQM (MSTRSRV
SQM CMS SENDFILE SQLMACH LINKPWDS A0 userid
```

where `MSTRSRV` is the service machine ID, `SQLMACH` is the database machine ID, and `userid` is the Administrator's ID.

Note:

- The Control Center service machine needs a separate *dbmachid LINKPWDS* file for each database it manages.
- The Control Center service machine also needs a file named *mstrsrv LINKPWDS* that contains the combined password information for all the databases it manages. The format of this file is the same as the individual *dbmachineid LINKPWDS*. *Mstrsrv* is the Control Center service machine ID.

READ Passwords on the Database Production Disk

Most Control Center functions invoke DBINIT (SQLINIT) which executes a READ (RR) link to a database production disk. This link is performed by the user or Control Center Support machine. For practical reasons, most VM systems do not require READ passwords for database production disks in order to ease access to the database.

If there are no passwords on the production disk or if the READ password is set to "ALL", then Control Center machines and users will not need their own password file.

If READ passwords ARE required on the production disk, then a separate LINKPWDS file will have to be created for every userid that will be executing Control Center code. These steps have to be done prior to using the system:

1. Create a LINKPWDS file named usermachid LINKPWDS A0, where usermachid is the userid of the person or Support machine that will be executing the Control Center code.
2. The usermachid LINKPWDS A0 file will contain the database userid, production minidisk address, and the READ password for every database that requires production disk passwords. The format is the same as the database LINKPWDS file.
3. Only the READ password is needed in the password file.
4. The usermachid LINKPWDS file will have to be placed on the A-disk of usermachid. If the file is sent to the Support machine using the CMS SENDFILE command, the file will automatically be received.

How it Works

At startup time, the database machine runs the DBSTART exec, which requests a copy of the LINKPWDS file from the service machine. The service machine responds by sending the proper LINKPWDS file, if it exists, to the database machine, where it is received to the A-disk. Then, when an exec runs on the database machine and tries to link to a disk (including the database code and service disks), the proper read and multiple-write passwords are extracted from the file and used in the LINK command. For all other users and Control Center machines, the LINKPWDS file must already exist on the userid's A-disk.

Password Support for Data Restore

If Data Restore functions are to be managed by the Control Center service machine, then a password file *must* be present on the Control Center service machine's 191 disk for each Data Restore machine. This password file will automatically be sent to the Data Restore machine before each Data Restore function is executed. The Data Restore machine's password file is required:

- To supply the server password for the "SQLDBA" userid to the Data Restore SYSIN file during UNLOAD, RELOAD and SELECT operations,
- To supply READ passwords, if necessary, to the Data Restore SYSIN file for UNLOAD, RELOAD, SELECT operations, .
- To supply WRITE passwords, if necessary, to the Data Restore SYSIN file for BACKUP and RESTORE operations, and

- To supply READ passwords, if necessary, to link to the database's archive disks.

Data Restore requires that the Data Restore machine have READ and WRITE access to all of the database's data, directory and log disks. Additionally, if the database performs log archives or full archives to disk, then Control Center support for Data Restore functions requires that the Data Restore machine have READ authority to the database's log archive and full archive disks. This link authority is in addition to any other password support required in the previous sections.

This set up is required:

1. All of a database's disks must have the same READ password. Different databases may have different passwords.
2. All of a database's disks must have the same WRITE and MULTI-WRITE password. Different databases may have different passwords.
3. The WRITE and MULTI-WRITE password must be the same on the same database.
4. If the database performs archives to disks, then the database must use the same CUU (virtual link) as the real address (link address) to link and access these disks. The CUU that is designated in the database TAPES file is recorded in the ARCHHIST file for the database when an archive completes. If a Data Restore TRANSLATE or RELOAD RECOVERY=Y is performed, the disk address recorded in the ARCHHIST file will be linked in order to read the archive.
5. The password file for the Data Restore machine is in addition to any password files you already use or require as described in previous sections.

Password File Setup for the Data Restore machine

One password file (LINKPWDS) will be required for each Data Restore machine. This file will contain password information about each database the Data Restore machine supports.

To use the password link facility with a given Data Restore machine, you must create a file named *drmachid LINKPWDS*, where *drmachid* is the Data Restore machine ID. The file should contain:

1. The database machine name, the label "DRF" and the read and write passwords for the database directory, data disks, log disks and archive disks. There should be only one "DRF" entry for each database.
2. The database machine name, the label "DBAPW" and database password (as recorded in SYSTEM.SYSUSERAUTH") for the userid "SQLDBA".
3. If passwords are required for links to other disks (Control Center code, database production, or tape manager code disk), additional entries must be added as described in the "Password File Setup" section.

An example of a formatted *drmachid LINKPWDS* file is shown below.

```
dbmach1   DRF     readpw1  writepw1
dbmach1   DBAPW   password1
dbmach2   DRF     readpw2  writepw2
dbmach2   DBAPW   password2
dbmach3   DRF     readpw3  writepw3
dbmach3   DBAPW   password3
```

*Figure 247. Example Data Restore machine LINKPWDS File. Columns 1-8 **must** contain the VM user ID of the database. The other columns are free form with only a blank necessary between values. The values must be specified in the order shown.*

This file must then be sent to the managing Control Center service machine, where it will automatically be received with filemode A0 for security reasons. A Control Center Administrator can later view the file by requesting the service machine to send a copy to their user ID. The commands for doing this are given below:

```
SQM (MSTRSRV
SQM CMS SENDFILE SQLMACH LINKPWDS A0 userid
```

where `MSTRSRV` is the service machine ID, `SQLMACH` is the database machine ID, and `userid` is the Administrator's ID.

Appendix D. Master Scheduling Tool

The Master Scheduling tool is a Control Center Administrator's tool for scheduling and managing jobs and events. It provides options for directly accessing the Job Scheduling tool where it can be used to schedule non-Control Center jobs; list all jobs known to the Control Center service machine using the Job Schedule List tool; and for viewing and scheduling product time events. In this section we discuss the latter two options and how to schedule non-Control Center jobs using the Job Scheduling tool. The Job Scheduling tool and Job Schedule List are discussed in Chapter 14, “Job Scheduling Tool” on page 165.

Before You Begin

You should read and understand the material presented in Chapter 14, “Job Scheduling Tool” on page 165 before continuing.

Who Can Use the Master Scheduling Tool

Use of the Master Scheduling tool requires Control Center Administrator authority.

About Viewing Times Events

Selecting the Master Schedule **(MS)** option on the Control Center Main Menu displays the Administrator Events option panel (not shown). On this panel, Option **VE** (View Times Events) allows you to view and delete entries in the SQLMSTR TIMES file using the the Event Schedule Display/Delete panel shown below. The SQLMSTR TIMES file contains entries for all events known to the Control Center service machine. This includes jobs, monitors, and product internal routines.

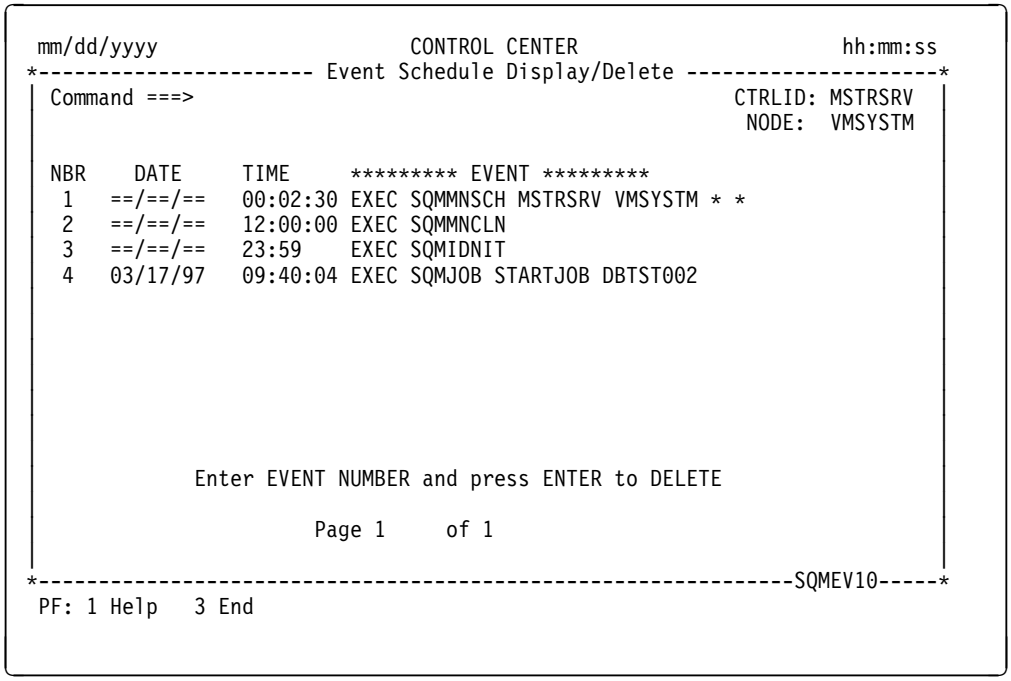


Figure 248. Event Schedule Display/Delete Panel

Individual events may be deleted by entering the event number and pressing ENTER.

Note:

The events listed below are product internal routines and are required for normal Control Center operation. They should not be deleted.

EXEC SQMMNSCH MSTRSRV VMSYSTM * *
: Control Center monitor scheduling routine. This routine is initiated at the beginning of each day. It adds an event to the SQLMSTR TIMES file for the first execution of each monitor scheduled for that day. After a monitor completes, a new event is added for the next execution that day.

EXEC SQMMNCLN
: Control Center monitor cleanup routine. This routine is initiated at midday to remove old monitor events from the SQLMSTR TIMES file.

  Note: Operational NoteDepending on the number of monitors executed by Control Center, you may want to schedule this event several times a day to periodically clean out the SQLMSTR TIMES file.

EXEC SQMIDNIT
: Control Center new-day initialization routine. This routine is initiated at the end of each day to perform daily closeout processing and new-day initialization.

About Scheduling Times Events

Events may be scheduled using option **SE** (Schedule Administrator Events), available on the Administrator Events option panel (not shown).

Note:

Event commands are executed on the Control Center service machine. They *must not* prompt for any input or be invoked in CMS FULLSCREEN MODE. Commands that, do will disable Control Center from further operation and require manual restart of the Control Center service machine.

Scheduling an event requires that you first specify the command that is to be executed using the Administrator Event entry panel shown below and then specify the time, date, and frequency of execution using the Event Schedule entry panel (not shown). Like jobs, events may be scheduled to execute once or repetitively.

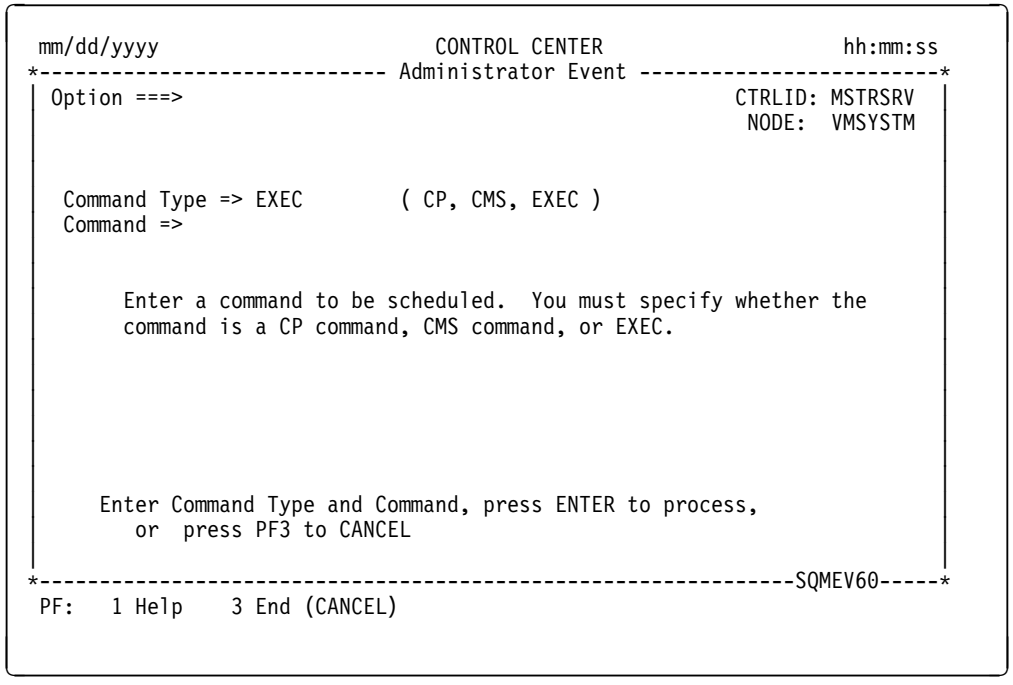


Figure 249. Administrator Event Entry Panel

Events are entered using the Administrator Event Entry panel shown above. Pressing ENTER will display the Event Schedule entry panel (not shown), where you can schedule the event for a single execution or repetitive execution.

Entry Field	Description
Command Type	Either CP, CMS, or EXEC. If the command is an executable MODULE, then enter CMS.
Command	The command name.

