DB2® Server for VSE & VM

Control Center Installation and Operations Guide for VSE

Version 6 Release 1

GC09-2678-00

DB2® Server for VSE & VM

Control Center Installation and Operations Guide for VSE

Version 6 Release 1

GC09-2678-00

Note!

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

This book is also provided as an online book that can be viewed with the IBM® BookManager® READ and IBM Library Reader™ licensed programs.

First Edition (December 1998)

This edition, GC09-2678, 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 the 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 the 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 vii
Trademarks vii

About This Manual ix
Who Should Use This Manual ix
Conventions Used in This Manual ix
Organization of This Manual ix
Prerequisite IBM Publications x

Summary of Changes for DB2 Version 6 Release 1 xi
Enhancements, New Functions, and New Capabilities xi
DRDA® RUOW Application Requestor for VSE (Online) xi
Stored Procedures xi
TCP/IP Support for DB2 Server for VM . . xii
New Code Page and Euro Symbol Code Page Support xii
DataPropagator™ Capture xii
QMF for VM, QMF for VSE, and QMF for Windows® xiii
RDS Above the Line xiii
Combining of NLS Feature Installation Tapes with Base Product Installation Tape xiii
Control Center Feature xiv
Data Restore Feature xiv
DB2 REXX SQL Feature xiv
Reliability, Availability, and Serviceability Improvements xiv
Migration Considerations xv
Library Enhancements xv

Chapter 1. Introduction 1
Product Benefits 1
Access Control 1
Single User Mode Parameters 1
Multiple Database Support 2
Database Operator Command Interface . . 2
DBSPACE Reorganization 2
DBSPACE Analysis 2
Group Authorization Tools 2
Monitor Tools 3
Package Tools 3
Work File Label Definition 3
CICS Report Controller Interface 3
Help Facility 3
Job Scheduling Tool 3
Query Management Facility 3
Table Utility Tool 3
Prerequisite Programs 4
About Control Center 4
Control Center's DBA ID 4
How to Invoke 5
Before You Begin Installation 5

Chapter 2. Installing Control Center 7
IBM-Supplied Installation Aids 7
Machine-Readable Material 7
Checklist for Installing Control Center for VSE 8
Installation Steps 8
Step 1: Ensure Adequate Production Library Space 8
Step 2: Install Control Center Into Your Library 9
Step 3: Allocate DASD for VSAM Files . . 10
Step 4: Define VSAM User Catalog and Datasets 10
Step 5: Load Standard Labels 11
Step 6: Prepare CICS 12
Step 7: Define Package Sublibrary 13
Step 8: Select A Language (Optional) . . 13
Step 9: Remove Unnecessary Languages (Optional) 14
Step 10: Load Error Message File 14
Step 11: Grant DBA Authority to SQLMSTR 15
Step 12: Define and Load the Help Table 16
Step 13: Define the Maintenance Tracking Table 16
Step 14: Define the Monitor Tables 17
Step 15: Define the Group Authorization Tables 18
Step 16: Load Packages into Server(s) . 18
Step 17: Define Work File Labels 19

Chapter 3. Migration to Control Center for VSE Version 6.1 23
Checklist for Migrating from VSE 1.2 23
Migrating From SQL Master V1.2 23
Step 1: Install Control Center Into the Server Library 23
Step 2: Define Package Sub-Library . . . 24
Step 3: Allocate DASD for SAM Package Work Files 24
Step 4: Define the Table Parameter File . 24
Step 5: Select A Language (Optional) . . 25
Step 6: Remove Unnecessary Languages (Optional) 25
Step 7: Prepare CICS 25
Step 8: Load Error Message File 26

Step 9: Define and Load the Help Table . 26
Step 10: Define the Monitor Tables 27
Step 11: Define the Group Authorization Tables 27
Step 12: Load Packages into Server(s) . 28
Step 13: Define Work File Labels 28
Checklist for Migrating from VSE 5.1 30
Migrating From Control Center Version 5 Release 1.0 30
Step 1: Install Control Center Into Server Library 30
Step 2: Select A Language (Optional) . . 30
Step 3: Remove Unnecessary Languages (Optional) 30
Step 4: Prepare CICS 31
Step 5: Load Error Message File 31
Step 6: Define and Load the Help Table . 31
Step 7: Load Packages into Server(s) . . 32

Chapter 4. Getting Started 33
Getting Started With Control Center 33

Chapter 5. Using the Operator Command Interface Tool 35

Chapter 6. Using the Database Monitors 37
How the Monitors Work 37
Options and Monitors Available 38
Monitor Thresholds and VSE Console Messages 38
Description of Monitor Options 39
Start the Monitor Kernel 39
Stop the Monitor Kernel 40
List Monitors 40
Add a Monitor 40
Modify a Monitor 40
Delete a Monitor 40
Display a Monitor 41
View Data 41
Reset Data 41
Print Report 41
Types of Monitors 41
SHOW ACTIVE 42
SHOW LOCK 42
SHOW DBEXTENT 42
SHOW LOG 42
SHOW CONNECT 42
SHOW DBSPACE 42
COUNTER * 42
Invocation 42
How To Use the Monitors 42
Adding A Monitor 43
Specifying Monitor Thresholds 45
Starting a Monitor 46
Changing a Monitor 46
Viewing Monitor Data 47
Stopping a Monitor 48

Chapter 7. Group Authorization Tool . . 49
About the Group Authorization Tool 49
ADD a Group 50
DROP a Group 50
Manage Group Objects and Users 50
Manage Privileges 51
LIST Functions 51
Using the Group Authorization Tool 51
Step 1: Define Application Groups 52
Step 2: Add Objects to the Application Group 53
Step 3: Define User Groups 54
Step 4: Add Users to the User Groups . . 55
Step 5: Grant Authorities to the User Groups 56
Special Considerations 56

Chapter 8. DBSPACE Reorganization Tool Tool 59
When To Reorganize 59
Features 60
How the DBSPACE Reorganization Tool Tool Works 61
Using the DBSPACE Reorganization Utility Screen 62
Optional Parameters 63
Using the DBSPACE REORGANIZATION Tool 65
Option 1 - GENERATE DDL 65
Option 2 - UNLOAD DBSPACE 66
Option 3 - REORGANIZE DBSPACE . . . 66
Option 4 - RELOAD DBSPACE 66
DBSPACE Reorganization Submit Screen . 66
Parameters 67
Single User Mode (SUM) DBSPACE Reorganization 69
Before You Choose Single User Mode Execution 69
Single User Mode Parameters 69
DBSPACE Reorganization Tape Support . . 70
Unloading to Tape 70
Special Considerations 70
Repetitive Scheduling 70
Failure Restart 70
Problem Analysis 71

Chapter 9. DBSPACE Analysis Tools . . 73
About the DBSPACE Analysis Tools 73
Before You Begin 73
How the DBSPACE Analysis Tools Work 73
SQLMAINT Table 74

DBSPACE Analysis Utility Screen 75
Functions 76
Selection Options 77
DBSPACE Reorganization Criteria (CRITERIA) 77
Update Statistics Analysis Tool 78
DBSPACE Reorganization Analysis Tool . . 79
DBSPACE Analysis Submit Screen 81
Job Options 82
Additional Topics 83
Initial Execution 84
Reorganization Work Space 84

Chapter 10. Package Utility 85
Introduction 85
Package Utility Functions 85
Package Function Descriptions 86
Package Migration 87
Invocation 87
Package Utility Parameters 88
Using the Package Utility 89

Chapter 11. Work File Label Definition Tool 93
About the Work File Label Definition Tool . . 93
Work File Label Definition Screen 93
How the Work File Label Definition Tool Works 94
Disk Work File Label Definition Screen . . . 94
Disk Work File Label Definition Fields . . 95
Tape Work File Definition Screen 96
Special Considerations 98

Chapter 12. CICS Report Controller Interface Tool 99
About the CICS Report Controller Interface Tool 99
A Sample CICS Report Controller Session . 99

Chapter 13. Control Center Help Facility 103
About the Help Facility 103

Chapter 14. Control Center Table Utility . 105
Invocation 106
LIST TABLES 107
List Tables Processing Flow 108
DROP TABLE 108
Drop Table Processing Flow 109
REORGANIZE TABLE 109
Reorganize Table Processing Flow 112
Table Reorganization Menu Required Parameters 113
Table Reorganization Menu Optional Parameters 113
Using the TABLE REORGANIZATION Option 115
Table Reorganization Submit Screen . . . 116
CREATE TABLE 119
Create Table Processing Flow 119
Using the Create Table Function 119
Create Table Parameters 121
UPDATE STATISTICS 124
Update Statistics Submit Screen Required Parameters 125
Update Statistics Submit Screen Optional Parameters 125

Appendix A. Installation JCL 127

Appendix B. Reorganization Job Streams 137

Appendix C. Control Center Packages . 157

Appendix D. Control Center Tool Tables 163
Maintenance Tracking Table 163
Database Monitor Tables 164
Group Authorization Tables 169

Appendix E. DBSPACE Reorganization Tool Related Files 173

Appendix F. Miscellaneous Members . . 179

Glossary 181

Bibliography 185

Index 189

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

| 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

Control Center is a set of database administration and operation support tools for IBM DB2 Server for VSE & VM Version 6.1 databases. This manual is intended for people who want to learn about the product or who are involved in its evaluation, installation, maintenance, administration, or use in a VSE/ESA™ environment.

Conventions Used in This Manual

Throughout this document and in the Control Center screen interfaces, the terms *database*, *database manager*, or *database server*, are used to refer to a DB2 Server for VSE composed of a directory, log(s), and one or more dbextents.

Unless otherwise specified, the term *Control Center* refers to Control Center Version 6 Release 1 Modification 0.

Organization of This Manual

Chapter 1, "Introduction" on page 1 introduces the Control Center product and tool set.

Chapter 2, "Installing Control Center" on page 7 presents the steps to install Control Center Version 6.1.

Chapter 3, "Migration to Control Center for VSE Version 6.1" on page 23 describes the steps to migrate a SQL Master for VSE Version 1 Release 2.0 product installation to Control Center Version 6.1.

Chapter 4, "Getting Started" on page 33 provides you with enough information to get you started using Control Center.

Chapter 5, "Using the Operator Command Interface Tool" on page 35 describes how to use the Operator Command interface to issue SHOW and COUNTER commands to any connectable DB2 Server in your VSE environment.