Using the Master Scheduling to Schedule Non-Control Center Tools

Using the Job Scheduling tool, you can schedule your own internally developed tools to execute on a Control Center support machine, regardless of whether the tool operates against a database or not.

External Tool Requirements: External tools must meet the following requirements before they can be scheduled for execution using the Job Scheduling tool:

1. Must not prompt for any input or invoke CMS FULLSCREEN MODE.

   The tool will execute on a Control Center support machine which is operating in disconnect mode. Therefore, it must run from start to finish without manual intervention.

2. Exit with a return code.

   A zero return code is interpreted as a successful execution; all other return codes are interpreted as a job failure.

3. Finally, the tool code must be accessible to the Control Center support machine where it will execute.

Output files created by your tool which are left on the 191 A-disk of the Control Center support machine are accessible using the Job Schedule List tool.

Scheduling External Tools: To schedule execution of your tool, select Option **SJ** (Schedule Job) from the Administrator Events option panel to invoke the Job Scheduling tool. All entry fields are the same as discussed earlier, except now, since it is not tied directly to another product tool, the Execute field is blank. Enter the command to invoke your tool as if you were entering it on the console yourself. For example, if your tool is REXX exec TSTPROC EXEC, you would enter TSTPROC in the Execute entry field.

Appendix E. Authorizations

Control Center Authorization Levels

The following defines the five-level hierarchy of Control Center authorization:

- Level 1 - Database User
- Level 2 - Database Operator
- Level 3 - Database Administrator
- Level 4 - Database Machine
- Level 5 - Control Center Administrator

Level 1 is the lowest level of authorization to the Control Center product, meaning that level 1 is limited to the smallest subset of Control Center tools. Level 5, on the other hand, is the highest level of authorization to the Control Center product and can access and use all Control Center tools. This is a hierarchy, meaning each level is authorized to all tools at its level and all levels beneath it (Level 2 = Database Operator + Database User). For more information regarding these levels, refer to “Authorization” on page 26.

Authorization to System Administration Tools

Table 13 is a list of System Administration tools and the authorization required to use each tool. You can change the authorization required by modifying the SQLMSTR PROFILE file on your Control Center service machine's 191 A-disk. To change a particular tools authorization, locate the referenced tool ID in the SQLMSTR PROFILE file and change the authorization level as required. For example, to prevent Database Operators (level 2) from being able to start database archives, you would locate the SQMARCH tool ID in the SQLMSTR PROFILE file and change the level required to 3 (meaning you need Database Administrator or higher authorization to use this tool).

Important!

Database User, Operator, and Administrator authorities are specified per database. This means Database Administrator authority to one database does not automatically give you Database Administrator authority to another. The authorization to databases is controlled and managed by the database parameters file on the Control Center 191 A-disk (refer to “About the Database Parameters Tool” on page 213).

Table 13 (Page 1 of 3). Control Center Default Authorization List

Tool ID	Menu Path	Level	Description
COUNTER	(O,CO)	1	Issue a specified Counter operator command
SHOW	(O)	1	Issue a specified SHOW operator command
SQMDBLST	(U,L)	1	List all files associated with a given database
SQMEVDIS	(MS,VE)	1	View events in the Master Schedule

Table 13 (Page 2 of 3). Control Center Default Authorization List

Tool ID	Menu Path	Level	Description
SQMQSTAT	(S,D/S/N/A)	1	Query status of databases
SQMSFILE	(list opts)	1	Display/list files requested by user
VERSION	(cmode only)	1	Display Control Center version information
CANCEL	(A,C)	2	Cancel a currently running database Archive
FORCE	(O,F)	2	Force an active user off/out of the database
RESET	(O,R)	2	Reset specified database Counter(s)
SET	(O,SP/SS/ST)	2	Set specified database Counter(s)
SQMARCH	(A,I/S)	2	Initiate a database archive
SQMCUARC	(cmode only)	2	Update Control Center's user archive status variables
SQMDBEGN	(SI/SS)	2	Start a database
SQMDBEND	(E,N/Q/S/SQ)	2	Stop a database
SQMDBINI	(P,U)	2	Read a database PARMS file into Control Center memory
SQMMNSVU	(M,S)	2	Database monitor Schedule display
SQMMODEU	(S,D/S/N/A,c)	2	Update Control Center SQMODE database status variable
SQMQUARC	(cmode only)	2	Query internal Control Center user archive variables
SQMRECOV	(R,I/S)	2	Database recovery setup process (not start)
SQMRECQT	(R,C)	2	Cancel a currently running database recovery
SQMRECST	(R,I/S,...)	2	Control Center database recovery START
SQMRECTQ	(R,I/S,...)	2	Query tapes routine during recovery setup
SQMSTATU	(S,D/S/N/A,c)	2	Update Control Center internal DBSTATUS variable
SQMTAPEA	(TM,S)	2	Add a tape to a database TAPES file
SQMTAPEU	(TM,M)	2	Update database TAPES file menu options
SQMACCES	(cmode only)	3	Invoke reaccess of Control Center code disks
SQMADBEX	(U,E)	3	Add DBEXTENTs utility
SQMADBSP	(U,A)	3	Add DBSPACEs utility
SQMCDBEX	(U,C)	3	Copy/Move DBEXTENTs utility
SQMCIREO	(U,RC)	3	Database System Catalog Index reorg (SQLCIREO)
SQMCOLDL	(U,CL)	3	Initiate a database COLDLOG (SQLLOG)
SQMCONS	(G,CO/SO/SS)	3	Control Center console Close, Stop, and Start options
SQMDCHKQ	(cmode only)	3	Answer a Monitor query request
SQMDBCMS	(U,CO)	3	Tell a database to issue a CMS command
SQMDDBEX	(U,D)	3	Delete DBEXTENTs utility
SQMEVDEL	(MS,VE,d)	3	Delete an event from the Master Schedule
SQMEVENT	(MS,SE)	3	Schedule an event in the Master Schedule
SQMFLDEL	(G,F,d)	3	Schedule an event in the Master Schedule
SQMFLIST	(VJ - G,F)	3	Display a job file or a Control Center file
SQMJDONE	(cmode only)	3	Invoke Control Center Job Completion Handler routine
SQMJEXEC	(VJ,s)	3	Start a job menu option
SQMJSTRT	(cmode only)	3	Control Center Job initiator routine
SQMMNRPR	(M,P- M,L,p)	3	Database monitor report Purge/Refresh
SQMMNSL	(M,L)	3	Database monitor List

Table 13 (Page 3 of 3). Control Center Default Authorization List

Tool ID	Menu Path	Level	Description
SQMMNSN	(M,V/L/M/A)	3	Database monitor View, Modify, Add, Delete
SQMMNSR	(M,R- M,L,r)	3	Database monitor Report display
SQMREORF	(U,SU)	3	Initiate a full single user mode reorg job
SQMREORG	(U,RS)	3	Initiate a single user mode database reorg job
SQMREOTI	(U,SU)	3	Collect tape control data for reorg unload job
TRACE	(O,TS/TO)	3	Initiate/Stop database tracing
CMD	(cmode only)	5	Display string specified on Control Center console
CMS	(A,G)	5	Tell Control Center to execute a specified CMS command
RDRLIST	(G,R)	5	Display Control Center's current reader list
SQMDBU	(cmode only)	5	Specific field update for database PARMS file
SQMPROFU	(AU,A/M/D)	5	Update Control Center user authorizations
SQMQPROF	(AU,L)	5	View Control Center user authorizations

Menu Path indicates the panel selections required to invoke a specific tool ID. The COUNTER (O,CO) tool, for example, is invoked by selecting Option **O** (Operator commands) on the main Control Center panel, followed by Option **CO** (Counter command) on the next panel displayed. Options listed with '/' or options after '-' indicate alternative selection paths. 'cmode only' indicates that the tool can only be invoked through the Control Center command mode interface (refer to Appendix G, “Command Mode Interface” on page 523).

Authorization to a Database Administration Tools

The Database Administration tools are not listed in the SQLMSTR PROFILE file, and therefore there are no corresponding tool IDs listed for these types of tools. This is because these are database application programs that are controlled by database connect authorizations. Use of the DBSPACE Reorganization tool for example, requires database connect authority. If a user has not been granted DBA connect authorization to a specified database, then that user would not be able to run any of the Database Administration tools for that database.

Appendix F. User Archiving

There are three parameters related to user archiving in the database PARMS file: Submit_routine, Cancel_routine, and Uarchive_enabled. Setting Uarchive_enabled to "Y" tells Control Center to use the user archive command when a database archive is required (switching logmodes from Y to L; or, after certain single user mode tools are executed, the Add DBEXTENTS tool). If this field is set to "N", then normal database archiving is used in these implicit archive situations. Parameters Submit_routine and Cancel_routine name REXX execs that are invoked by the Control Center service machine when a user archive command or a database cancel command, respectively, is received.

Sample execs ($execs) have been provided on the code disk. These can be modified according to your type of user archive. These $execs have been provided as samples to help implement user archiving:

- UARCSUB
- UARCCAN
- SPBSTART
- SPBEND
- SPBANLZ

The above execs will be referenced in the discussion of a sample user archive implementation which follows below.

Process Flow

The following paragraphs explain the flow of events when the UARCHIVE command is received:

Control Center changes the database STATUS to UARCHIVEDOWN. Next, a log archive is taken if logmode=L. The database then comes down and Control Center checks the Submit_routine field for a valid name. If there is no name, or the name is invalid, Control Center issues an error message and exits. The database remains down; the database status is still UARCHIVEDOWN; an internal global variable remains NONE. This variable is called UARCSTAT and is used to respond to the database when it comes back up after a user archive and requests a response to whether the user archive was successful or not. The four values it can have are NONE (initial setting), RUNNING (when the user archive command is sent to the database), SUCCESSFUL (when the message is received from the disconnected server machine that the archive was successful) and FAILED (when the message sent by the service machine designates a failed user archive). The DBA would then have to correct the problem after determining the cause. This will be elaborated upon later.

If the submit routine exists, Control Center sets the UARCSTAT global variable to RUNNING, invokes the routine, then updates the ARCHIST and ARCHLOG files with archive starting information. The sample submit routine provided with the product (UARCSUB $EXEC) sends a message to the database to execute a TELL command to a disconnected server ID (support machine) to start a backup job. A support machine is used here so that the Control Center service machine is free to

attend to database servicing. The support machine will receive any information or messages from the backup product since it is the ID which interfaces directly with the backup product.

The database then awaits a response from the server ID or DBA about the success or failure of the archive (the response corresponds to the messages in the message file). Two messages (UARC001I and UARC002E) are included in the DATABASE MESSAGES file which cause Control Center to react to a successful or failed user archive notification.

If the response signifies a successful archive (UARC001I), the UARCSTAT global variable is set to SUCCESSFUL; if it signifies a failure (UARC002E), the global is set to FAILED. The database is then brought back up; Control Center responds 1 or 0 (good or bad) to the database's inquiry about whether the user archive was successful, and normal operations resume. The global variable is then set to NONE and the database status to NORMAL.

Preparing for User Archiving

The following is a list of steps to be followed when setting up the user archiving facility:

1. DBA supplies execs for the START (UARCSUB $EXEC) archive command and CANCEL (UARCCAN $EXEC) user archive command and updates the corresponding fields (Submit_routine, Cancel_routine) of the database PARMS file of each database which will be user archived.
2. DBA optionally sets the uarchive_enabled field in each database PARMS file to "Y" for use by Control Center to determine which type of archive to use during DB2 Server for VM-initiated archives (logmode switching). If this field is not "Y", then a database archive will be performed.
3. Set up the TAPES file for each database to look like archive to tape. An example is:

   ```
   100  ARCHIVE  00000  00:00:00 UNUSED UARCH1
   100  LOG      00000  00:00:00 UNUSED VM1201
   200  ARCHIVE  97008  19:01:52 UNUSED UARCH2
   200  LOG      97009  07:54:28 UNUSED VM1202
   300  ARCHIVE  97009  15:41:43 UNUSED UARCH3
   300  LOG      00000  00:00:00 UNUSED VM1203
   ```

   The key point is that there must be one entry for each ARCHIVE series so that a proper FILEDEF can be done by Control Center (the tape ID can be a dummy volid, as shown above).

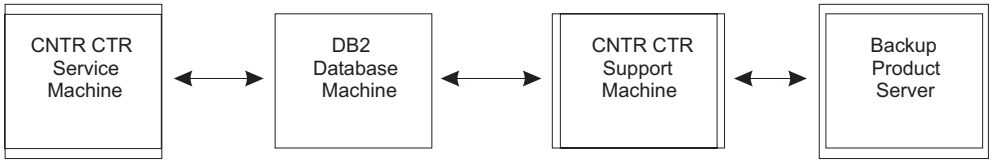
   Note: It is strongly suggested that a regular TAPES file be set up to ensure that, in certain cases when a database archive is initiated, the TAPES file is set up to handle the archive. This would also allow for interleaving user and regular archives which could occur if the Uarchive_enabled switch was set to "N" and a database-initiated archive took place.
4. DBA provides the program on the support machine to initiate the user archive (if this architecture is chosen), SPBSTART $EXEC (see below). Or, if some other approach is used, the DBA gets the user archive started.
5. DBA provides the mechanism for sending the successful or failed message (UARC001I or UARC002E) back to the database upon completion of the user

archive. If the architecture described below is used, then the SPBEND $EXEC can be used.