Chapter 6, "Using the Database Monitors" on page 37 describes how the Database Monitoring tools are used to issue specific database operator commands at specified frequencies during designated time periods.

Chapter 7, "Group Authorization Tool" on page 49 describes how this tool 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 8, "DBSPACE Reorganization Tool Tool" on page 59 describes the automated Control Center tools for reorganizing DBSPACES within a database.

Chapter 9, "DBSPACE Analysis Tools" on page 73 describes the Control Center tools for analyzing database DBSPACES and performing maintenance upon

them to improve performance. Maintenance includes DBSPACE reorganization and UPDATE STATISTICS.

Chapter 10, "Package Utility" on page 85 describes how you can automate tasks associated with packages within the database.

Chapter 11, "Work File Label Definition Tool" on page 93 describes the Work File Label Definition tools that help you define the reorganization tool's work files.

Chapter 12, "CICS Report Controller Interface Tool" on page 99 describes the Control Center interface to the CICS® Report Controller function.

Chapter 13, "Control Center Help Facility" on page 103 describes the Control Center Help Facility.

Chapter 14, "Control Center Table Utility" on page 105 describes how you can easily view a list of tables (and some of their attributes) stored in a DB2 Server for VSE database and do specific operations on it.

Appendix A, "Installation JCL" on page 127 describes the JCL provided by IBM.

Appendix B, "Reorganization Job Streams" on page 137 describes sample jobstreams provided by IBM.

Appendix C, "Control Center Packages" on page 157 describes Control Center packages provided by IBM.

Appendix D, "Control Center Tool Tables" on page 163 gives examples of the DBSU provided for the Maintenance Tracking, Database Monitor and Group Authorization tables.

Appendix E, "DBSPACE Reorganization Tool Related Files" on page 173 lists the DBSPACE Reorganization Tool related files and an explanation of their use.

Appendix F, "Miscellaneous Members" on page 179 describes miscellaneous members provided by IBM.

Prerequisite IBM Publications

This manual assumes you have reviewed and understand the IBM manuals for the related products. You should be familiar with VSE systems, VSE job control language, VSE/VSAM, the CICS system and have a working technical knowledge of System Administration and Database Administration in a DB2 Server for VSE environment.

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

First failure data capture support is now provided not only on the application server, but also the application requester. This is to corporate the DRDA RUOW Application Requester support added in this release.

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

Control Center is an IBM licensed program that works with the DB2 Server for VSE licensed program to automate many of the manual Database Administrator (DBA) functions required to support databases. It automates functions such as DBSPACE backup, migration, reorganization, and analysis. It generates the complete set of Data Definition Language (DDL) required to redefine any DBSPACE and its objects. It also keeps catalog statistics updated.

Through full support for VSE/POWER Job Scheduling, Control Center functions can be initiated immediately or scheduled to execute at any later date or time. Repetitive execution is also supported.

By automating the complex steps required to perform many DBA activities, Control Center simplifies the task of supporting databases. Now, DBA functions can be scheduled and performed automatically during periods of low system use. This improves the operational productivity of the entire system and allows these functions to be performed in a consistent and repeatable manner with a high degree of security and control.

Control Center is easily installed into DB2 Server for VSE databases. Its automated control lessens the workload of managing many databases.

The product consists of a set of online and batch programs, VSAM and SAM files, and database tables. It does not directly attach to any of your operating system or database management system code. All interfaces used are standard and documented.

Control Center provides you with a set of tools to help you manage your databases using conventional COBOL CICS and related programs. The Online Resource Adapter (ORA) provides the connection to the database servers on your VSE system, while the CICS Report Controller feature provides the interface to your VSE/POWER spool file. Using these interfaces, applications can CONNECT to your databases and build and submit batch jobstreams to maintain them.

Product Benefits

Access Control

Lets you control access to Control Center by assigning a special CICS transaction security key to its transactions. Then, include it in the CICS signon table (SNT) entries for the users to whom you want to allow access. You can also control access to the product by granting or withholding RUN authority on its packages.

Single User Mode Parameters

Such as LOGMODE, NDIRBUF, and NPAGEBUF give you the flexibility to choose whether or not to LOG transactions. Using these parameters, for instance during reorganizations, can improve performance by allowing the database to use more buffers in single user mode than it would use in multiple user mode.

Multiple Database Support

Exploits CICS Database Switching to CONNECT to, and manage, all of the databases on a VSE/ESA system.

Database Operator Command Interface

Lets you issue SHOW and COUNTER operator commands easily without having to use ISQL. It displays operator command output in a user-friendly format with full scrolling and online help.

DBSPACE Reorganization

Offers four (4) main functions:

- DDL generation
- UNLOAD DBSPACE
- RELOAD DBSPACE
- Reorganize DBSPACE

Reorganizing at the DBSPACE level helps you improve performance and eliminate wasted space. The DROP DBSPACE command is used to reduce logging and return pages to the storage pool for use elsewhere in the database. TABLES are RELOADed individually in the sequence of their clustering index.

Execution options let you move a DBSPACE to a larger DBSPACE, a different storage pool, or a different database for migration or regeneration. You can store data externally on tape or disk. You can also choose to REBIND all PACKAGES that are dependent on an object in the DBSPACE and optionally UPDATE ALL STATISTICS.

The Generate DDL option captures all of the DDL necessary to recreate a DBSPACE. DDL is placed in the VSE/POWER punch queue. From there, you can copy it into your editor and make whatever changes you want with it.

DBSPACE Analysis

Evaluates your databases using built-in DBA expertise. The DBSPACE Analysis tools build a list of DBSPACES that require maintenance (reorganization or UPDATE STATISTICS) and allow you to view it online. From the list you can select what DBSPACES you want to maintain and the DBSPACE Analysis tools build and submit the appropriate batch job.

Group Authorization Tools

Simplify the management of access to database tables, views, and packages. The Group Authorization Tool allows DBAs to issue authorizations to groups of users on groups of objects rather than one by one. Control Center stores group information in database tables and provides reports designed to make authorization administration easier.

Monitor Tools

Record database activity such as locking or log percent full and provide notification when the threshold you define has been exceeded. Monitor information is stored in tables that you can view on-line or print in a batch report.

Package Tools

Allow you to unload, reload, rebind, or view any of the packages stored in your databases. You can also use the package tools to migrate packages from one application server to another.

Work File Label Definition

Allows you to define your tape and SAM work files easily. For compatibility with tape management systems, all TLBL parameters are supported. For SAM work files, you define a set of small, medium, and large files that are used for all the DBSPACES in your database.

CICS Report Controller Interface

Provides quick and easy access to the jobs and reports you have submitted. Using the CICS Report Controller Interface, you can hold, or delete batch jobs, and browse, print, or delete reports. You can also view or change job and report characteristics.

Help Facility

Provides comprehensive Help and "How-To" information on all aspects of Control Center. A scrollable menu of Help topics is presented allowing you to select more specific Help information.

Job Scheduling Tool

Utilizes the full power and capabilities of VSE/POWER Time Event Scheduling as it builds batch jobs. For your jobs, you may specify the day and time a job is to be scheduled for processing. If you want to schedule a repetitive job, you can choose:

- Daily
- Weekly (for example, each Monday)
- Specific day of the month (for example, each first day)
- Specific day of specific months (for example, January 1, July 1).

Query Management Facility

Provides quick and easy access to QMF from the Control Center Main Menu.

Table Utility Tool

Provides quick and easy ways to list, reorganize (including unloading and reloading), create, drop, and update the statistics for tables stored in a DB2 Server for VSE database.

Prerequisite Programs

This section summarizes required program products. Unless otherwise stated, Control Center works with all subsequent versions, releases, and modification levels of the products listed in this section as well as with equivalent non-IBM products.

These are prerequisite products:

- VSE/Enterprise Systems Architecture Version 2 Release 2 (5690-VSE) or later
- DB2 Server for VSE Version 6 Release 1 (5648-158)
- VSE/REXX (that is part of VSE/Central Functions, Program Number 5686-066, and all subsequent releases)
- LE for VSE/ESA Version 1 Release 4 (5686-094)

About Control Center

The product executes as a set of CICS transactions and VSE batch jobstreams. Its transactions share the same CICS partition as your other applications. Its batch jobs can be run in any open partition.

Control Center applications use static, pre-planned SQL. The database optimizer determines the most efficient access path at compile-time, and stores it in the database as a package. This results in better performance at run-time. Its programs are pseudo-conversational which allow more transactions to run concurrently.

The CICS Report Controller is utilized to submit batch jobs from CICS. This way, potentially long-running tasks such as DBSPACE reorganizations do not adversely affect online users. A menu interface is also provided to allow users to manage jobs and reports in the VSE/POWER spool file and browse reports online.

Additional capabilities include:

- Control Center takes advantage of CICS Database Switching; this lets online users dynamically connect to different servers.
- Its batch jobs are intelligent, using return codes and conditional JCL to alter execution in the event of failure; this gives the DBA application recovery capability.
- It temporarily stores unloaded data and DDL statements in SAM datasets. For permanent storage, you may select VSAM or tape (for data). This allows you to reload data and DDL from prior DBSU unload operations.

Control Center's DBA ID

Control Center does all database work under the "SQLMSTR" ID. This ID needs DBA authority in every database it manages. Be sure you have completed the installation step that grants DBA authority to SQLMSTR.

How to Invoke

Using the Screens

You can enter the transaction ID "**SQM**" from a blank screen to access Control Center tools using the panel interface. From there, you can navigate through the product quickly and easily using ENTER and the function keys.

Using the Transaction ID (TRANSID)

If you know the transaction ID of the function you want to execute and if it supports direct invocation, you may simply enter its TRANSID. When you exit from that function, you return to a blank CICS screen. The functions that support direct invocation are:

- Main Menu (SQM)
- Group Authorization Tool (SQGA)
- Operator Commands Menu (SQOM)
- DBSPACE Reorganization (SQDR)
- DBSPACE Analysis (SQMM)
- Package Utility Tool (SQPM)
- Work File Label Definition (SQFM)
- CICS Report Controller (CEMS)
- Help Facility (SQHM)
- Table Utility (SQTU)

Before You Begin Installation

Prior to installation, verify that you have the prerequisite programs described in “Prerequisite Programs” on page 4. You should also be familiar with VSE Job Control Language, VSE/VSAM, CICS, and DB2 Server for VSE.