6. In the event of any problems whereby the messages cannot be sent to the database, the DBA can update the global variable UARCSTAT by using the SQM command (SQM database UARCSTAT newstatus) to reset the UARCSTAT variable to FAILED or SUCCESSFUL, depending on the condition.

Sample User Archive Architecture

The following diagram depicts four disconnected server machines. The backup product may be SYBACK, VMBACKUP, DDR, or any other product. All interfacing to the backup product is actually done through a Control Center support machine. This is done to make the management of the backup jobs much simpler. Depending on the product, the Control Center support machine may receive many messages and/or files from the backup product. Additionally, there is only one machine to authorize to the backup product.



Steps Required on the Service Machine

The following must be done to or on the Control Center service machine:

1. UARCSUB and UARCCAN exec:

   Create from provided templates the UARCSUB and UARCCAN execs. The templates have commented sections of code explaining where the updates should be made. Shown below are the lines in these execs which must be updated. Replace `support_id` with the name of the support machine ID:

   ```
   if uarchive
     then 'CP SEND' dbname 'TELL support_id UARCHIVE START BACKUP'
     else 'CP SEND' dbname 'TELL support_id UARCHIVE START RESTORE'
   ```

   Note that the first word (UARCHIVE) in the message to the support machine *must* match the keyword identified in the support machine's SQLMSTR PROFILE - this is covered again in the support machine section below.

2. database PARMS file

   For each database that will be user archived, the database PARMS file must be updated to specify a "submit-routine" matching the name of the UARCSUB $EXEC you create. Also update the PARMS file to specify the "cancel-routine" and indicate "Y" or "N" for "uarchive-enabled".

Steps Required on the Database Machine

There is no code or change required on the database server ID. Control Center, with the UARCSUB and UARCCAN execs, will in effect TELL the database how to start the user archive/backup process. The service machine tells the database server with the CP SEND command. CP SEND is a SCIF command (see UARCSUB and UARCCAN templates for examples). Deviating from this format can lead to several problems, so be advised to follow the format in the UARCSUB and UARCCAN templates.

Steps Required on the Support Machine

The following is required for the Support Machine:

1. Install a Control Center support machine (C5648A70 EXEC):

   The support machine runs Control Center code, but is not responsible for any databases using the SCIF facility. After installation, you will have a disconnected service machine that can be used to perform User archives, Automated DBSPACE Maintenance, Table Reorganization and Redefinition, DBSPACE Reorganization, and other DB or non-DB-related jobs.

2. Update the SQLMSTR PROFILE file:

   Give the database machine an authority level of 4. (The Backup Product Server (the server controlling user archive) only needs an authority level of 1. It should already be there from the USER ALL 3 statement. Check to be sure.)

3. UARCHIVE Statement:

   This line must be added to the SQLMSTR PROFILE file:

   ```
   MESSAGE UARCHIVE 4 ( EXEC SPBSTART .US .NO .MS
   ```

   This MESSAGE line *must* be added before any other MESSAGE-type lines, *not at the bottom* of the profile. The UACHIVE word matches the word specified in the UARCSUB and UARCCAN execs. The specific lines from the UARCSUB exec are shown below:

   ```
   if uarchive
     then 'CP SEND' dbname 'TELL support_id UARCHIVE START BACKUP'
     else 'CP SEND' dbname 'TELL support_id UARCHIVE START RESTORE'
   ```

   The first word of the TELL command (in this case UARCHIVE) must match the word added to the support machine's SQLMSTR PROFILE. Note also that the word `support_id` in the code above should be replaced with the ID of your support machine.

4. READER Statement:

   If the backup product sends a file to the support machine (as is the case with VMBACKUP), then in the SQLMSTR PROFILE, you will need to add this line:

   ```
   READER FROM userid AT nodeid 1 ( EXEC SPBEND .US .NO .SP .FN .FT
   ```

   Replace `userid` and `nodeid` with appropriate values for your backup product machine (VMBACKUP AT VMSYSTM1). For example, when VMBACKUP sends the support machine a file, then the SPBEND routine will be performed. Be sure to add this READER line before any other READER-type lines in the support machine's SQLMSTR PROFILE. Otherwise, the other READER-type lines will be invoked because the sequence of processing is from top to bottom.

5. SPBSTART and SPBEND (optionally SPBANLZ exec):

Make the SPBSTART and SPBEND execs available to the support machine through links or by placing code on the A-disk. Samples of the SPBSTART and SPBEND execs have been included with Control Center on the code disk. An optional exec, SPBANLZ $EXEC, has also been provided. It is used to do automatic verification based on a scan of the output report provided by VMBACKUP, and is included as a sample. The support machine will require read access to the database's code disk in order to access the SQLFDEF file.

6. VMBACKUP TEMPLATE:

   If the backup tool is VMBACKUP, set up the VMBACKUP TEMPLATE file with the name of the submit and restore VMBACKUP template names for each database machine (see Figure 250 on page 521). This file is read by the support machine to obtain a template name for a given database. Template names must match the template names set up in VMBACKUP.

7. VMBACKUP Authority:

   The support machine will need the authority within VMBACKUP to submit and cancel VMBACKUP jobs.

Steps Required on the BACKUP Product Machine

If the backup product is VMBACKUP, the following is required for the VMBACKUP Machine:

1. Authorize:

   Give the support machine the authority to submit and cancel backup jobs.

2. Templates:

   Create backup and restore templates for each database machine. The names of these templates should match the names specified in the VMBACKUP TEMPLATE file on the support machine's A-disk.

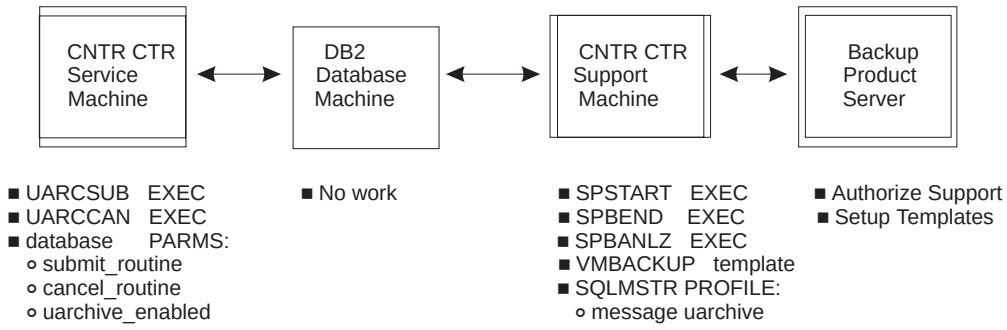
3. Messaging:

   Try to minimize the messaging that the support machine receives from VMBACKUP. The support machine should only receive messages and files for those backup/restore jobs that it submits, *not* for all jobs run by VMBACKUP.

Other backup products will need similar steps to be taken.

Summary Diagram of Control Center User Archive Architecture

The following diagram is a summary of the components of the example user archive presented in this appendix:



Sample Code Supplied with the Product

The following sample execs support the logic explained above in the suggested architecture of a user archive driven by Control Center using VMBACKUP as the tool:

Disclaimer

The following execs (and template) are provided as sample code only and are not part of the official product; thus, they are not supported.

UARCSUB $EXEC - Resides on the Service Machine

The submit routine is used when the Submit_routine field in the PARMS file has an entry with the name of the exec and a DBA submits a SQLEND UARCHIVE request or a database CANCEL request, or when an internal function causes a database to take an archive and the Uarchive_enabled field is set to "Y". Note that the exec does a CP SEND to the database machine which in turn issues a TELL command to the support machine to start a backup or restore, depending on the setting of the database mode.

UARCCAN $EXEC - Resides on the Service Machine

The cancel routine is used when the Cancel_routine field in the PARMS file has an entry with the name of the exec and a DBA submits a database CANCEL request. Note that the exec does a CP SEND to the database machine which in turn issues a TELL command to the support machine to cancel a backup or restore, depending on the setting of the database mode. The actual implementation of the cancel user archive or recovery would be done by some means in the support machine.

SPBSTART $EXEC - Resides on the Support Machine

This is a sample of an exec that runs on the Control Center support machine and responds to the START BACKUP message sent by the database. Note that message UARC003I need not be in the DATABASE MESSAGES file since it is used only for informational purposes. It will be logged in the service machine's log and in either the RECOVERY or ARCHHIST log. If the message ends in W instead of I, it will also be sent to Control Center administrators.

Additional messages may be added to provide more detailed status information in the logs. It is suggested to use the UARCnnnI format when so doing.

Important:

Do not change the ID or text of messages UARC001I and UARC002E.

SPBEND $EXEC - Resides on the Support Machine

SPBEND is a sample of an exec that runs on the Control Center support machine when the console file from VMBACKUP is received after a backup is done. The filename of the console is the template name (the database in our scenario); the filetype is VMBACKUP. The primary logic used here is to scan the report for error indications, and then, if no error messages appear, to call a specially tailored exec, SPBANLZ, which compares the list of minidisks backed up to the list of minidisks in the SQLFDEF file of the database (excepting the log FILEDEFs, since the logs are not backed up). If any errors or inconsistencies are found, the UARC002E error

message is sent to the database. If no problems are found, then the UARC001I successful message is sent back.

VMBACKUP TEMPLATE - Resides on the Support Machine

This template is a file used to determine what the VMBACKUP-required template name is on the VMBACKUP machine for a backup and restore series. Each database supported by this machine would have an entry in this file which corresponds to the name of the file on the VMBACKUP machine. The file is a description of the minidisks which are to be backed up by the VMBACKUP tool. This template is *not* supplied on the code disk but is presented here as a sample template when using VMBACKUP as the backup product. The fields which must be present are those pointed to by the arrow.

```
 VMBACKUP TEMPLATE:
 /*****************************************************/
 /*          VMBACKUP TEMPLATES TABLE                 */
 /*          ------------------------                 */
 /*           USED BY SUPPORT MACHINE                 */
 /*                                                   */
 /*                                                   */
 /*                   BACKUP         RESTORE          */
 /*    DATABASE       TEMPLATE       TEMPLATE         */
 /*    --------       --------       --------         */
---->   dbname         backup_tname   restore_tname
```

Figure 250. Sample of a VMBACKUP template on the Support Machine

SPBANLZ $EXEC - Resides on the Support Machine

SPBANLZ is a sample of an exec which verifies the success of the VMBACKUP by scanning the console file from VMBACKUP and comparing the disks dumped with the disks defined in the database SQLFDEF file.

Appendix G. Command Mode Interface

Control Center provides a command mode interface allowing most Control Center tools and tool options to be invoked directly from the CMS command prompt or from within a REXX exec. In some cases, options available in command mode are not available in panel mode.

Depending on the command and the command options specified, a data entry or options selection panel may be displayed. For example, entering `SQM VERSION` at the CMS command prompt will return a message identifying, among other things, the Control Center product release and modification level, whereas entering `SQM` by itself displays the Control Center Main Menu (panel mode interface).

Issuing Control Center Commands

Issuing Control Center commands requires READ access to the Control Center code disk and, depending on the command specified, read access to the database production disk.

DBINIT: Database Access

The DBINIT command sets up access to a database by linking to the database production disk and, if necessary, performing an SQLINIT.

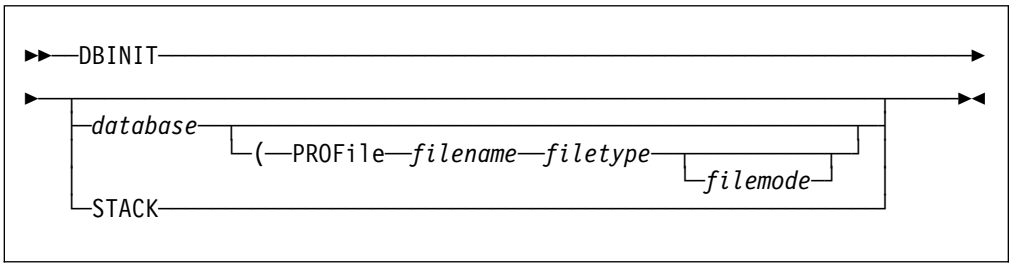
Operational Consideration: DBINIT requires access to these files:

Database Product Modules:

- SQLDBID EXEC
- ARISQLLD EXEC

Control File:

- DBINIT CONTROL is discussed in Chapter 11, “Managing the Environment” on page 113.



If DBINIT is entered without any parameters, a message indicating the currently accessed database is displayed.

database the name or nickname of the database to access.

PROFile *filename filetype filemode*
specifies an alternative file to be used instead of DBINIT CONTROL.

STACK The name of the currently accessed database is placed in the program stack.

SQLMAINT: DBSPACE Maintenance

The SQLMAINT command lets you automate dbspace maintenance. It performs UPDATE STATISTICS and SQLREORG functions on selected dbspaces. DBSPACE selection parameters include the number of days that have elapsed since the same maintenance was last performed on the dbspace. Refer to Chapter 34, "Automated DBSPACE Maintenance Tools" on page 427 for additional information.