You will need these:

- Control Center distribution tape
- Control Center Program Directory
- This manual
- *DB2 Server for VSE Messages and Codes*
- *IBM VSE/ESA Messages & Codes*

Distribution Library:

The default delivery library for Control Center is PRD2.DB2610. The installation procedures in the next chapter describe how you can specify a different library. If you are using SQL/DS™ Version 3.5 or DB2 Server for VSE Version 5.1, change any LIBDEF statements to point to the correct library. The required installation library for SQL/DS Version 3 Release 5 is PRD2.SQL350. The required installation library for DB2 Server for VSE Version 5.1 is PRD2.DB2510.

Preventive Service Planning:

Read the *Control Center Program Directory* provided with the distribution tape and check for any program temporary fixes (PTFs) that you may need to install. If you obtained Control Center individually from IBM Software Distribution, you should contact the IBM Support Center, or use either Information/Access, or the IBMLink system (ServiceLink) for additional preventive service planning (PSP) information.

This program release will be maintained through the use of program temporary fix (PTF) tapes. An updated version or release replaces the entire program code. A PTF tape replaces the changed program code only.

Chapter 2. Installing Control Center

To install Control Center, ensure that the VSE operating system and all necessary software is already installed and running correctly. See “Prerequisite Programs” on page 4.

IBM-Supplied Installation Aids

The product tape includes job control members to help you install Control Center. These are distributed as Z-type source members in the Control Center library. The member names begin with **SQ**; load them as part of the installation process described in “Step 4: Define VSAM User Catalog and Datasets” on page 10.

You may have to change some of the job control members before submitting them for execution. These changes are discussed in the Installation Steps described in this chapter. Punch the members out from the distribution library for editing and submission. Alternatively, you can type them in manually.

For a list of the IBM-supplied installation aids distributed with Control Center, see:

- Appendix A, “Installation JCL” on page 127
- Appendix C, “Control Center Packages” on page 157
- Appendix F, “Miscellaneous Members” on page 179.

Machine-Readable Material

The format of the machine-readable product tape is as follows:

- *File 1* Copyright Records
- *File 2* History File
- *File 3* Product
- *File 4* Tape Mark
- *File 5* Tape Mark

This tape is intended for processing by the VSE Maintain System History Program (MSHP), or VSE/II. To download the Control Center product, you need:

- to know that the tape label for Control Center is DB2CC.6.1.0
- and to make sure you have a distribution tape in the correct format for your VSE system. To verify this tape, scan it by running the sample JCL shown in Figure 1 on page 8 and check that the tape format is correct.

```
* $$ JOB JNM=SQMTSCAN,CLASS=0,DISP=D,PRI=9
* $$ LST CLASS=Q
// JOB SQMTSCAN SCAN CONTROL CENTER TAPE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* PLEASE MOUNT CONTROL CENTER DISTRIBUTION TAPE AND
* ASSIGN SYS006 TO THAT TAPE DRIVE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// PAUSE PLEASE ASSIGN SYS006 NOW...
// ASSGN SYS006,5A0
// MTC REW,SYS006
// EXEC LIBR
RESTORE PRD2.DB2610.*.* TAPE=SYS006 SCAN=YES
/*
/&
* $$ EOJ
```

Figure 1. Scanning the Control Center Distribution Library (SQMTSCAN JCL)

Checklist for Installing Control Center for VSE

Notes:

- *Perform the steps in order.*
- *Page references appear in parentheses.*

Installation Steps

__ 1. Ensure Adequate Production Library Space (8)

__ 2. Install Control Center Into Your Library (9)

__ 3. Allocate DASD for VSAM Files (10)

__ 4. Define VSAM User Catalog and Datasets (10)

__ 5. Load Standard Labels (11)

__ 6. Prepare CICS (12)

__ 7. Define Package Sub Library (13)

__ 8. Select A Language (Optional) (13)

__ 9. Remove Unnecessary Languages (Optional) (14)

__ 10. Load Error Message File (14)

__ 11. Grant DBA Authority to SQLMSTR (15)

__ 12. Define and Load the Help Table (16)

__ 13. Define the Maintenance Tracking Table (16)

__ 14. Define the Monitor Tables (17)

__ 15. Define the Group Authorization Tables (18)

__ 16. Load Packages into Server(s) (18)

__ 17. Define Work File Labels (19)

Installation Steps

Steps 1 through 6 prepare your environment for Control Center, Step 7 prepares CICS for Control Center, and Steps 8 through 17 load and set up Control Center.

Step 1: Ensure Adequate Production Library Space

Control Center is designed to be installed into any production library. Before you install Control Center, make sure the library contains enough space. Table 1 on page 9 shows the space requirements on various DASD devices.

Control Center Version 6.1 requires approximately 2450 blocks of 1024 bytes for the base product's one language (for example, English) production environment. Control Center requires approximately 3800 blocks for installation, after which the space requirement can be reduced to 2450.

The distribution tape and installation process will load support for **ALL** of the languages that Control Center supports, such as English, and French, and German, and so forth. The table shows the space required for initial loading of Control Center in the Base + National Language Support column, and the space required if you delete the screen maps for all but one language in the Base column. The space requirements shown in the table include a 25% allowance for growth.

The installation process tailors Control Center to use just one language's maps, and message and help files. After installation is complete, you can delete those for the languages you are not using. JCL and procedures for selecting one specific language and deleting the unused language's maps, and message and help files, is given elsewhere in this manual.

The MSHP column shows the VSE Maintain System History Product additional space requirements. These do **not** include a 25% growth pad.

Table 1. Approximate DASD Space Requirements Including 25% Growth Pad

DASD Device Type	Space For Installation	Space For One Language	Additional VSE MSHP File Space Required
3375	16 cylinders	12 cylinders	1 track
3380	12 cylinders	8 cylinders	1 track
3390	10 cylinders	6 cylinders	1 track
9345	12 cylinders	8 cylinders	1 track
FB-512	9000 blocks	6000 blocks	47 blocks

NOTES:

1. These allocations include approximately 25% freespace to allow for growth due to maintenance and enhancement.
2. FB-512 devices include 0671, 3370, 9332, 9335, and 9336s.

Step 2: Install Control Center Into Your Library

You can load Control Center in one of two ways. You can use the VSE/Interactive Interface, Product Installation Dialog in VSE/ESA, or you can create a jobstream like the one shown in Figure 2 on page 10 and execute it.

To install the product into your library, make these changes to the job:

1. "DB2CC.6.1.0" is the actual file ID of the tape.
2. "PRD2.DB2610" is the default library name. If you want to install Control Center in a different library, replace it with the new name.
3. Update LIBDEF.PROC to include the Control Center Version 6.1 product library.
4. During execution, you will receive various messages from MSHP. See the *VSE/ESA Messages and Codes* manual to obtain more information about MSHP messages.

```
* $$ JOB JNM=SQMMSHPI,CLASS=0,DISP=D,PRI=9
* $$ LST CLASS=Q
// JOB SQMMSHPI LOAD CONTROL CENTER TAPE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* PLEASE MOUNT CONTROL CENTER DISTRIBUTION TAPE AND
* ASSIGN SYS006 TO THAT TAPE DRIVE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// PAUSE PLEASE ASSIGN SYS006 NOW...
// ASSGN SYS006,5A0
// MTC REW,SYS006
// EXEC MSHP,SIZE=1024K
INSTALL PRODUCT                    -
     FROMTAPE ID='DB2CC.6.1.0'     -
     PROD INTO=PRD2.DB2610
/*
/&
* $$ EOJ
```

Figure 2. Load Control Center Distribution Library (SQMMSHPI JCL)

Step 3: Allocate DASD for VSAM Files

In this step, you need to allocate VSAM space for SQMCAT, the Control Center VSAM catalog. Table 2, shows the approximate size of a starter catalog on various DASD devices. SQMCAT contains the following system files:

SQLMSTR.MESSAGES	Error Messages
SQLMSTR.REORG.DATA	Batch REORG Data
SQLMSTR.REORG.PARMS	DBSPACE REORG Parameters
SQLMSTR.TABLE.PARMS	TABLE REORG Parameters
SQLMSTR.WORK.FILES	Work File Labels

In addition, the catalog holds VSAM-managed SAM Data Definition Language (DDL) and data files for long-term storage and re-use. If you intend to run many REORGs or if your DBSPACES or tables are very large, you should monitor the VSAM catalog and be prepared to expand it.

Table 2. Average Starter Control Center Catalog Sizes

	3350 Cyls	3375 Cyls	3380 Cyls	3390 Cyls	9345 Cyls	FB-512 Blocks
Minimum	120	150	100	90	110	90000
Average	240	300	200	180	220	180000

Step 4: Define VSAM User Catalog and Datasets

Control Center uses VSAM to manage its own control files as well as unloaded data and DDL files. The batch jobstreams Control Center generates to perform functions like reorganization expect to find these files in a catalog with the filename "SQMCAT". Library member SQMVSAM.Z, shown in Figure 61 on page 128, contains the JCL and Access Methods Services commands needed to define an appropriate VSAM environment.

SQMVSAM defines the following:

- SQLMSTR.USER.CATALOG — User Catalog
- A VSAM data space
- A default model — Managed SAM files

- SQLMSTR.MESSAGES Error Message Repository
- SQLMSTR.REORG.PARMS DBSPACE Parameter File
- SQLMSTR.REORG.DATA Timekeeping Data
- SQLMSTR.TABLE.PARMS Table Parameter File
- SQLMSTR.WORK.FILES Work File Labels

To execute SQMVSAM:

1. Punch member SQMVSAM.Z for editing and submission.
2. Replace all occurrences of "$ $$" with "* $$".
3. Replace all occurrences of "#" with "/".
4. Replace "XXXXXX" with your VOLID.
5. Replace "YYYY" with your catalog origin.
6. Replace "ZZZZ" with your space origin.

In addition, if you prefer NOT to define "SQLMSTR.USER.CATALOG" but place the Control Center files in another existing catalog:

1. Remove the DEFINE UCAT statement.
2. Remove the DEFINE SPACE statement.
3. Remove the default model DEFINE CLUSTER statement.
4. Change the CATALOG parameter in the DEFINE CLUSTER statements to point to your catalog.

Step 5: Load Standard Labels

In order for Control Center to find its VSAM files, their labels must be added to the system standard label area. Library member SQMSTD.Z, shown in Figure 3 on page 12, contains a jobstream that will add the DLBLS to the system standard label area. SQMSTD.Z will also update the standard label procedure.

The DLBL name of the Control Center catalog must be "SQMCAT". If you have chosen NOT to define "SQLMSTR.USER.CATALOG", change the SQMCAT DLBL to point to the catalog on which the Control Center files will reside.

To complete this step:

1. Punch member SQMSTD.Z for editing and submission.
2. Replace all occurrences of "$ $$" with "* $$".
3. Replace all occurrences of "#" with "/".
4. Run this job in the background (BG) partition.

```
$ $$ JOB JNM=SQMSTD,CLASS=0,DISP=D,PRI=9
$ $$ LST CLASS=Q
// JOB SQMSTD SETUP CONTROL CENTER STANDARD LABELS
* STEP0001 ADD DLBLS TO SYSTEM STANDARD LABEL AREA
// OPTION STDLABEL=ADD
// DLBL SQMCAT,'SQLMSTR.USER.CATALOG',,VSAM
// DLBL SQMMESG,'SQLMSTR.MESSAGES',,VSAM,CAT=SQMCAT
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// DLBL SQMWORK,'SQLMSTR.WORK.FILES',,VSAM,CAT=SQMCAT
// DLBL SQMTPRM,'SQLMSTR.TABLE.PARMS',,VSAM,CAT=SQMCAT
#*
* STEP0002 UPDATE SYSTEM STANDARD LABEL PROCEDURE
// EXEC IESVCLUP,SIZE=AUTO
A SQLMSTR.USER.CATALOG                          SQMCAT
A SQLMSTR.MESSAGES                              SQMMESG SQMCAT
A SQLMSTR.REORG.PARMS                           SQMPARM SQMCAT
A SQLMSTR.REORG.DATA                            SQMRDAT SQMCAT
A SQLMSTR.WORK.FILES                            SQMWORK SQMCAT
A SQLMSTR.TABLE.PARMS                           SQMTPRM SQMCAT
#*
#&
$ $$ EOJ
```

Figure 3. Control Center Standard Label Definitions (SQMSTD.Z)

If Control Center is installed in a library other than PRD2.DB2610, then that library must be included in standard labels STEP0002 that updates the system standard label procedure. If the IBM supplied default procedure is not being used, or it is maintained manually, then the DLBL statements in STEP0001 must be added to your version.

Step 6: Prepare CICS

This step defines the resources required to support Control Center to your CICS CICS environment. There are three types of resources that must be defined:

1. Programs
2. Transactions
3. Files

Member SQMCSDUP.Z, shown in Figure 62 on page 130, contains a job to define the IBM Control Center programs and transactions to your CICS environment.

To execute this job:

1. Punch member SQMCSDUP.Z out of the distribution library.
2. Import it into ICCF or another editing facility.
3. Replace all occurrences of "$ $$" with "* $$".
4. Replace all occurrences of "#" with "/".
5. Submit the job for execution.

To define the Control Center files to your CICS environment, you must update and reassemble your File Control Table (FCT). Member SQMFCT.A, shown in Figure 63 on page 133, illustrates the required macros.

To complete this step:

1. Punch member SQMFCT.A out of the distribution library.

2. Import it into ICCF or another editing facility.
3. Imbed or include it in your FCT table.
4. Assemble your FCT.
5. Stop and restart CICS.

Step 7: Define Package Sublibrary

This step defines SQLMSTR.PACKAGE, the VSE sub-library into which Control Center will unload database packages and from which they will be reloaded. Unloaded packages are stored as "**.PKG**" members and are retained until you specifically delete them.

Member SQMLIBDF.Z in the distribution library contains a sample job to define the SQLMSTR.PACKAGE library. Example SQMLIBDF.Z is shown in Figure 64 on page 134. To complete this step:

1. Punch member SQMLIBDF.Z and edit as required.
2. Replace all occurrences of "$ $$" with "* $$".
3. Replace all occurrences of "#" with "/".
4. Replace "XXXXXX" with your VOLID.

Step 8: Select A Language (Optional)

This step replaces the default install language with the language of your choice. Member SQMRENAM.Z, shown in Figure 4 on page 14 contains the JCL required to rename the National Language Support (NLS) specific parts to the default names. To complete this step:

- Punch member SQMRENAM.Z and edit as required.
- Replace all occurrences of "$ $$" with "* $$".
- Replace all occurrences of "#" with "/".
- Make sure the CCLIB parameter in the SETPARM statement points to your Control Center Code Library.
- In the SETPARM statement, replace "A" with the one character language code, based on this table:

Language	Code
Mixed English	A
Uppercase English	U
French	F
German	G
Japanese	J
Simplified Chinese	C

```
$ $$ JOB JNM=SQMRENAM,DISP=D,CLASS=0,PRI=9
$ $$ LST CLASS=Q
// JOB SQMRENAM RENAME NLS SPECIFIC MEMBERS
// SETPARM LANG='A',CCLIB='PRD2.DB2610'
// LIBDEF *,SEARCH=&CCLIB
// EXEC REXX=SQMRENAM,SIZE=1024K,PARM='&LANG &CCLIB'
#*
#&
$ $$ EOJ
```

Figure 4. Select A Language (SQMRENAM.Z)

Note: REXX/VSE support must be active in the VSE system for this step to work. See the *VSE/REXX Reference* manual for more information.

Step 9: Remove Unnecessary Languages (Optional)

This step deletes the language-specific members associated with an unnecessary language. Member SQMDELET.Z, shown in Figure 5 contains the JCL required to delete the NLS specific parts. To complete this step:

- Punch member SQMDELET.Z and edit as required.
- Replace all occurrences of "$ $$" with "* $$".
- Replace all occurrences of "#" with "/".
- Make sure the CCLIB parameter in the SETPARM statement points to your Control Center Code Library.
- In the SETPARM statement, replace "A" with the one character language code, shown in the table in “Step 8: Select A Language (Optional)” on page 13.

```
$ $$ JOB JNM=SQMDELET,DISP=D,CLASS=0,PRI=9
$ $$ LST CLASS=Q
// JOB SQMDELET DELETE NLS SPECIFIC MEMBERS
// SETPARM LANG='A',CCLIB='PRD2.DB2610'
// LIBDEF *,SEARCH=&CCLIB
// EXEC REXX=SQMDELET,SIZE=1024K,PARM='&LANG &CCLIB'
#*
#&
$ $$ EOJ
```

Figure 5. Delete Unnecessary Languages (SQMDELET.Z)

Note: REXX/VSE support must be active in the VSE system for this step to work. See the *VSE/REXX Reference* manual for more information.

Step 10: Load Error Message File

This step loads the error message text into SQLMSTR.MESSAGES, the Control Center VSAM error message file. Member SQMLDMSG.Z, shown in Figure 6 on page 15, illustrates the JCL necessary to load the error message file. Member SQMESSGS.Z contains the error message texts.

To load the error message file:

1. Punch member SQMLDMSG.Z for editing and submission.
2. Replace all occurrences of "$ $$" with "* $$".
3. Replace all occurrences of "#" with "/".

4. Make sure the LIBDEF card points to your Control Center code library.
5. Make sure the sublibrary parameter (S=) on the SLI statement points to your database production library.
6. Make sure SQMMESG is closed to CICS (that is, not in use) when you execute this job. Otherwise, the load will fail with an Open error.

```
$ $$ JOB JNM=SQMLDMSG,CLASS=0,DISP=D,PRI=9
$ $$ LST CLASS=Q
// JOB SQMLDMSG LOAD CONTROL CENTER ERROR MESSAGES
// ASSGN SYS008,SYSRDR
// DLBL SQMMESG,'SQLMSTR.MESSAGES',,VSAM,CAT=SQMCAT,DISP=(NEW,KEEP)
// ASSGN SYS011,SYSLST
// LIBDEF *,SEARCH=PRD2.DB2610
// PAUSE OPERATOR: PLEASE BE SURE SQMMESG IS CLOSED TO CICS
// EXEC SQB03,SIZE=AUTO
$ $$ SLI MEM=SQMESSGS.Z,S=PRD2.DB2610
#*
#&
$ $$ EOJ
```

Figure 6. Load Control Center Error Messages (SQMLDMSG.Z)

You can remove the // LIBDEF statement if you have installed Control Center in another library. It is shown above only for documentation purposes.

During execution, you will receive two messages stating the total number of messages in and the total number of messages out. It is normal for these totals to differ.

Step 11: Grant DBA Authority to SQLMSTR

This step grants DBA authority to SQLMSTR, the ID under which Control Center does its work. Control Center requires DBA authority to DROP and ACQUIRE public DBSPACES and private DBSPACES for other users. Library member SQMGRANT.Z (Figure 7 on page 16) contains a multi-user mode DBSU jobstream that will grant DBA authority to SQLMSTR. The database must be up for this job to complete successfully.

To complete this step:

1. Punch member SQMGRANT.Z for editing and submission.
2. Replace all occurrences of "$ $$" with "* $$".
3. Replace all occurrences of "#" with "/".
4. Change the LIBDEF card to point to your Control Center code library if it is not PRD2.DB2610, or remove the LIBDEF card.
5. Change the database name parameter on the // EXEC ARIDBS card to point to your database if it is not DB2VSE61 .
6. Member SQMCDBA.C assumes the default password for SQLDBA. If you have changed SQLDBA's password, punch, update, and re-catalog SQMCDBA.C with the updated SQLDBA password.
7. Member SQMGDBA.C contains the statement that GRANTs DBA authority to SQLMSTR. The password it assigns must not be changed.

```
$ $$ JOB JNM=SQMGRANT,CLASS=0,DISP=D,PRI=9
$ $$ LST CLASS=Q
// JOB SQMGRANT GRANT CONTROL CENTER DBA AUTHORITY
// LIBDEF *,SEARCH=PRD2.DB2610
// ASSGN SYSLST,IGN
// EXEC ARIDBS,SIZE=AUTO,PARM='D(DB2VSE61)'
READ MEMBER SQMCDBA.C
READ MEMBER SQMGDBA.C (NOCONT
#*
#&
$ $$ EOJ
```

Figure 7. Grant Control Center DBA Authority (SQMGRANT.Z)

Step 12: Define and Load the Help Table

This step defines and loads the SQMHELP table. This table holds the information you access when you enter the Help Facility, or press F1 from a Control Center screen.