Usage Considerations:

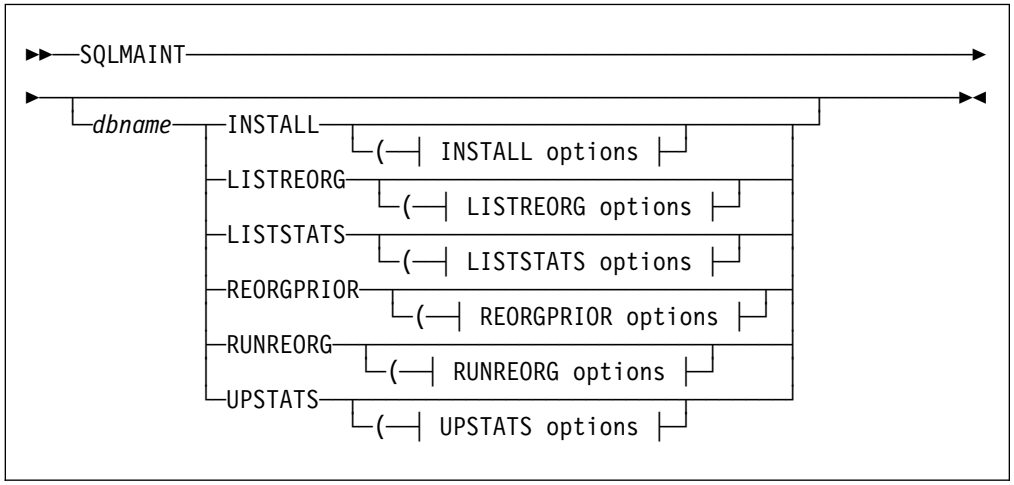
1. Authority required:

   SQLMAINT must be executed from a user ID that has DBA authority in the target database.

2. Authorization for SFS Databases:

   Upon execution of SQLREORG, the Control Center support machine must be:

   - Granted READ access on the production code directory files
   - Enrolled as a USER in the SFS filepool where the production code directory is located
   - Granted DBA on the servers where the database administration activities will be performed.



The SQLMAINT panel interface is invoked if no parameters are specified.

dbname is the database name (the dbname, *not* the database machine ID) of the database where the maintenance will be performed.

INSTALL will create a public table within the target database which is used to keep track of SQLMAINT activities.

INSTALL options
Panel Option I. See INSTALL options on page 525.

Note: SQLMAINT *must* first be installed using the INSTALL option before it is fully operational.

LISTREORG will list candidate dbspaces that require reorganization based on specified criteria.

LISTREORG options
: Panel option LR. See LISTREORG options on page 525.

LISTSTATS will list dbspaces which need to have statistics updated.

LISTSTATS options
: Panel option LS. See LISTSTATS options on page 526.

REORGPRIOR
: will reorganize dbspaces based on candidates chosen by a previous evaluation (generated by either LISTREORG or RUNREORG). The previous candidate list *must* be available on the A-disk (filename=database, filetype=SQLMAINT).

  The essential purpose of this option is to allow one execution of the candidate evaluation process using either LISTREORG or RUNREORG, with multiple subsequent reorganization runs based on the single prior evaluation. This can save considerable dbspace examination time, but should be used with *caution* since the evaluation data may quickly become invalid.

  REORGPRIOR options
  : Panel Option RP. See REORGPRIOR options on page 527.

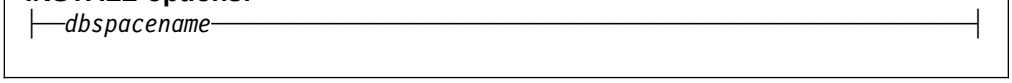
RUNREORG will reorganize candidate dbspaces that meet the specified criteria.

RUNREORG options
: Panel Option RR. See RUNREORG options on page 528.

UPSTATS will update statistics on dbspaces that meet the specified criteria.

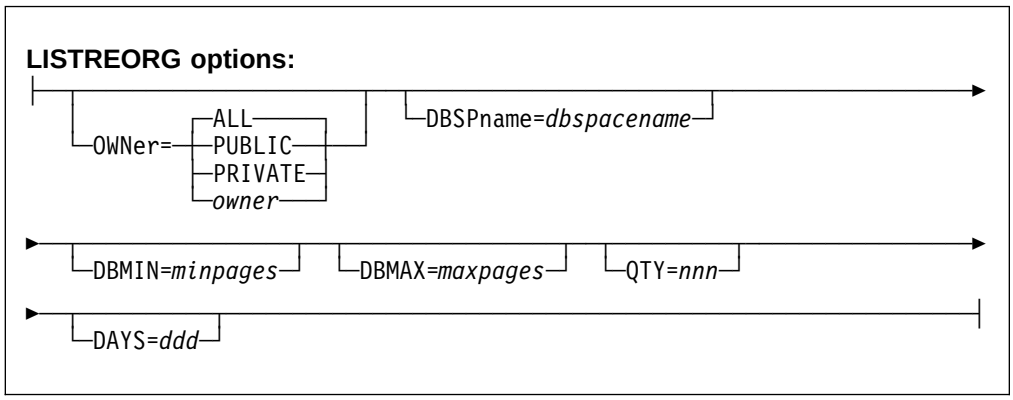
UPSTATS options
: Panel Option US. See UPSTATS options on page 529.

INSTALL options:



dbspacename is a public dbspace where the SQLMAINT table may be created. If not specified, SQLMAINT will prompt the user for the dbspace name.

Note: The SQLMAINT table *must* first be created before the SQLMAINT function is fully operational.



OWNer=*owner*
is the owner of dbspaces to be listed (PUBLIC, PRIVATE, ALL, or specific owner). ALL is the default. Owner name may include a leading or trailing percent sign (%) to indicate a wildcard match.

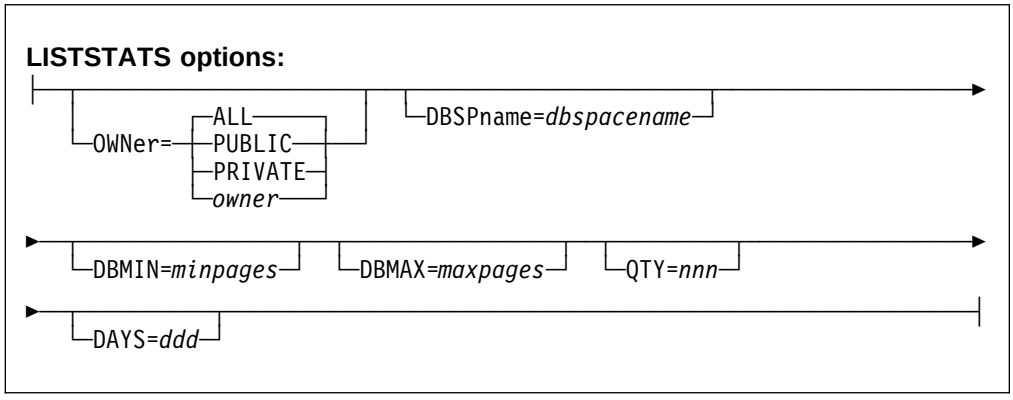
DBSPname=*dbspacename*
is a dbspace name to list. A blank entry will select ALL dbspaces. For specific dbspace names, a leading or trailing percent sign (%) can be used for wildcard matching.

DBMIN=*minpages*
is the minimum dbspace size to be included, expressed as number of pages (NPAGES).

DBMAX=*maxpages*
is the maximum dbspace size to be included, expressed as number of pages (NPAGES).

QTY=*nnn* is the number of dbspaces to be listed (default is ALL).

DAYS=*ddd* is the number of days since the previous update that must have passed for a dbspace to be listed.



OWNer=*owner*
is the owner of dbspaces to be listed (PUBLIC, PRIVATE, ALL, or specific owner). ALL is the default. Owner name may include a leading or trailing percent sign (%) to indicate a wildcard match.

DBSPname=*dbspacename*
is a dbspace name to list. A blank entry will select ALL dbspaces. For specific dbspace names, a leading or trailing percent sign (%) can be used for wildcard matching.

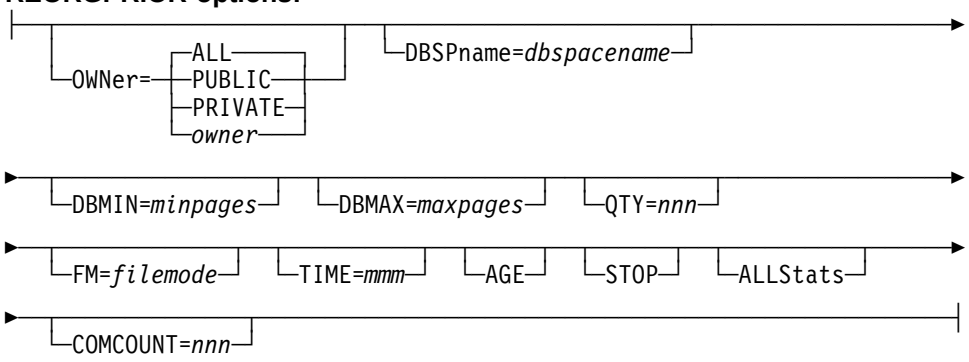
DBMIN=*minpages*
is the minimum dbspace size to be included, expressed as number of pages (NPAGES).

DBMAX=*maxpages*
is the maximum dbspace size to be included, expressed as number of pages (NPAGES).

QTY=*nnn* is the number of dbspaces to be listed (default is ALL).

DAYS=*ddd* is the number of days since the previous update that must have passed for a dbspace to be listed.

REORGPRIOR options:



OWNer=*owner*
is the owner of dbspaces to be reorganized (PUBLIC, PRIVATE, ALL, or specific owner). ALL is the default. This value must match the value provided for the previous LISTREORG or RUNREORG job, or it may be a valid subset of the previous candidate list.

DBSPname=*dbspacename*
is a dbspace name to reorganize. A blank entry will select ALL dbspaces. This value must match the value provided for the previous LISTREORG or RUNREORG job, or it may be a valid subset of the previous candidate list.

DBMIN=*minpages*
is the minimum dbspace size to be included, expressed as number of pages (NPAGES).

DBMAX=*maxpages*
is the maximum DBSPACE size to be included, expressed as number of pages (NPAGES).

QTY=*nnn* is the number of dbspaces to be reorganized (default is ALL).

FM=*filemode* is an optional filemode of a disk linked in WRITE mode to be used for reorganizations. The default is to use a temporary disk.

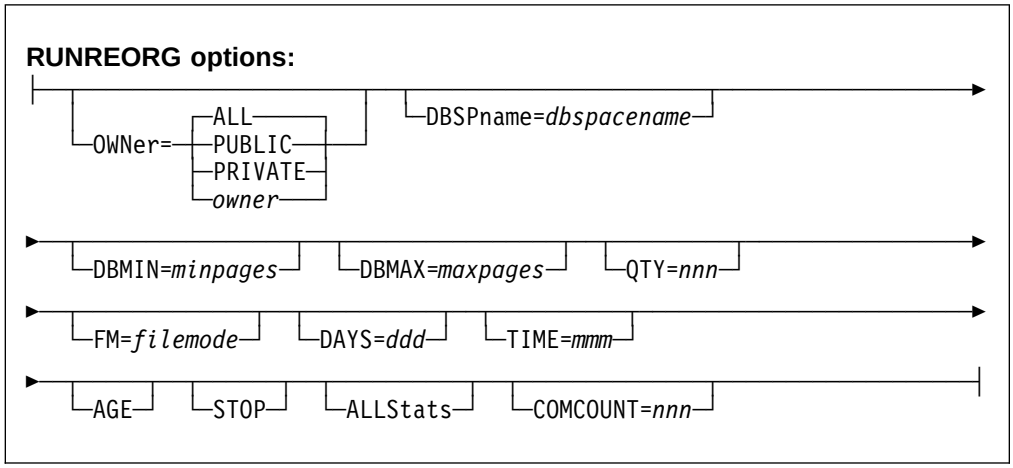
TIME=*mmm* is the maximum number of minutes that will be allowed to complete the reorganizations. If all selected dbspaces cannot be completed within the specified minutes, SQLMAINT will complete as many as possible.

AGE is an optional selection sequence for reorganizations. The default sequence is by weight.

STOP indicates that SQLMAINT should immediately terminate if a reorganization fails on any dbspace, other than failures caused by insufficient external workspace.

ALLSTATS is the keyword to use when it is desired to have SQLREORG issue an UPDATE ALL STATISTICS command in the DDL instead of the default UPDATE STATISTICS.

COMCOUNT=*nnn*
is the COMMITCOUNT value desired in RELOAD DDL.



OWNer=*owner*
is the owner of dbspaces to be listed (PUBLIC, PRIVATE, ALL, or specific owner). ALL is the default. Owner name may include a leading or trailing percent sign (%) to indicate a wildcard match.

DBSPname=*dbspacename*
is a dbspace name to list. A blank entry will select ALL dbspaces. For specific dbspace names, a leading or trailing percent sign (%) can be used for wildcard matching.

DBMIN=*minpages*
is the minimum dbspace size to be included, expressed as number of pages (NPAGES).

DBMAX=*maxpages*
is the maximum dbspace size to be included, expressed as number of pages (NPAGES).

QTY=*nnn* is the number of dbspaces to be reorganized (default is ALL).