Member SQMCRHLP.Z, shown in Figure 65 on page 135, in the distribution library contains a batch DBSU job to define and load SQMHELP. SQMCRHLP acquires a 128-page public DBSPACE in storage pool 1. If you want to place SQMHELP elsewhere, you may change the STORPOOL parameter (about line 22). Since SQMHELP is not updated, you can place it in a non-recoverable storage pool.

SQMCRHLP reads input from member SQMHLPTX.Z using a * $$ SLI statement. You do not need to punch SQMHLPTX.Z into your library.

For defining and loading the SQMHELP table:

1. Punch member SQMCRHLP.Z for editing and submission.
2. Replace all occurrences of "$ $$" with "* $$".
3. Replace all occurrences of "#" with "/".
4. If necessary, change the STORPOOL parameter of the ACQUIRE command.
5. If necessary, change the LIBDEF card to point to your Control Center code library, or remove the LIBDEF card.
6. If necessary, change the sub-library parameter (S=) on the SLI statement to point to your database production library.
7. If necessary, change the database name parameter on the // EXEC ARIDBS card to point to your database.
8. Return Code 6 is acceptable if the PUBLIC.SQMHELP DBSPACE has never been previously acquired.

Step 13: Define the Maintenance Tracking Table

This step defines SQLMAINT, the maintenance table. Control Center stores maintenance statistics, dates, and elapsed times in the SQLMAINT table.

SQMCRMNT.Z (Figure 75 on page 163), in the distribution library, contains a batch DBSU job to define the SQLMAINT table. SQMCRMNT acquires a 128-page public DBSPACE in storage pool 1. If you want to place SQLMAINT in a different pool, you may change the STORPOOL parameter. SQLMAINT should be placed in a recoverable storage pool.

To complete this step:

1. Punch member SQMCRMNT.Z for editing and submission.
2. Replace all occurrences of "$ $$" with "* $$".
3. Replace all occurrences of "#" with "/".
4. If necessary, change the STORPOOL parameter of the ACQUIRE command.
5. If necessary, change the LIBDEF card to point to your Control Center code library, or remove the LIBDEF card.
6. If necessary, change the database name parameter on the // EXEC ARIDBS card to point to your database.
7. Return Code 6 is acceptable if the PUBLIC.SQLMAINT DBSPACE has never been previously acquired.

Step 14: Define the Monitor Tables

This step defines the monitor tables and indexes. Control Center stores monitor definition information and the data it collects in these tables. The indexes provide efficient access to the data.

Member SQMCRMON.Z, shown in Figure 76 on page 164, contains a batch DBSU job to define the monitor tables and indexes. SQMCRMON acquires a 128-page public DBSPACE in storage pool 1. If possible, you should change the STORPOOL parameter to specify another recoverable storage pool so that I/O contention for storage pool 1 is minimized.

You may also want to consider specifying a larger DBSPACE, if you intend to keep large amounts of monitor information. A final consideration is whether you want to put some of the monitor data tables such as SQLMSTR.SHOW_ACTIVE in their own DBSPACES. This is good design and might provide improved concurrency.

To define the monitor tables:

1. Punch out member SQMCRMON.Z
2. Replace all occurrences of "$ $$" with "* $$".
3. Replace all occurrences of "#" with "/".
4. If necessary, change the LIBDEF card to point to your Control Center code library, or remove the LIBDEF card.
5. If necessary, change the sublibrary parameter (S=) on the SLI statement to point to your database production library.
6. If necessary, change the database name parameter on the // EXEC ARIDBS card to point to your database.
7. If desired, change the PAGES parameter of the ACQUIRE command.
8. If desired, change the STORPOOL parameter of the ACQUIRE command.
9. Submit SQMCRMON.
10. Return Code 6 is acceptable if the PUBLIC.CC_MONITOR DBSPACE has never been previously acquired.

Step 15: Define the Group Authorization Tables

Before you can use the group authorization tool, you must run a DBSU job that creates five Group Authorization tables together with their indexes. These tables are in a separate public DBSPACE. SQMCRGRP.Z in the distribution library, contains a DBSU jobstream to define these tables. A free public DBSPACE, as referenced in the jobstream, must be available. See member SQMCRGRP.Z in Figure 77 on page 169.

To run SQMCRGRP:

1. Punch out member SQMCRGRP.Z
2. Replace all occurrences of "$ $$" with "* $$".
3. Replace all occurrences of "#" with "/".
4. If necessary, change the LIBDEF to point to your Control Center code library, or remove the LIBDEF card.
5. If necessary, change the database name parameter on the EXEC ARIDBS card to point to your database.
6. If necessary, change the DBSPACE NAME, PAGES, and/or STORPOOL parameters of the ACQUIRE command.
7. Submit SQMCRGRP.
8. Return Code 6 is acceptable if the PUBLIC.ADMGROUP DBSPACE has never been previously acquired.

Step 16: Load Packages into Server(s)

This step loads the Control Center packages into your databases. These are the programs that contain calls to the server. In order for Control Center to work with your database(s), you must load these packages into each database.

Library member SQMRLDPK.Z, Figure 74 on page 158, contains a multiple user mode DBSU job to RELOAD the Control Center packages. SQMRLDPK searches the production database library for the DBSU phase. If you are running SQL/DS™ 3.5, you must change the LIBDEF statements.

The // EXEC ARIDBS card specifies the server-name as "DB2VSE61". If your server-name is different, replace "DB2VSE61" with the name of your database.

SQMRLDPK reads the packages (.Q members) from the distribution library. Do not punch the .Q members for editing and submission.

To load the Control Center packages into your databases:

1. Punch member SQMRLDPK.Z for editing and submission.
2. Replace all occurrences of "$ $$" with "* $$".
3. Replace all occurrences of "#" with "/".
4. If necessary, change the LIBDEF card to point to your Control Center code library, or remove the LIBDEF card.
5. If necessary, change the database name parameter on the // EXEC ARIDBS card to point to your database.

Step 17: Define Work File Labels

This step defines your work files to Control Center. Before you start, you will need to run a VTOC for the DASD volumes on which you intend to place your SAM files.

IBM Data Interfile Transfer, Testing, and Operations Utility for ESA Release 1 for VSE/ESA (DITTO/ESA for VSE) Release 1, 5648-099, is an easy-to-use product for obtaining interactive VTOC listings.

To define your work file labels:

1. Type the TRANSID "**SQM**" from a blank CICS screen and press ENTER. You will be presented with the Control Center Main Menu. Then,
2. Select Option **4** (Work File Label Definition) and press ENTER. The Work File Label Definition Menu will be displayed.

DDL and Data Files

To define SQMDAT1, enter the FILE TYPE and FILE NUMBER as:

```
FILE TYPE         => 1 (1=DATA                    2=DDL                 )
                       (3=PACKAGES                4=MESSAGES            )
FILE NUMBER       => 1
```

Press ENTER to display the Disk Work File Label Definition screen. Then enter SERIAL-NUMBER, RELATIVE-TRACK/BLOCK, and NUMBER-OF-TRACKS/BLOCKS. Following the sample allocation provided in “Step 3: Allocate DASD for VSAM Files” on page 10, to define SQMDAT1 on SYSWK1 on 3380 DASD, the entries would be:

```
SERIAL-NUMBER                  => SYSWK1

RELATIVE-TRACK/BLOCK           => 02985_____

NUMBER-OF-TRACKS/BLOCKS        => 00375_____
```

Press ENTER to return to the Work File Label Definition Menu and look for this message: WORKFILE UPDATED SUCCESSFULLY!

Note: If the the update was not successful, an error message indicating the problem will be displayed.

Repeat the above for SQMDAT2, SQMDAT3, SQMDDL1, SQMDDL2, and SQMDDL3. You are now ready to reorganize your DBSPACES unloading to disk.

To define a tape file to Control Center, type a TLBL FILE-ID:

```
TLBL FILE-ID      => SQMHELP_________
```

Press ENTER to display the Tape Work File Label Definition screen. Then, type all, some, or none of the Tape Operands, as desired:

```
*************************     TAPE OPERANDS     **********************

FILE-SERIAL-NUMBER            => 004001
VOLUME-SEQUENCE-NUMBER        => ____
FILE-SEQUENCE-NUMBER          => ____
GENERATION-NUMBER             => ____
VERSION-NUMBER                => __
DATE                          => ________ (YYYY/DDD OR 0-9999)
DEVICE CLASS                  => 1        (1=CARTRIDGE/2=TAPE)
MODE                          => __
```

Press ENTER to return to the Work File Label Definition Menu and look for this message: WORKFILE UPDATED SUCCESSFULLY!

Package Files

You must define a package work file for each partition in which you intend to run a batch View Package jobstream. Control Center unloads the package into a work file and processes it to produce a package report. Only static partitions are supported.

If you want to run View Package jobs in background (CLASS 0), you must specify FILE TYPE 3 (PACKAGES) and FILE NUMBER 0 (CLASS 0) as follows:

```
FILE TYPE        => 3 (1=DATA                    2=DDL                )
                      (3=PACKAGES                4=MESSAGES           )
FILE NUMBER      => 0
```

Press ENTER to display the Disk Work File Label Definition screen. Then enter SERIAL-NUMBER, RELATIVE-TRACK/BLOCK, and NUMBER-OF-TRACKS/BLOCKS:

```
SERIAL-NUMBER                 => SYSWK1

RELATIVE-TRACK/BLOCK          => 03685_____

NUMBER-OF-TRACKS/BLOCKS       => 00010_____
```

Press ENTER to return to the Work File Label Definition Menu and look for this message: WORKFILE UPDATED SUCCESSFULLY!

Repeat this process for each partition into which you want to submit batch View Package jobstreams.

Package Messages File

You must define a single package messages file that will be shared by all partitions. The REXX/VSE Control Center batch Package Report job obtains report headings from this file. To define the package messages file, specify FILE TYPE 4 (MESSAGES) as shown:

```
FILE TYPE          => 4 (1=DATA                    2=DDL                 )
                        (3=PACKAGES                4=MESSAGES            )
FILE NUMBER        => _
```

Press ENTER to display the Disk Work File Label Definition screen. Then enter SERIAL-NUMBER, RELATIVE-TRACK/BLOCK, and NUMBER-OF-TRACKS/BLOCKS:

```
SERIAL-NUMBER                  => SYSWK1

RELATIVE-TRACK/BLOCK           => 03695_____

NUMBER-OF-TRACKS/BLOCKS        => 00005_____
```

Press ENTER to return to the Work File Label Definition Menu and look for this message: WORKFILE UPDATED SUCCESSFULLY!

Chapter 3. Migration to Control Center for VSE Version 6.1

This chapter describes how to migrate from SQL Master for VSE Version 1.2 and from Control Center Version 5.1 to Control Center Version 6.1. Prior to starting migration, refer to “Prerequisite Programs” on page 4.

The migration processes are similar to, but not exactly the same, as the installation process. Therefore, much of this chapter appears to be the same as the Installation chapter, but slightly different.

All servers (databases) that have been used with previous versions of SQL Master or Control Center must be migrated to Control Center Version 6.1 (that is, the installation process must be completed) before being accessed by other Control Center Version 6.1 machines.

Checklist for Migrating from VSE 1.2

Notes:

- *Perform the steps in order.*
- *Page references appear in parentheses.*

Migration Steps

__ 1. Install Control Center Into the Server Library (23)

__ 2. Define Package Sub-Library (24)

__ 3. Allocate DASD for SAM Package Work Files (24)

__ 4. Define the Table Parameter File (24)

__ 5. Select A Language (Optional) (25)

__ 6. Remove Unnecessary Languages (Optional) (25)

__ 7. Prepare CICS (25)

__ 8. Load Error Message File (26)

__ 9. Define and Load the Help Table (26)

__ 10. Define the Monitor Tables (27)

__ 11. Define the Group Authorization Tables (27)

__ 12. Load Packages into Server(s) (28)

__ 13. Define Work File Labels (28)

Migrating From SQL Master V1.2

Step 1: Install Control Center Into the Server Library

You can load Control Center in one of two ways. You can use the VSE/Interactive Interface, Product Installation Dialog in VSE/ESA, or you can create a jobstream like the sample shown in Figure 2 on page 10.

To install the product into your database library, make these changes:

1. "DB2CC.6.1.0" is the actual file ID of the tape.
2. "PRD2.DB2610" is the default library name. If you want to install Control Center in a different library, enter the new library name.
3. Update LIBDEF.PROC to include the Control Center Version 6.1 product library.

4. During execution, you will receive various messages from Maintain System History Program (MSHP). See the *VSE/ESA Messages and Codes* manual to obtain more information about MSHP messages.

Step 2: Define Package Sub-Library

This step defines SQLMSTR.PACKAGE, the VSE sub-library into which Control Center will unload database packages and from which they will be reloaded. Unloaded packages are stored as "**.PKG**" members and are retained until you specifically delete them.

Member SQMLIBDF.Z, shown in Figure 64 on page 134, in the distribution library contains a sample job to define the SQLMSTR.PACKAGE library.

To complete this step:

1. Punch member SQMLIBDF.Z and edit as required.
2. Replace all occurrences of "$ $$" with "$ $$".
3. Replace all occurrences of "#" with '/'.
4. Replace "XXXXXX" with your VOLID.

Step 3: Allocate DASD for SAM Package Work Files

In this step, you allocate SAM space to be used for temporary SAM work files that hold the unloaded package, used to produce the Package Report. A SAM work file is also used to hold package report headings. Allocate a SAM work file for each batch partition in which you intend to run View Package jobs and a single message file that is used by all partitions. Table 3 shows a sample set of SAM allocations.

Table 3. Sample Control Center SAM Space Allocations

File Name	Size	Type	3350 Trks	3375 Trks	3380 Trks	3390 Trks	9345 Trks	FB-512 Blocks
SQMPKG0		BG	10	15	10	10	10	600
SQMPMSG		MSGS	5	10	5	5	5	300
TOTAL			15	25	15	15	15	900

Step 4: Define the Table Parameter File

Control Center uses VSAM to manage its own control files as well as unloaded data and DDL files. The batch jobstreams Control Center generates to perform functions like reorganization expect to find these files in a catalog with the filename "SQMCAT". Library member SQMVSAM.Z, shown in Figure 61 on page 128, contains the JCL and Access Methods Services commands needed to define an appropriate VSAM environment.

If you are migrating from an earlier version of Control Center only one of the files defined in member SQMVSAM.Z, the Table Parameters File, is needed because the others already exist in your environment.

To execute SQMVSAM to define the Table Parameters File:

1. Punch member SQMVSAM.Z for editing and submission.

2. Delete all lines following the "EXEC IDCAMS" line through but not including the line containing "DEFINE SQMTPRM".
3. Delete all lines starting with the line containing "DEFINE SQMMESG" through but not including the line containing "# *" at the end of the file.
4. Replace all occurrences of "$ $$" with "* $$".
5. Replace all occurrences of "#" with "/".
6. Replace "XXXXXX" with your VOLID.

Step 5: Select A Language (Optional)

This step replaces the default install language with the language of your choice. The step is detailed in “Step 8: Select A Language (Optional)” on page 13.

Step 6: Remove Unnecessary Languages (Optional)

This step deletes the language specific members associated with an unnecessary language. The step is detailed in “Step 9: Remove Unnecessary Languages (Optional)” on page 14.

Step 7: Prepare CICS

This step defines the resources required to support Control Center in your CICS environment. There are three types of resources that must be defined:

1. Programs
2. Transactions
3. Files

Member SQMCSDUP.Z, shown in Figure 62 on page 130, contains a job to define the IBM Control Center programs and transactions to your CICS environment.

To execute this job:

1. Punch member SQMCSDUP.Z out of the distribution library.
2. Import it into ICCF or another editing facility.
3. Replace all occurrences of "$ $$" with "$ $$".
4. Replace all occurrences of "#" with "/".
5. Submit the job for execution.

To define the Control Center files to your CICS environment, you must update and reassemble your File Control Table (FCT). Member SQMFCT.A, shown in Figure 63 on page 133, illustrates the required macros.

To complete this step:

1. Punch member SQMFCT.A out of the distribution library.
2. Import it into ICCF or another editing facility.
3. Imbed or include it in your FCT table.
4. Assemble your FCT.
5. Stop and restart CICS.

Step 8: Load Error Message File

This step loads the error message text into SQLMSTR.MESSAGES, the Control Center VSAM error message file. Member SQMLDMSG.Z, shown in Figure 6 on page 15, contains the JCL necessary to load the error message file. Member SQMESSGS.Z contains the error message text itself.

For loading the Error Message file:

1. Punch member SQMLDMSG.Z for editing and submission.
2. Replace all occurrences of "$ $$" with "$ $$".
3. Replace all occurrences of "#" with "/".
4. Make sure the LIBDEF card points to your Control Center code library.
5. Make sure the sublibrary parameter (S=) on the SLI statement points to your database manager production library.
6. Make sure SQMMESG is closed to CICS when you execute this job. (Otherwise, the load will fail with an open error).

Step 9: Define and Load the Help Table

This step defines and loads SQMHELP, the table that holds the help information you access when you enter the Help Facility, or press F1 from the Control Center screen.

Member SQMCRHLP.Z shown in Figure 65 on page 135, in the distribution library contains a batch DBSU job to define and load SQMHELP. As delivered, SQMCRHLP acquires a 128-page public DBSPACE in storage pool 1. If you want to place SQMHELP elsewhere (many DBAs reserve storage pool 1 for SYS0001, the catalog DBSPACE), you may change the STORPOOL parameter. Since SQMHELP is not updated, you can place it in a non-recoverable storage pool.

SQMCRHLP reads input from member SQMHLPTX.Z using a "* $$ SLI" statement. You do not need to punch SQMHLPTX.Z into your library.

To load the Error Message file:

1. Punch member SQMCRHLP.Z for editing and submission.
2. Replace all occurrences of "$ $$" with "$ $$".
3. Replace all occurrences of "#" with "/".
4. If necessary, change the STORPOOL parameter of the ACQUIRE command.
5. If necessary, change the LIBDEF card to point to your Control Center code library, or remove the LIBDEF card.
6. If necessary, change the sublibrary parameter (S=) on the SLI statement to point to your database manager production library.
7. If necessary, change the database name parameter on the // EXEC ARIDBS card to point to your database manager.
8. Return Code 6 is acceptable if the PUBLIC.SQMHELP DBSPACE has never been previously acquired.

Step 10: Define the Monitor Tables

This step defines the monitor tables and indexes. Control Center stores monitor definition information and the data it collects in these tables. The indexes provide efficient access to the data.

Member SQMCRMON.Z, shown in Figure 76 on page 164, contains a batch DBSU job to define the monitor tables and indexes. As delivered, SQMCRMON acquires a 128-page public DBSPACE in storage pool 1. If possible, you should change the STORPOOL parameter to specify another recoverable storage pool so that I/O contention for storage pool 1 is minimized.

You may also want to consider specifying a larger DBSPACE, if you intend to keep large amounts of monitor information. A final consideration is whether you want to put some of the monitor data tables such as SQLMSTR.SHOW_ACTIVE in their own DBSPACES. This is good design and may improve concurrency.

To define the monitor tables:

1. Punch out member SQMCRMON.Z:
   a. Replace all occurrences of "$ $$" with "$ $$".
   b. Replace all occurrences of "#" with "/".
   c. If necessary, change the LIBDEF card to point to your Control Center code library, or remove the LIBDEF card.
   d. If necessary, change the sublibrary parameter (S=) on the SLI statement to point to your database manager production library.
   e. If necessary, change the database name parameter on the // EXEC ARIDBS card to point to your database manager.
   f. If desired, change the PAGES parameter of the ACQUIRE command.
   g. If desired, change the STORPOOL parameter of the ACQUIRE command.
2. Submit SQMCRMON.
3. Return Code 6 is acceptable if the PUBLIC.CC_MONITOR DBSPACE was not previously acquired.

Step 11: Define the Group Authorization Tables

Before you can use the Group Authorization tool, you must run a DBSU job that creates five Group Authorization tables together with their indexes. These tables are in a separate public DBSPACE. SQMCRGRP.Z in the distribution library, contains a DBSU jobstream to define these tables. A free public DBSPACE, as referenced in the jobstream, must be available.

Member SQMCRGRP.Z, is shown in Figure 77 on page 169.