FM=*filemode* is an optional filemode of a disk linked in WRITE mode to be used for reorganizations. The default is to use a temporary disk.

DAYS=*ddd* is the number of days since the previous update that must have passed for a dbspace to be reorganized.

TIME=*mmm* is the maximum number of minutes that will be allowed to complete the reorganizations. If all selected dbspaces cannot be completed within the specified minutes, SQLMAINT will complete as many as possible.

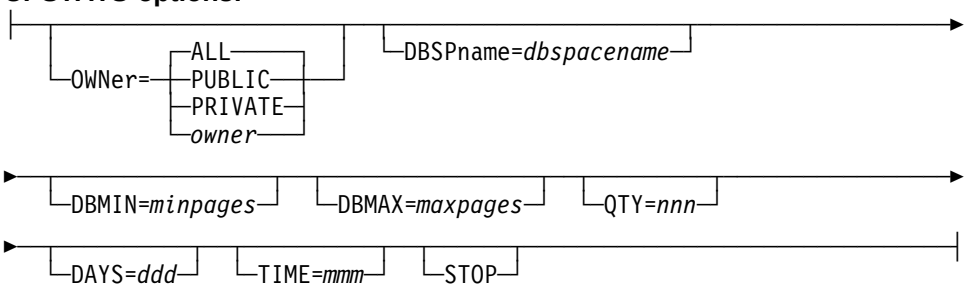
AGE is an optional selection sequence for reorganizations. The default sequence is by weight.

STOP indicates that SQLMAINT should immediately terminate if a reorganization fails on any dbspace, other than failures cause by insufficient external work space.

ALLSTATS is the keyword to use when it is desired to have SQLREORG issue an UPDATE ALL STATISTICS command in the DDL instead of the default UPDATE STATISTICS.

COMCOUNT=*nnn*
is the COMMITCOUNT value desired in RELOAD DDL.

UPSTATS options:



OWNer=*owner*
is the owner of dbspaces to be listed (PUBLIC, PRIVATE, ALL, or specific owner). ALL is the default. Owner name may include a leading or trailing percent sign (%) to indicate a wildcard match.

DBSPname=*dbspacename*
is a dbspace name to list. A blank entry will select ALL dbspaces. For specific dbspace names, a leading or trailing percent sign (%) can be used for wildcard matching.

DBMIN=*minpages*
is the minimum dbspace size to be included, expressed as number of pages (NPAGES).

DBMAX=*maxpages*
is the maximum dbspace size to be included, expressed as number of pages (NPAGES).

QTY=*nnn* is the number of dbspaces to be updated (default is ALL).

DAYS=*ddd* is the number of days since the previous update that must have passed for a dbspace to be updated.

TIME=*mmm* is the maximum number of minutes that will be allowed to complete the UPDATE STATISTICS. If all selected dbspaces cannot be completed within the specified minutes, SQLMAINT will complete as many as possible.

STOP indicates that SQLMAINT should immediately terminate if the UPDATE STATISTICS command fails on any DBSPACE.

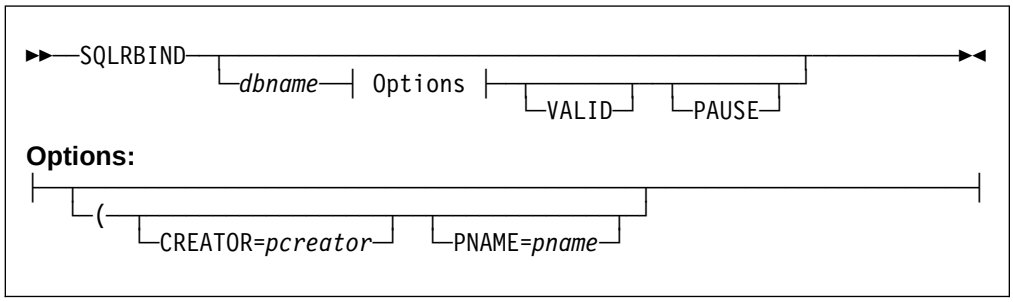
SQLRBIND: Rebind Package

The SQLRBIND command lets you selectively rebind packages within the database. SQLBIND uses the REBIND command of the Database Services Utility (DBSU). Refer to Chapter 35, “Rebind Package Tool” on page 449 for additional information.

Usage Considerations:

1. Authority required:

   SQLRBIND must be executed from a user ID that has DBA authority in the target database.



The SQLRBIND panel interface is invoked if no parameters are specified.

dbname is the database name (the dbname, *not* the database machine ID) of the database where the maintenance will be performed.

CREATOR=*pcreator*
will only REBIND packages that were created by the specified pcreator. A leading or trailing percent sign (%) may be used for wildcard matching.

PNAME=*pname*
will only REBIND packages with the specified pname. A leading or trailing percent sign (%) may be used for wildcard matching.

VALID will REBIND all packages, regardless of whether they are currently invalid. The default is to REBIND only those packages that are currently invalid.

PAUSE will cause SQLRBIND to capture all REBIND commands within a Database Services Utility (DBSU) command file and stop without executing the REBIND commands.

SQLREORG: DBSPACE Reorganization

The SQLREORG command lets you reorganize a database dbspace. The specified dbspace will be unloaded, dropped, re-acquired, and reloaded. This will assure maximum efficiency and space utilization. Refer to “Multiple User Mode DBSPACE Reorganization Tool” on page 373 for additional information.

Usage Considerations:

1. Authority required:

   You *must* have DB2 Server for VM Database Administrator authority for the database specified.

2. Processing:

   SQLREORG will acquire a temporary disk for the unloaded dbspace, unless an optional linked disk filemode or TAPE is specified.

   A copy of the unloaded data will also be sent to your reader to provide recovery backup in case the disk copy is lost.

3. Tape Processing:

   SQLREORG will unload the dbspace to tape if you specify the TAPE option. Prior to invoking SQLREORG, you must issue a FILEDEF command for ddname DBSFILE and have the tape mounted and attached. If your unloaded data will span more than one tape, you will also need to issue a LABELDEF command to provide for multivolume tape handling. SQLREORG executes the UNLOAD utility. For UNLOAD, always specify a record format (RECFM) of variable-length blocked spanned (VBS). For instance, an example of FILEDEF and LABELDEF commands would be:

```
FILEDEF DBSFILE TAP1 SL (RECFM VBS BLKSIZE 28672
LABELDEF DBSFILE VOLID SCRATCH
```

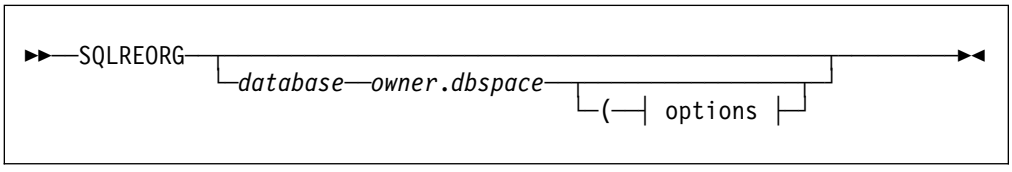
   You would then need to have a scratch tape mounted and attached as 181.

   If your unloaded data fits on a single tape, SQLREORG may be invoked without the PAUSE option and a tape rewind will automatically be performed between the UNLOAD and RELOAD processing. If your unloaded data spans more than one tape, then you *must* use the PAUSE option and issue the proper DETACH and MOUNT commands between the UNLOAD and RELOAD processing (detaching the last tape and remounting the first tape).

4. Authorization for SFS Databases:

   Upon execution of SQLREORG, the Control Center support machine must be:

   - Granted READ access on the production code directory files
   - Enrolled as a USER in the SFS filepool where the production code directory is located
   - Granted DBA on the servers where the database administration activities will be performed.

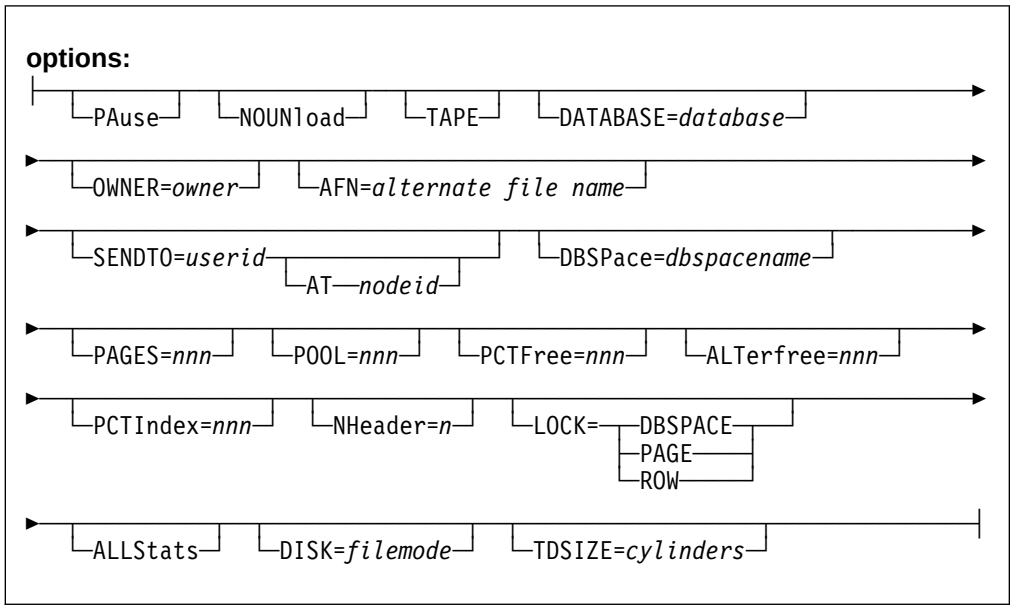


The SQLREORG panel interface is invoked if no parameters are specified.

database is the name of the applicable database.

owner.dbspace
is the database owner and name of the dbspace to be reorganized.

options See options which follow.



PAuse will perform the UNLOAD portion of the reorganization only. To continue processing, run the REXX exec that is created.

NOUNload will PAUSE before the data UNLOAD, providing only the DDL capture.

TAPE will perform the UNLOAD to TAPE instead of DISK.

DATABASE=*database*
is the database where the dbspace will be moved.

OWNER=*owner*
is the new owner of the dbspace.

AFN=*alternate file name*
is the user selected file name for SQLREORG generated files.

SENDTO=*userid* AT *nodeid*
is where a copy of the unloaded data is sent for recovery backup when a temporary disk is used. The default is your user ID.

DBSPace=*dbspacename*
is the new name for the dbspace.

PAGES=*nnn* is the new size of the dbspace.

POOL=*nnn* is the new STORPOOL.

PCTFree=*nnn* is the PCTFREE to use during the RELOAD.

ALTerfree=*nnn*
is the new PCTFREE for *after* the RELOAD.

PCTIndex=*nnn*
is the new PCTINDEX.

NHeader=*n* is the new NHEADER size.

LOCK=*xxxxx* is the new LOCK size (DBSPACE, PAGE, ROW).

ALLStats will use UPDATE ALL STATISTICS instead of UPDATE STATISTICS.

DISK=*filemode*
is the filemode of a linked disk for the data UNLOAD (defaults to using a TEMP disk).

TDSIZE=*cylinders*
is the temporary disk size (defaults to calculation based on the DBSPACE size).

SQLTABLE: Table Reorganization and Redefinition

The SQLTABLE command lets you reorganize and redefine database tables. Options can be specified to reorganize a table, copy a table, move a table, or redefine columns in a table. Column redefinition is available *only* when run in panel mode. Refer to “Overview” on page 405 for additional information.

Two Database Services Utility (DBSU) files are created by the SQLTABLE command. A DBSU file with a filetype of LOADDBSU will contain the DATALOAD statement, along with any required DROP TABLE and CREATE TABLE statements (depending on selected options). A DBSU file with a filetype of DDLDBSU will contain additional data definition language (DDL) statements associated with a table, such as CREATE INDEX, CREATE VIEW, and GRANT. The PAUSE option allows you to modify these files prior to execution. This is useful when copying tables to a new name under the same owner, allowing the user to change the index names and view names to avoid duplication of existing names.

Usage Considerations:

1. Authority required:

   The DATA option will require SELECT authority on the source table and INSERT authority on the target table.

   The DDL option requires that you own the target DBSPACE, have RESOURCE authority if the target DBSPACE is PUBLIC, or have DBA authority (required for complete capture of all VIEW and GRANT DDL capture).

2. Processing options:

   SQLTABLE will perform various functions depending on the options specified.

   Table Reorganization:
   If only the source database and table is specified, SQLTABLE will perform a table reorganization, using DATAUNLOAD to capture the data from the table and DATALOAD to restore the data to the table. If the DATA option is specified, SQLTABLE will delete all rows in the table prior to performing the DATALOAD. The DATA option requires that you have SELECT and INSERT authority to the table.

   If the DDL option is specified, SQLTABLE will capture the table create statement and all associated database objects, including indexes, views, and grants.

When data and DDL are both captured for a reorganization, you will require DBA authority for complete capture of *all* dependent objects within the database (including those of other users). The table reorganization for DDL and DATA will capture the data using DATAUNLOAD, DROP and CREATE the table, DATALOAD the data, and then recreate all dependent objects (indexes, views, and grants).

Table Migration

SQLTABLE will migrate data or table DDL between databases if the New Database (NEWDB) option is used. The source table will *not* be dropped or modified in any way.

Table Copy

SQLTABLE will copy either data or the table definition (DDL) or both if a new owner.table is specified with the NEWTABLE option. If DATA is specified, the target table must already exist and the column definitions must match the source table exactly. If DDL is specified, you must have DBA authority to capture all dependent database objects. It is important to note that if the target owner is the same as the source table owner, the index and view create names will be identical to the existing ones on the old table, so the create statements will fail unless the user takes steps to eliminate the duplication.

Table Move

SQLTABLE will move a table to a new DBSPACE if the DATA *and* DDL option are specified with the new dbspace (NEWSpace) option. Note that the old table will *only* be deleted if the new owner and table remain the same as the old owner and table.

Redefine Table

SQLTABLE will allow you to redefine a table, changing column names, adding columns, deleting columns, changing column datatypes and lengths, through the panel mode option only. Since the DATAUNLOAD/DATALOAD functions of the Database Services Utility (DBSU) are used for this process, all restrictions described in the "Data Conversion Summary" section of the *DB2 Server for VSE & VM Database Services Utility* manual apply. You should be aware of which datatypes are compatible and which ones will cause data truncation or dataload failures.

3. Tape Processing

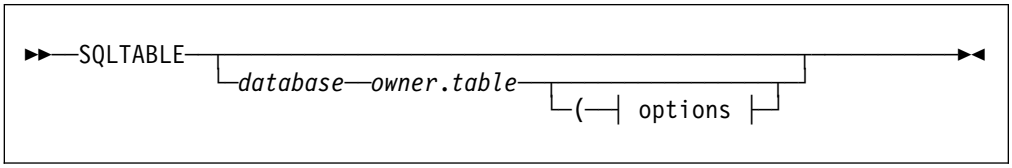
   SQLTABLE will unload the table data to tape if you specify the TAPE option. Prior to invoking SQLTABLE, you must issue a FILEDEF command for ddname DBSFILE and have the tape mounted and attached. If your unloaded data will span more than one tape, you will also need to issue a LABELDEF command to provide for multivolume tape handling.

   SQLTABLE executes the DATAUNLOAD utility. DATAUNLOAD supports all record format (RECFM) values with the exception of undefined (U). For instance, an example of a FILEDEF and LABELDEF command are given below:

```
FILEDEF DBSFILE TAP1 SL (RECFM VBS BLKSIZE 28672
LABELDEF DBSFILE VOLID SCRATCH
```

You would then need to have a scratch tape mounted and attached as 181.

If your unloaded data fits on a single tape, SQLTABLE may be invoked without the PAUSE option and a tape rewind will automatically be performed between the UNLOAD and RELOAD processing. If your unloaded data spans more than one tape, then you *must* use the PAUSE option and issue the proper DETACH and MOUNT commands between the UNLOAD and RELOAD processing (detaching the last tape and remounting the first tape).

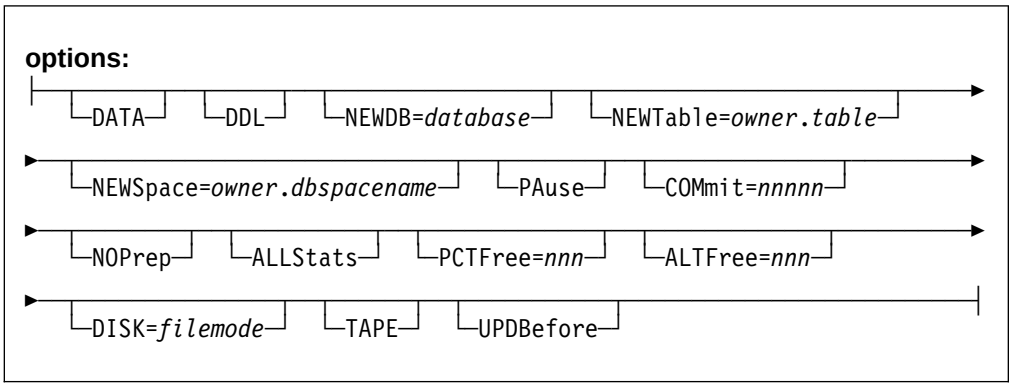


The SQLTABLE panel interface is invoked if no parameters are specified.

database is the name of the database where the table is located.

owner.table is the database owner and table name of the table that will be the target of SQLTABLE operations.

options See options below.



DATA will only unload the DATA within the specified table, and will *not* capture associated Data Definition Language (DDL) statements (CREATE TABLE, INDEX).

DDL will only capture the Data Definition Language statements associated with the specified table, and will *not* unload the data.

NEWDB=*database*
is the target database for the data or DDL. If this option is used, SQLTABLE will perform a COPY function, and the original table will remain intact.

NEWTable=*owner.table*
is the target table for the data or DDL. If this option is specified, SQLTABLE will perform a COPY function, and the original table will remain intact. If DDL is captured, the INDEX and VIEW names on the new table will be identical to the names on the old table, causing them to *fail* unless the PAUSE option is used and either the old or new names are changed.

NEWSpace=*owner.dbspacename*
is the target dbspace for the data and DDL. If this option is used without specifying a new database or owner.table name, SQLTABLE will perform a MOVE function, moving the table, indexes and views to the new dbspace.

PAuse will perform the DATAUNLOAD and DDL capture only.

COMmit=*nnnnn*
is the COMMITCOUNT for the DATALOAD.

NOPrep packages will *not* be reprepped.

ALLStats will use UPDATE ALL STATISTICS instead of UPDATE STATISTICS.

PCTFree=*nnn* is the PCTFREE to be used during the DATALOAD.

ALTFree=*nnn* is the new PCTFREE for *after* the DATALOAD.

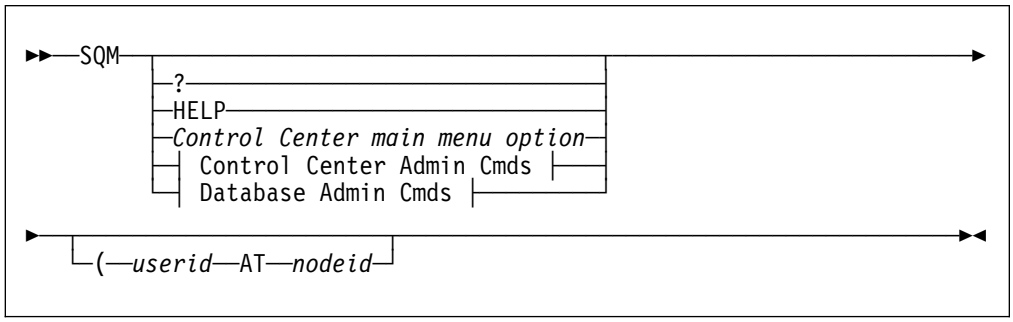
DISK=*filemode*
is the filemode of a linked disk for the DATAUNLOAD.

TAPE will perform the DATAUNLOAD to TAPE instead of DISK.

UPDBefore will UPDATE STATISTICS before calculating temp disk space.

SQM: Control Center

The SQM command is used to start Control Center in panel mode and to issue Control Center administrator commands and database administrator commands in command mode.[11]



The Control Center Main Menu is displayed if no parameters are specified. Refer to Chapter 12, “Getting Started” on page 135 for additional information.

? displays the current Control Center communication path.

HELP displays help for the SQM command.

Control Center main menu option
allows you to go directly to any option on the Control Center Main Menu when invoking Control Center in panel mode. For example, you can go directly from CMS to the ARCHIVING selection panel by specifying SQM A at the CMS command level.

[11] Control Center administrator options and database administrator options are not the same as Control Center Administration tools and Database Administration tools, respectively.

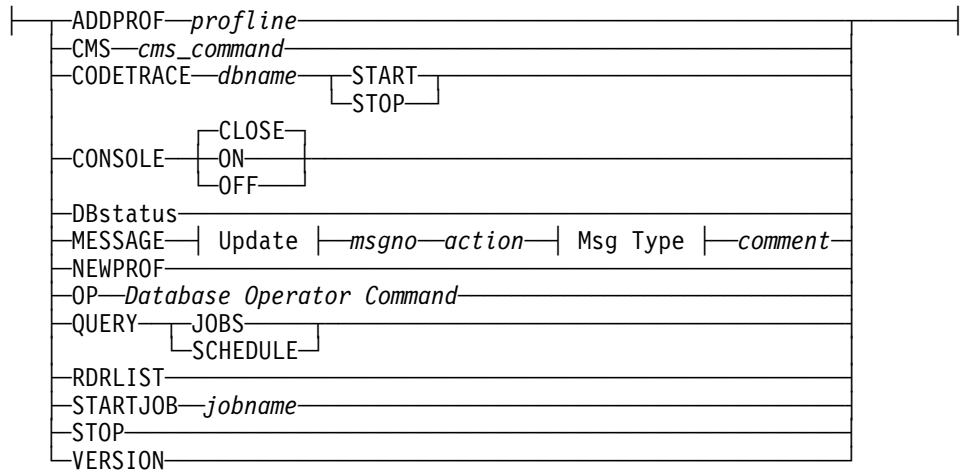
Control Center Administrator Commands
See Control Center Admin Cmds on page 537.

Database Administrator Commands
See Database Admin Cmds on page 539.

userid* AT *nodeid
is used to change communications to a different Control Center machine.

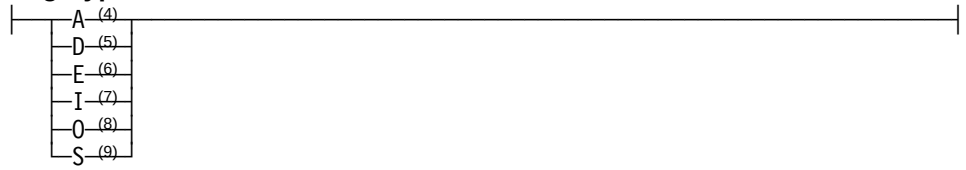
Note: This *must* be the first SQM command issued by a new user in order to establish initial communications.

Control Center Admin Cmds:



Update:

Msg Type:



Notes:
1. Add
2. Change
3. Delete
4. Administrator alert.
5. Database alert.
6. Error alert.
7. Information only.
8. Operator alert.
9. Serious error alert.

ADDPROF is used to add a line to the SQLMSTR PROFILE. The syntax must conform to SQLMSTR PROFILE requirements.

CMS is used to have Control Center execute any valid CMS command.

cmscommand is any valid CMS command.

Note: Caution must be exercised with this command. If the command causes any console input (VM READ), Control Center will be disabled.

CODETRACE is used to start or stop tracing of certain code within Control Center for a database. The results of the tracing will appear in the Control Center CONSOLE and LOG files.

dbname is the database which will have code tracing started or stopped.
START will start code tracing.
STOP will stop code tracing.

CONSOLE is used to control the spooling of the Control Center virtual machine console.

CLOSE is used to have Control Center close the current console and send it to your reader for examination. This is the default.
ON specifies that the Control Center virtual machine should begin spooling its console.
OFF specifies that the Control Center virtual machine should stop spooling its console.

DBstatus is used to display the Master Database Status panel.

MESSAGE is used to add/change/delete an action routine in the DATABASE MESSAGES file (message filtering).

Update
- A to add a message.
- C to change a message.
- D to delete a message.

msgno is the message identifier (ARI0001E).
action is the executable command that will be performed when this message is received from a database.

Message Type
- A for Administrator alert.
- D for Dbstatus alert.
- E for Error alert.
- I for Information only.
- O for Operator alert.
- S for Serious Error alert.

comment is any text which describes the message.

NEWPROF causes Control Center to temporarily halt execution and re-read the SQLMSTR PROFILE, and then restart. This must be done if any changes are made to the SQLMSTR PROFILE in order for the changes to become effective.

OP issues a database operator command to the database specified in the last executed SQLINIT or DBINIT command.

Database Operator Command
the database operator command to execute. Refer to the *DB2 Server for VSE & VM Operation* manual.

QUERY

JOBS will display the current Job Schedule for ALL databases.

QUERY SCHEDULE
will display the current SQLMSTR TIMES schedule.

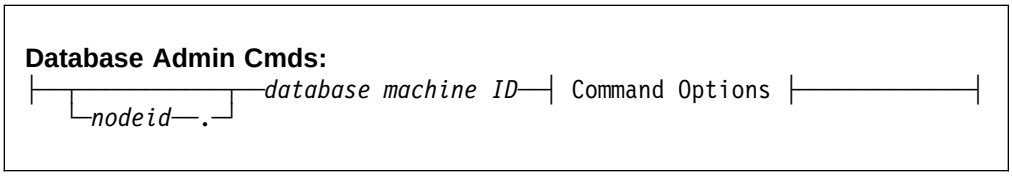
RDRLIST causes Control Center to send you a list of Control Center reader files.

STARTJOB is used to immediately start a scheduled job.

jobname the name of the Control Center job to start.

STOP causes Control Center execution to terminate.

VERSION displays Control Center code level (release, modification, service level, and last PTF applied).