To run SQMCRGRP:

1. Punch out member SQMCRGRP.Z:
   a. Replace all occurrences of "$ $$" with "$ $$".
   b. Replace all occurrences of "#" with "/".
   c. If necessary, change the LIBDEF to point to your Control Center code library, or remove the LIBDEF card.
   d. If necessary, change the database name parameter on the EXEC ARIDBS card to point to your database manager.

e. If necessary, change the DBSPACE NAME, PAGES, and/or STORPOOL parameters of the ACQUIRE command.

2. Submit SQMCRGRP.
3. Return Code 6 is acceptable if the PUBLIC.ADMGROUP DBSPACE has never been previously acquired.

Step 12: Load Packages into Server(s)

This step loads the Control Center packages into your databases. These are the programs that contain calls to the server. In order for Control Center to work with your database(s), you must load these packages into each database.

Library member SQMRLDPK.Z (Figure 74 on page 158) contains a multiple user mode DBSU job to RELOAD the Control Center packages. As delivered, SQMRLDPK searches the DB2 Server for VSE Version 6.1 library for the DBSU phase. If you are running SQL/DS 3.5, you must change the LIBDEF statements.

The // EXEC ARIDBS card specifies the server-name as "DB2VSE61". If your server-name is different, replace "DB2VSE61" with the name of your database.

SQMRLDPK reads the packages (.Q members) from the distribution library. Do not punch the .Q members for editing and submission.

To load the Control Center packages into your databases:

1. Punch member SQMRLDPK.Z for editing and submission.
2. Replace all occurrences of "$ $$" with "$ $$".
3. Replace all occurrences of "#" with "/".
4. Change the LIBDEF, if necessary, to point to your Control Center code library, or remove the LIBDEF card.
5. If necessary, change the database name parameter on the // EXEC ARIDBS card to point to your database manager.

Step 13: Define Work File Labels

This step defines your work package work files. Before you start, you will need to run a VTOC for the DASD volumes on which you intend to place your SAM package work files.

IBM Data Interfile Transfer, Testing, and Operations Utility for ESA Release 1 for VSE/ESA (DITTO/ESA for VSE Release 1, 5648-099), is an easy-to-use product for obtaining interactive VTOC listings.

To define your package work file labels, type the transid "**SQM**" from a blank CICS screen and press ENTER. You will be presented with the Control Center Main Menu. Then, select Option **4** (Work File Label Definition) and press ENTER. The Work File Label Definition Menu will be displayed.

You must define a package work file for each partition in which you intend to run a batch View Package jobstream. Control Center unloads the package into a work file and processes it to produce a package report. Only static partitions are supported.

If you want to run View Package jobs in background (CLASS 0), you must specify FILE TYPE 3 (PACKAGES) and FILE NUMBER 0 (CLASS 0) as follows:

```
FILE TYPE         => 3 (1=DATA                    2=DDL                )
                       (3=PACKAGES                4=MESSAGES           )
FILE NUMBER       => 0
```

Press ENTER to display the Disk Work File Label Definition screen. Then enter SERIAL-NUMBER, RELATIVE-TRACK/BLOCK, and NUMBER-OF-TRACKS/BLOCKS:

```
SERIAL-NUMBER                  => SYSWK1

RELATIVE-TRACK/BLOCK           => 03685_____

NUMBER-OF-TRACKS/BLOCKS        => 00010_____
```

Press ENTER to return to the Work File Label Definition Menu and look for this message: WORKFILE UPDATED SUCCESSFULLY!

Repeat this process for each partition into which you want to submit batch View Package jobstreams.

Package Messages File

You must define a package messages file. A single package message file will be shared by all partitions. The REXX/VSE Control Center batch Package Report job obtains report headings from this file. To define the package messages file, specify FILE TYPE 4 (MESSAGES) as shown:

```
FILE TYPE         => 4 (1=DATA                    2=DDL                )
                       (3=PACKAGES                4=MESSAGES           )
FILE NUMBER       => _
```

Press ENTER to display the Disk Work File Label Definition screen. Then enter SERIAL-NUMBER, RELATIVE-TRACK/BLOCK, and NUMBER-OF-TRACKS/BLOCKS:

```
SERIAL-NUMBER                  => SYSWK1

RELATIVE-TRACK/BLOCK           => 03695_____

NUMBER-OF-TRACKS/BLOCKS        => 00005_____
```

Press ENTER to return to the Work File Label Definition Menu and look for this message: WORKFILE UPDATED SUCCESSFULLY!

Checklist for Migrating from VSE 5.1

Notes:

- *Perform the steps in order.*
- *Page references appear in parentheses.*

Migration Steps

__ 1. Install Control Center Into Server Library (30)
__ 2. Select A Language (Optional) (30)
__ 3. Remove Unnecessary Languages (Optional) (30)
__ 4. Prepare CICS (31)
__ 5. Load Error Message File (31)
__ 6. Define and Load the Help Table (31)
__ 7. Load Packages into Server(s) (32)

Migrating From Control Center Version 5 Release 1.0

Step 1: Install Control Center Into Server Library

You can load Control Center into the database Distribution Library in one of two ways. You can use the VSE/Interactive Interface, Product Installation Dialog in VSE/ESA, or you can create a jobstream like the one shown in Figure 2 on page 10.

To install the product into your database library, make these changes:

1. "DB2CC.6.1.0" is the actual file ID of the tape.
2. "PRD2.DB2610" is the default library name. If you want to install Control Center in a different library, replace that with the new library name.
3. Update LIBDEF.PROC to include the Control Center product library.
4. During execution, you will receive various messages from Maintain System History Program (MSHP). Refer to the VSE/ESA Messages and Codes manual to obtain more information about MSHP messages.

Step 2: Select A Language (Optional)

This step replaces the default install language with the language of your choice. The step is detailed in “Step 8: Select A Language (Optional)” on page 13.

Step 3: Remove Unnecessary Languages (Optional)

This step deletes the language specific members associated with an unnecessary language.

Step 4: Prepare CICS

To prepare CICS for Control Center, you need to update the control tables below, and then re-start CICS so that the modifications made to the tables have taken effect.

- DFHPCT - Program Control Table (Control Center Transactions)
- DFHPPT - Processing Program Table (Control Center Programs, Maps)

Source decks containing the required entries have been provided on the Control Center distribution tape.

To complete this step:

1. Copy the source decks included on this tape into your CICS tables and reassemble them. Then Stop and re-start CICS; the new entries will have taken effect. Or,
2. Alternatively, you may add the PCT and PPT entries using the Resource Definition Online (RDO) facility. For more information on RDO, refer to the *CICS Resource Definition (Online)* manual.

Step 5: Load Error Message File

This step loads the error message text into SQLMSTR.MESSAGES, the Control Center VSAM error message file. Member SQMLDMSG.Z, shown in Figure 6 on page 15, contains the JCL necessary to load the error message file. Member SQMESSGS.Z contains the error message text itself.

To load the Error Message file:

1. Punch member SQMLDMSG.Z for editing and submission.
2. Replace all occurrences of "$ $$" with "$ $$".
3. Replace all occurrences of "#" with "/".
4. Make sure the LIBDEF card points to your Control Center code library.
5. Make sure the sublibrary parameter (S=) on the SLI statement points to your database manager production library.
6. Make sure SQMMESG is closed to CICS when you execute this job. (Otherwise, the load will fail with an open error).

Step 6: Define and Load the Help Table

This step defines and loads SQMHELP, the table that holds the help information you access when you enter the Help Facility, or press F1 from the Control Center screen.

Member SQMCRHLP.Z, shown in Figure 65 on page 135, in the distribution library contains a batch DBSU job to define and load SQMHELP. As delivered, SQMCRHLP acquires a 128-page public DBSPACE in storage pool 1. If you want to place SQMHELP elsewhere (many DBAs reserve storage pool 1 for SYS0001, the catalog DBSPACE), you may change the STORPOOL parameter. Since SQMHELP is not updated, you can place it in a non-recoverable storage pool.

SQMCRHLP reads input from member SQMHLPTX.Z using a "* $$" SLI statement. You do not need to punch SQMHLPTX.Z into your library.

To load the Help text file:

1. Punch member SQMCRHLP.Z for editing and submission.
2. Replace all occurrences of "$ $$" with "$ $$".
3. Replace all occurrences of "#" with "/".
4. If necessary, change the STORPOOL parameter of the ACQUIRE command.
5. If necessary, change the LIBDEF card to point to your Control Center code library, or remove the LIBDEF card.
6. If necessary, change the sublibrary parameter (S=) on the SLI statement to point to your database manager production library.
7. If necessary, change the database name parameter on the // EXEC ARIDBS card to point to your database manager.
8. Return Code 6 is acceptable if the PUBLIC.SQMHELP DBSPACE has never been previously acquired.

Step 7: Load Packages into Server(s)

This step loads the Control Center packages into your databases. These are the programs that contain calls to the server. In order for Control Center to work with your database(s), you must load these packages into each database.

Library member SQMRLDPK.Z (Figure 74 on page 158) contains a multiple user mode DBSU job to RELOAD the Control Center packages. As delivered, SQMRLDPK searches the DB2 Server for VSE Version 6.1 library for the DBSU phase. If you are running SQL/DS 3.5, you must change the LIBDEF statements.

The // EXEC ARIDBS card specifies the server-name as "DB2VSE61". If your server-name is different, replace "DB2VSE61" with the name of your database.

SQMRLDPK reads the packages (.Q members) from the distribution library. Do not punch the .Q members for editing and submission.

To load the Control Center packages into your databases:

1. Punch member SQMRLDPK.Z for editing and submission.
2. Replace all occurrences of "$ $$" with "$ $$".
3. Replace all occurrences of "#" with "/".
4. Change the LIBDEF, if necessary, to point to your production database manager library.
5. If necessary, change the database name parameter on the // EXEC ARIDBS card to point to your database manager.

Chapter 4. Getting Started

This chapter explains how to start using the Control Center feature and introduces you to the main menu. The following chapters describe how to use the Control Center tools.

Getting Started With Control Center

You can type **SQM** on a blank CICS screen to reach the main menu.