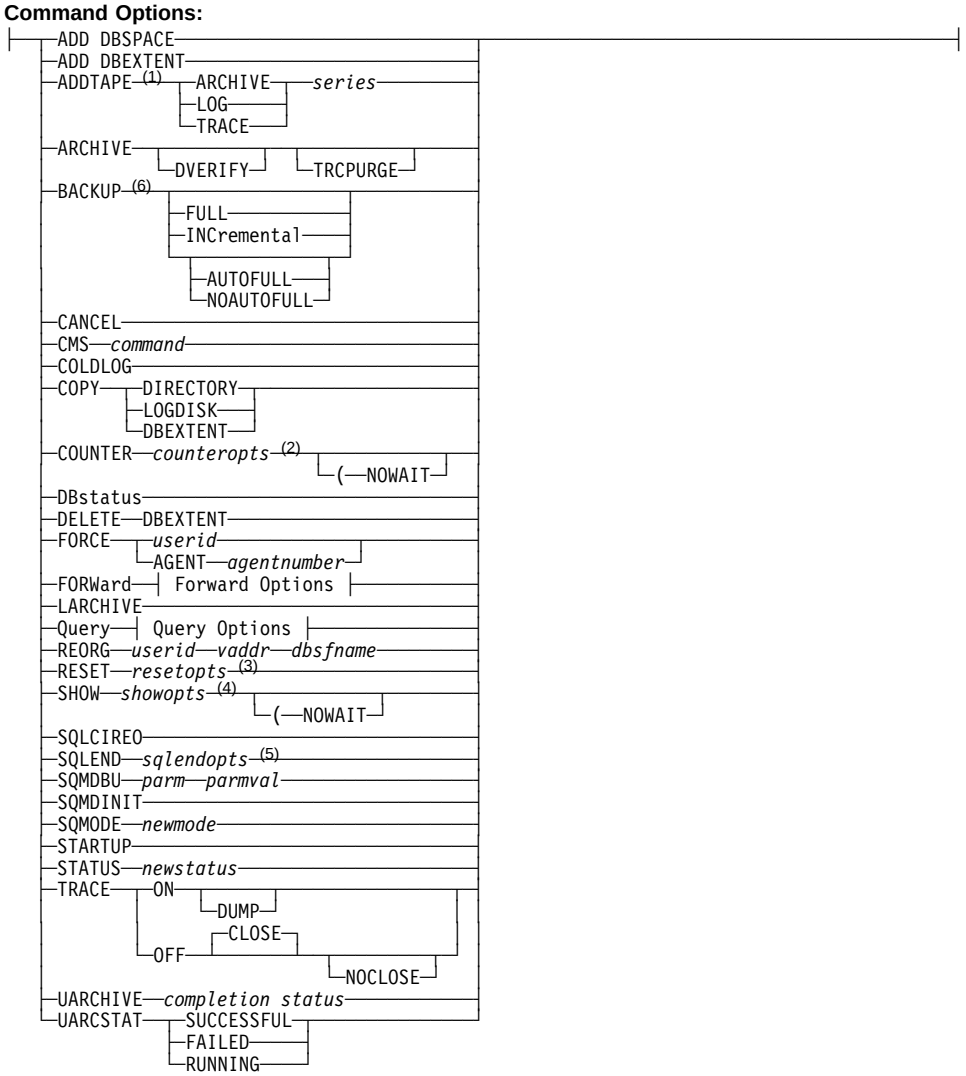


nodeid is the CPU node ID. Specify when it is necessary to distinguish between database machines with the same name on different CPUs. For example: VMSYSTM1.SQLDBA can be used to specify the SQLDBA database machine on CPU VMSYSTM1.

database machine ID
is any valid database machine ID that is controlled by Control Center. It must precede every Control Center DBA command.

Command Options
See Command Options on page 539.

Command Options:



Forward Options:

Notes:

1. Only valid when using VMTAPE.
2. Any valid COUNTER command option; refer to the *DB2 Server for VSE & VM Operation* manual.
3. Any valid RESET command option; refer to the *DB2 Server for VSE & VM Operation* manual.
4. Any valid SHOW command option; refer to the *DB2 Server for VSE & VM Operation* manual.
5. Any valid SQLEND command option; refer to the *DB2 Server for VSE & VM Operation* manual.
6. If Incremental Backup is enabled, then only Full and Incremental options are valid.

ADD DBSPACE

will initiate the Add DBSPACE tool using the database SQLADBSP file on Control Center.

ADD DBEXTENT

will initiate the Add DBEXTENTS tool using the database ADEXTENT and ADNLPOOL files on Control Center.

ADDTAPE is used to request a scratch tape to be mounted, catalogued, and added to a database TAPES file.

Usage Consideration: This function is available only with VMTAPE.

Specify the tape type:

ARCHIVE
LOG
TRACE

series must be a three-digit number that indicates to which tape series the tape should be added.

ARCHIVE is used to begin an immediate archive of a database with the database remaining up.

Options DVERIFY and TRCPURGE may be specified.

CANCEL is used to immediately terminate an active archive or recovery. It will cause a crash of the database by detaching any active tape drives, forcing an I/O error. Manual cleanup and restoration must then be performed.

CMS is a general purpose command that will pass any valid command executable under CMS to a database machine.

command is any valid command executable under CMS. If the database machine is operational, only CP commands will be accepted and they must be explicitly preceded by CP.

COLDLOG is used to immediately begin a COLDLOG (SQLLOG) on the target database. If the database is running, it will be terminated with SQLEND.

COPY will initiate the Copy/Move DBEXTENTS tools, using the database CDBEXDIR, CDBLOGEX, or CDBEXTNT file on Control Center.

Specify the type of extent to copy:

DIRECTORY
LOGDISK
DBEXTENT

COUNTER will issue the COUNTER command to the target database.

counteropts is any valid COUNTER command option. Refer to the *DB2 Server for VSE & VM Operation* manual.

NOWAIT optional parameter to issue command and immediately return control without waiting for response. The response from the database will come back as a file in your virtual reader.

DBstatus is used to display the Master Database Status panel.

DELETE DBEXTENT
will initiate the Delete DBEXTENTS tool using the database SQLADBEX file on Control Center.

FORCE will issue the database SHOW ACTIVE and FORCE commands to the target database for a specified userid/agent.

userid is an active userid in the target database.
AGENT *agnt* specifies an active agent number to be forced.

FORWard MESSages | MSGs
will cause all messages from a database virtual machine to be forwarded to a specified userid.

OFF keyword to stop forwarding database messages.
userid AT *nodeid*
the userid that will begin receiving messages forwarded from a database virtual machine.

LARCHIVE is used to immediately begin a Log Archive on the target database, with the database remaining up.

QUERY See Query Options on page 543.

REORG will initiate the Single User Mode DBSPACE Reorganization tool.

userid is the userid which owns the minidisk to which the dbspace data has been unloaded.
vaddr is the virtual address of the minidisk to which the dbspace data has been unloaded.
dbsfname is the CMS file name of the unloaded dbspace.

RESET will reset the COUNTER statistics for a given database.

resetopts are valid RESET command options (* or certain value). Refer to the *DB2 Server for VSE & VM Operation* manual.

SHOW will execute any of the database SHOW operator commands.

showopts are any valid database SHOW command parameters. Refer to the *DB2 Server for VSE & VM Operation* manual.
NOWAIT optional parameter to issue command and immediately return control without waiting for response. The response from the database will come back as a file in your virtual reader.

SQLCIREO will initiate an immediate Catalog Index reorganization for the specified database. If the database is up, it will be terminated using SQLEND.

SQLEND will immediately terminate the specified database.

sqlendopts are any valid SQLEND options. Refer to the *DB2 Server for VSE & VM Operation* manual.

SQMDBU is used to update a parameter in the database parameters file.

parmid is a valid database parameter name.
parmval is a valid database parameter value.

SQMDINIT will cause Control Center to re-read the parameters in the database parameters into memory.

SQMODE is used to change the current database status contained in the SQMODE parameter.

newmode is the new value for the SQMODE parameter.

STARTUP is used to start the database.

STATUS is used to change the current database status contained in the DBSTATUS parameter.

newstatus is the new value for the DBSTATUS parameter.

TRACE is used to start or stop TRACING within a database.

ON to start tracing. Trace option DUMP may be specified.

OFF to stop tracing. Trace option CLOSE or NOCLOSE may be specified.

UARCHIVE is used to indicate to Control Center the completion status for a database user archive.

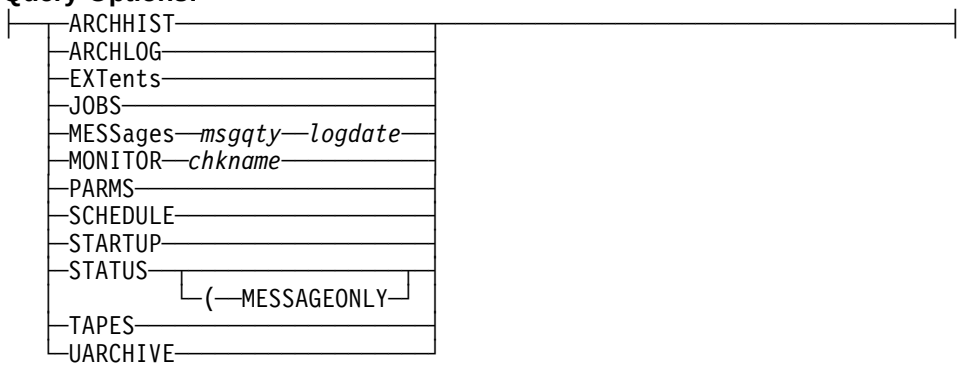
completion status
COMPLETE, SUCCESSFUL, FAILED, GOOD, BAD, ...

UARCSTAT is used to change the current database user archive status contained in the UARCSTAT parameter.

Valid values are:

SUCCESSFUL
FAILED
RUNNING

Query Options:



ARCHHIST will display the Archive History file for the specified database. This includes BACKUP, Full Backup, and Incremental Backup history information.

ARCHLOG will display the last Archive console file for the specified database.

EXTents will display the current DBEXTENT versus STORPOOL mapping for the specified database.

JOBS will display the current Job Schedule for the specified database.

MESSAGES will display the messages between the Control Center virtual machine and a specified database. This information is obtained from the Control Center LOG*jjddd* files, where *jjddd* is the Julian date.

msgqty is an integer to limit the number of message lines displayed to the most recent quantity.

logdate is the Julian date (91022) of the log file.

MONITOR will display the characteristics of a given database monitor routine, or *all* monitor routines for the database if parameter *chkname* is not is specified.

chkname is a specific monitor check name to display.

PARMS will display the database parameters for the specified database.

SCHEDULE will display the current SQLMSTR TIMES schedule for the specified database.

STARTUP will display the startup parameters used by the target database the last time it was started.

STATUS will display the current status of the target database.

MESSAGEONLY
Database status is returned as a single line message.

TAPES will display the tape catalog of the target database.

UARCHIVE will display a user archive status for target database.

Appendix H. Tape Hopper Support

In support of dark-room processing, many installations have invested in tape hoppers or stackers. The hoppers are filled with tapes and have the capability of feeding a tape drive one tape after another without any required operator intervention.

Tape Hoppers and Archiving

When using a tape hopper in support of a database archive activity, a tape drive is attached to a database virtual machine, the hopper for the drive is filled with the anticipated number of tapes required, and the drive is readied (the first tape from the stack is mounted).

When the archive (log, full, or combination of both) begins, the database will issue a prompt requesting the virtual address for the tape device to be written to. Since the tape drive has been "pre-attached" with a tape available, the response should be given immediately to start the archive. Once the response is made, the archive should run to completion, using new tapes from the hopper as required. As tapes are filled (end of volume conditions), they are replaced by the next tape in the hopper stack, and the processing continues without interruption until the archive completes.

No Tape Mounting Required

In the above scenario there are no explicit requests for tapes to be mounted. This is because the tapes and the tape drive were "pre-attached" and ready for use; therefore, no mount request to an operator console should be issued.

Control Center and Tape Hopper Support

Control Center does not explicitly support the use of tape hoppers; however, Control Center management of databases using the DMSTVI tape exit is very similar to the tape hopper support required. Therefore, to get Control Center to support tape hoppers, you must convince Control Center that the database machine is using the DMSTVI exit (see steps below).

Special Considerations

If you decide to use this type of configuration, then you should be aware of certain conditions that could cause you some problems and potentially compromise your database's recoverability.

First, to implement tape hopper support using Control Center, you will have to disable non-hopper archive capability. This could potentially be a problem when the database needs to perform an implicit archive and the tape hopper has not been readied because the archive was not anticipated.

Second, the pre-stacking of tapes relies heavily on operator interface to provide the correct tapes when filling the hopper. Without a tape management system checking for specific tapes identifiers, you could potentially overlay previously used archive tapes.

Steps for Implementing Control Center Tape Hopper Support

These steps should be followed when establishing tape hopper support:

1. Disable the database virtual machines access to any system tape exits, tape code, or tape product code disks. The database virtual machine must, in effect, have no more than native CMS tape handling capability. This is done so as to disable any tape manager code that would interfere with the tape processing.
2. Issue this command from an authorized Control Center administrator user virtual machine if your database virtual machine does not already use the DMSTVI tape exit:

   ```
   SQM CMS GLOBALV SELECT dbname SETLP TAPEEXIT DMSTVI
   ```

Steps for Turning Control Center Tape Hopper Support Off

The following steps are to be followed when disabling tape hopper support:

1. The database virtual machine must reaccess any system-provided tape exits and tape code disks.
2. Issue the following command from an authorized Control Center administrator user virtual machine if your database virtual machine does not normally use the DMSTVI tape exit;

   ```
   SQM CMS GLOBALV SELECT dbname SETLP TAPEEXIT DMSTVS
   ```

Scheduling Consideration

It may be desirable to schedule the GLOBALV SELECT commands specified above. For example, you could create a job that runs at a designated time each night to turn the DMSTVI flag on for your database, then run a second job each morning to turn it off and revert back to normal tape management support.

Appendix I. Parameter Support

Single User Mode Startup Parameters

When a database is started under Control Center in multiple user mode all startup parameters are taken from the database PARMS file. Database default values are used for those parameters not specified. When started in single user mode only the parameters listed below are taken from the PARMS file, database default values are used for all others.

- NPAGEBUF
- NDIRBUF
- NSCANS
- LOGMODE
- CHKINTVL
- SLOGCUSH
- ARCHPCT
- SOSLEVEL
- CHARNAME
- ACCOUNT
- DUMPTYPE

Override Startup Parameters Capability

Depending on the situation in which a database is being started (to take an archive; certain time of day), certain database parameters can be changed outside of the values specified in the database PARMS file. The exec which controls this is SQMOPARM EXEC. Currently, it checks primarily for certain log mode switches, and archive statuses, and overrides the NCUSERS parm to equal 1. This exec lends itself to user modification for any number of situations (specific database IDs; time of day start-ups; additional messages not currently checked for). The override will occur only when the database is brought up under the pre-defined condition. At other times, the regular database PARMS file is used.

Appendix J. Database Shutdown Procedures

DBFORCE: Database Shutdown Procedures

With the DBFORCE tool, the product now provides a more structured and controlled approach to removing active or inactive users from the database when necessary work or maintenance must be processed. The features are as follows:

- sends messages to connected users,
- issues a database checkpoint command,
- after a grace period, forces active users from the database,
- or, by using the VM FORCE (CP FORCE) command, removes inactive connections

The DBFORCE tool, found on product's code disk, includes parameters that can be customized, depending on a site's local requirements. Those parameters include:

Sleep_after_warning	time designation (sleep) between user message or warning and the actual force
User_msg	MSG or WNG sent for user alerts
SQL_force	Database force option (ROLLBACK)
VM_force	VM force command (CP FORCE)

The NOVmforce (NOV) option only removes ACTIVE users from the database (using the SQL FORCE command), while the VMForce (VMF) option is used to remove INACTIVE database users. The NOVmforce is the default option.

NOTE: DBFORCE was designed to run on a support Machine and should **NOT** be run from the primary product service ID.

DBFORCE can be executed immediately from a user machine with Control Center Database Administrator authority, or it can be scheduled to run at a later time from a support machine.

Who Can Use the DBFORCE Tool

To use DBFORCE, the product support ID will require the following authority:

- Database DBA (SQL Force or NOVmforce)
- Control Center Database Administrator
- CP FORCE (VMForce)

How to Use DBFORCE

The DBFORCE routine can be invoked directly from CMS, or it can be set up to run as a scheduled job on a product support machine.

An authorized service machine userid (DBA authority or higher) can directly issue the DBFORCE command from CMS. Either command format that follows is correct:

```
►►─DBFORCE─database_name─┬─VMForce──┬─(─SENDTO=userid──────────►◄
                         └─NOVmforce┘

►►─DBFORCE─database_name─(─┬─VMForce──┬─SENDTO=userid──────────►◄
                           └─NOVmforce┘
```

The required parameters and options are below:

database_name (required)
is the name of the database where the force will be executed

SENDTO= sends the generated logfile to the designated userid

DBFORCE Options:

```
├─┬─VMForce──┬──────────────────────────────────────────────────┤
  └─NOVmforce┘
```

VMForce removes inactive agents with the CP force command

NOVmforce removes active agents with the database force command (default option)

To execute DBFORCE from a product support machine, the following statement must be included in the SQLMSTR profile found on the 191-disk of the support machine.

MESSAGE DBFORCE 5 (EXEC DBFORCE .MS

The example in Figure 251 illustrates how DBFORCE can be scheduled to run on a disconnected support machine (SQSUPPD1).

```
:Nick.FORCEUSE
:Jobname.FORCEUSE
:Server_machine.SQSUPPD1
:Job_status.A
:Priority.5
:Next_start.19970312 12:10
:Window_end.19970312 12:30
:Schedule_interval.ONCE
:Required_database.SQDB4PD1
:Required_dbstatus.A
:Dependent_jobname.
:Dependent_jobresult.
:Average_runtime.00:02
:Last_jobstart.19970311 16:53
:Last_jobend.19970311 16:55
:Notify.DBA_USER_ID
:Execute.DBFORCE SQDB4PD1 NOV ( SENDTO=DBA_USER_ID
```

Figure 251. Scheduled DBFORCE Job

When the DBFORCE routine is executed, a logfile is generated by using the SENDTO= option. The example logfile, shown in Figure 252 on page 551, will be sent to the requesting userid (in this example, DBA_USER_ID).

```
DBFORCE SQDB4PD1 NOV ( SENDTO=DBA_USER_ID

  Executed by: SQMSUPT
      Started: 11 Mar 1997 16:53:15
-------------------------------------------------------------
The previous connection was to the SQDB4PD1 database.
  Active user SQMDEMO2 has been asked to exit the database.
-------------------------------------------------------------
Sleeping for 2 minutes...
-------------------------------------------------------------
SQL FORCE ROLLBACK DISABLE has been issued on user SQMDEMO2,
agent  1

1 user forced using "SQL FORCE" command...

-------------------------------------------------------------
0 users forced using "VM FORCE" command...
-------------------------------------------------------------
DBFORCE completed with return code =  0
```

Figure 252. DBFORCE Logfile

Bibliography

This bibliography lists publications that are referenced in this manual or that may be helpful.

Related Publications

- *DB2 Server for VSE & VM Data Restore*, SC09-2677
- *DRDA: Every Manager's Guide*, GC26-3195
- *IBM SQL Reference, Version 2, Volume 1*, SC26-8416
- *IBM SQL Reference*, SC26-8415

Other Distributed Data Publications

- *IBM Distributed Data Management (DDM) Architecture, Architecture Reference, Level 3*, SC21-9526
- *IBM Distributed Data Management (DDM) Architecture, Implementation Programmer's Guide*, SC21-9529
- *VM/Directory Maintenance Licensed Program Operation and User Guide Release 4*, SC23-0437
- *IBM Distributed Relational Database Architecture Reference*, SC26-4651
- *IBM Systems Network Architecture, Format and Protocol*
- *SNA LU 6.2 Reference: Peer Protocols*
- *Reference Manual: Architecture Logic for LU Type 6.2*
- *IBM Systems Network Architecture, Logical Unit 6.2 Reference: Peer Protocols*
- *Distributed Data Management (DDM) List of Terms*
- *IBM Distributed Data Management (DDM) Architecture, Architecture Reference, Level 3*, SC21-9526
- *IBM Distributed Data Management (DDM) Architecture, Architecture Reference, Level 3*, SC21-9526
- *IBM Distributed Data Management (DDM) Architecture, Architecture Reference, Level 3*, SC21-9526

CCSID Publications

- *Character Data Representation Architecture, Executive Overview*, GC09-2207
- *Character Data Representation Architecture Reference and Registry*, SC09-2190

C/370 Publications

- *IBM C/370 Installation and Customization Guide*, GC09-1387
- *IBM C/370 Programming Guide*, SC09-1384

Communication Server for OS/2 Publications

- *Up and Running!*,GC31-8189
- *Network Administration and Subsystem Management Guide* SC31-8181
- *Command Reference*, SC31-8183
- *Message Reference*, SC31-8185
- *Problem Determination Guide*, SC31-8186

Distributed Database Connection Services (DDCS) Publications

- *DDCS User's Guide for Common Servers*, S20H-4793
- *DDCS for OS/2 Installation and Configuration Guide* S20H-4795

VTAM Publications

- *VTAM Messages and Codes*, SC31-6493
- *VTAM Network Implementation Guide*, SC31-6494
- *VTAM Operation*, SC31-6495
- *VTAM Programming*, SC31-6496
- *VTAM Programming for LU 6.2*, SC31-6497
- *VTAM Resource Definition Reference*, SC31-6498
- *VTAM Resource Definition Samples*, SC31-6499

CSP/AD and CSP/AE Publications

- *Developing Applications*, SH20-6435
- *CSP/AD and CSP/AE Installation Planning Guide*, GH20-6764
- *Administering CSP/AD and CSP/AE on VM*, SH20-6766
- *Administering CSP/AD and CSP/AE on VSE*, SH20-6767
- *CSP/AD and CSP/AE Planning*, SH20-6770
- *Cross System Product General Information*, GH23-0500

Query Management Facility (QMF) Publications

- *QMF General Information*, GC26-4713
- *QMF VSE/ESA Setup and Usage Guide*, GG24-4196
- *Managing QMF for VSE/ESA*, SC26-3252

- *Installing QMF on VSE/ESA*, SC26-3254
- *QMF Learner's Guide*, SC26-4714
- *QMF Advanced User's Guide*, SC26-4715
- *QMF Reference*, SC26-4716
- *Installing QMF on VM*, SC26-4718
- *QMF Application Development Guide*, SC26-4722
- *QMF Messages and Codes*, SC26-4834
- *Using QMF*, SC26-8078
- *Managing QMF for VM/ESA*, SC26-8219

DL/I DOS/VS Publications

- *DL/I DOS/VS Application Programming*, SH24-5009

COBOL Publications

- *VS COBOL II Migration Guide for VSE*, GC26-3150
- *VS COBOL II Migration Guide for MVS and CMS*, GC26-3151
- *VS COBOL II General Information*, GC26-4042
- *VS COBOL II Language Reference*, GC26-4047
- *VS COBOL II Application Programming Guide*, SC26-4045
- *VS COBOL II Application Programming Debugging*, SC26-4049
- *VS COBOL II Installation and Customization for CMS* SC26-4213
- *VS COBOL II Installation and Customization for VSE* SC26-4696
- *VS COBOL II Application Programming Guide for VSE* SC26-4697

Systems Network Architecture (SNA) Publications

- *SNA Transaction Programmer's Reference Manual for LU Type 6.2*, GC30-3084
- *SNA Format and Protocol Reference: Architecture Logic for LU Type 6.2*, SC30-3269
- *SNA LU 6.2 Reference: Peer Protocols*, SC31-6808
- *SNA Synch Point Services Architecture Reference* SC31-8134

Miscellaneous Publications

- *IBM 3990 Storage Control Planning, Installation, and Storage Administration Guide*, GA32-0100
- *Dictionary of Computing*, ZC20-1699
- *APL2 Programming: Using Structured Query Language*, SH21-1056
- *ESA/390 Principles of Operation*, SA22-7201

Related Feature Publications

- *IBM Replication Guide and Reference*, S95H-0999

Index

D

E

F

G

I

J

L

M

O

P

Q

R

S

T

V

W

Communicating Your Comments to IBM

DB2® Server for VSE & VM
Control Center Installation and
Operations Guide for VM
Version 6 Release 1

Publication No. GC09-2679-00

If there is something you like—or dislike—about this book, please let us know. You can use one of the methods listed below to send your comments to IBM. If you want a reply, include your name, address, and telephone number. If you are communicating electronically, include the book title, publication number, page number, or topic you are commenting on.

The comments you send should only pertain to the information in this book and its presentation. To request additional publications or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give it to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:
  - United States and Canada: 416-448-6161
  - Other countries: (+1)-416-448-6161
- If you prefer to send comments electronically, use the network ID listed below. Be sure to include your entire network address if you wish a reply.
  - Internet: torrcf@ca.ibm.com
  - IBMLink: toribm(torrcf)
  - IBM/PROFS: torolab4(torrcf)
  - IBMMAIL: ibmmail(caibmwt9)

Readers' Comments — We'd Like to Hear from You

DB2® Server for VSE & VM
Control Center Installation and
Operations Guide for VM
Version 6 Release 1

Publication No. GC09-2679-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	□	□	□	□	□

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? □ Yes □ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name _____ Address _____

Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You
GC09-2679-00

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM Canada Ltd. Laboratory
Information Development
2G/345/1150/TOR
1150 EGLINTON AVENUE EAST
NORTH YORK ONTARIO CANADA M3C 1H7

Fold and Tape | **Please do not staple** | Fold and Tape

GC09-2679-00

Cut or Fold
Along Line

File Number: S370/4300-50
Program Number: 5648-A70

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC09-2679-00

Spine information:

DB2® Server for VSE & VM

Control Center Installation and Operations Guide for VM

Version 6 Release 1