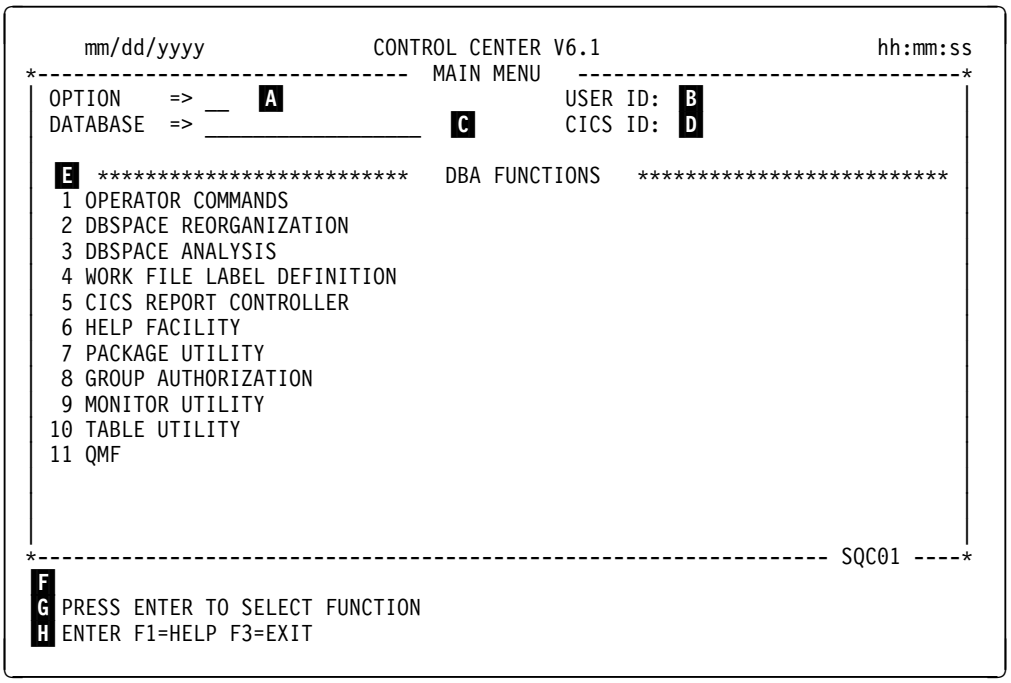

Figure 8. Control Center Main Menu

Some things you should know about the Control Center Main Menu are:

A OPTION—Selects the function you want to use. Enter the number of the tool you want to use in this field. The option number is the highlighted, 1 digit identifier displayed to the left of the option description.

B USER ID—The 8-character USER ID of the signed on user.

C DATABASE—Identifies the database you are working with. When you sign on, this field displays the name of the default database. To work with a different database you can type the name of the new database in this field.

D CICS ID—Shows the 8-character APPLID of the CICS system owning the transaction.

E DBA FUNCTIONS—Lists the functions available to a database server administrator.

F Error Message Line—Error messages, if any, are displayed on this line.

G Instruction Line—Instructions are displayed on this line.

H Function (F) Key Line—Displays active function keys and control keys.

Each of the DBA Functions shown in Figure 8 is described in detail in the chapters that follow, except for the QMF function. Selecting option 11, QMF, directly invokes QMF. If it completes normally, control returns to Control Center, otherwise QMF handles the error.

Chapter 5. Using the Operator Command Interface Tool

The Operator Command Interface tool provides an interface between you and the database to perform operator SHOW and COUNTER commands. With this tool, you do not need to directly enter ISQL commands, nor do you need to issue the commands from the VSE Operator Console.

```
     mm/dd/yyyy               CONTROL CENTER V6.1                      hh:mm:ss
*----------------------------- OPERATOR COMMANDS -----------------------------*
| OPTION ===>                                                USER ID: VS12     |
| PARMS ====> _____________________________________          CICS ID: SVSCVSAX |
| DATABASE => DB2VSE61_____________                                            |
| ****************************** SHOW COMMANDS ************************       |
| 1  ACTIVE                            11 ADDRESS module                       |
| 2  BUFFERS                           12 CONNECT (ALL|uid|AGENT n|LUWID id|   |
| 3  DBCONFIG                                              ACTIVE|INACT)      |
| 4  DBEXTENT                          13 DBSPACE n                            |
| 5  INVALID                           14 LOCK ACTIVE                          |
| 6  LOCK DBSPACE ALL|n                15 LOCK GRAPH uid|AGENT n               |
| 7  LOCK MATRIX                       16 LOCK USER (ALL|uid|AGENT n)          |
| 8  LOCK WANTLOCK (ALL|uid|AGENT n)   17 LOG                                  |
| 9  LOGHIST (ALL|n|SERVICE)           18 POOL (ALL|SUMMARY|DELETED|n)         |
| 10 STORAGE                           19 SYSTEM                               |
| ***************************** OTHER COMMANDS ****************************** |
| 20 COUNTER *|name                                                            |
|                                                                              |
*------------------------------------------------------------------ SQC11 ----*

 ENTER OPTION, PARMS, AND DATABASE NAME AND PRESS ENTER
 ENTER F1=HELP F3=EXIT
```

Figure 9. DBA Operator Commands Screen

The DBA Operator Commands screen is shown in Figure 9.

The database server you are currently working with is identified by the **DATABASE** field shown near the top of the screen.

To execute an operator command:

- Enter the 1 or 2 digit OPTION number that precedes each command.
- Enter any required parameters. (The parameters shown in uppercase are to be entered as shown. Those shown in lowercase are for you to supply.)
- Enter the database name in the DATABASE field.

For example, if you want to issue a SHOW LOCK USER on AGENT1 command against the DB2PROD database, enter:

```
OPTION ===> 16
PARMS ====> AGENT 1
DATABASE => DB2PROD
```

The result of the command will be displayed. For example, if you issued the following SHOW ACTIVE command against server DB2VSE61:

```
OPTION ===> 1
PARMS ====>
DATABASE => DB2VSE61
```

you would see:

```
     mm/dd/yyyy              CONTROL CENTER V6.1                      hh:mm:ss
*---------------------- OPERATOR COMMAND DISPLAY SCREEN ----------------------*
| COMMAND  => SHOW ACTIVE                                                     |
| DATABASE => DB2VSE61                                                        |
|                                                                             |
| Status of agents:                                                           |
|    Checkpoint agent is not active.                                          |
|    User Agent: 3 User ID: SVSCVSAX is NIW  SUBS                             |
|       Agent is not processing and is in communication  wait.                |
|    User Agent: 4 User ID: SQLMSTR  is R/O  SUBS  BDFF                       |
|       Agent is processing an operator command.                              |
|    User Agent: 5 User ID: SVSCVSAX is NIW  SUBS                             |
|       Agent is not processing and is in communication  wait.                |
|    2 agent(s) not connected to an APPL or SUBSYS.                           |
| ARI0065I Operator command processing is complete.                           |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
*------------------------------------------------------------------ SQC12 ----*


F1=HELP,F3=EXIT,F4=TOP,F5=BOT,F7=BWD,F8=FWD,F12=CANCEL
```

Figure 10. Show Active Command Display Screen

For a complete description of each operator command and appropriate parameters, refer to the *DB2 Server for VSE & VM Operation* manual.

Chapter 6. Using the Database Monitors

The Control Center Monitor Utilities are a DBA's best friend. They keep watch over the database while you are busy doing something else. They notify you when exceptional conditions occur and they capture key information that you need to track performance problems and resource consumption.

Whether you are interested in tracking user activity, locking, physical and logical space use, or the database itself, there are monitors designed to give you that information.

The SHOW ACTIVE and SHOW CONNECT monitors collect information that tells you if users have been waiting for database resources. You can choose to set the Monitor Utility to alert you immediately if it detects a user in wait state.

The SHOW LOCK monitor records the number of lock buffers in use and who is locking whom. You can also set this monitor to alert you immediately if it detects any kind of user locking.

The SHOW LOG, SHOW DBEXTENT, and SHOW DBSPACE monitors record the use of these vital resources. You can direct the SHOW LOG and SHOW DBEXTENT monitors to notify you immediately if their use exceeds the percentage you specify. The COUNTER * monitor records the occurrence of key events in the database and stores this information in the database. You are then free to develop your own custom queries to track statistics like buffer hit ratio or LPAGBUFF and PAGEREAD.

How the Monitors Work

The monitor "kernel" is the heart of the Monitor Utility. It is a COBOL/CICS program that runs unattached to any terminal. There is only one monitor kernel per database. Each time a monitor kernel is activated, it determines what monitors should be scheduled and starts a task for each. At midnight, the kernel schedules the Reset Monitor task, that deletes the monitor data and optionally produces a report for all monitors scheduled to be reset.

Each individual monitor:

- Issues a database operator command,
- Captures the output,
- Writes information to the appropriate monitor table.
- Optionally provides notification of exceptional conditions where applicable.

This monitor kernel reactivates itself every minute.

Options and Monitors Available

As indicated in Figure 15 on page 43, the Monitor Utilities include options that allow you to:

- Start and stop monitor kernels
- List defined monitors
- Add, modify, delete, and display monitors
- View monitor data online
- Reset monitor data
- Print monitor report

These options use seven monitors:

- SHOW ACTIVE
- SHOW LOCK
- SHOW DBEXTENT
- SHOW LOG
- SHOW CONNECT
- SHOW DBSPACE
- COUNTER *

Monitor Thresholds and VSE Console Messages

All the monitors listed, except SHOW DBSPACE and COUNTERs *, let you specify a threshold at which Control Center will generate a VSE console message.

For instance, if the SHOW ACTIVE monitor finds a CHECKPOINT or USER WAIT situation when it executes, or if SHOW LOG or SHOW DBEXTENT reaches a PERCENT USED value, Control Center sends a message to the VSE console. Similarly, if SHOW CONNECT finds AGENTS WAITING, NOT PROCESSING or INACTIVE, or if the SHOW LOCK monitors meet the locking indicators, Control Center will issue the appropriate message.

Figure 11 on page 39 is an example of the messages sent to the VSE console when a SHOW ACTIVE monitor threshold is reached.

```
SYSTEM:  VSE/ESA       V2.1.0  --  SVSCVSA                    USER:  VS22
                                                              TIME:  hh:mm:ss
F2 0002 SQM0632 SQLMSTR  IS IN COMMUNICATION WAIT
F2 0002 SQM0318 DATABASE:      DB2VSE61
F2 0002 SQM0632 CMORGAN  IS IN COMMUNICATION WAIT
F2 0002 SQM0318 DATABASE:      DB2VSE61




==>

1=HLP 2=CPY 3=END 4=RTN 5=DEL 6=DELS 7=RED 8=CONT 9=EXPL 10=HLD         12=RTRV

ACT_MSG: HOLD              PAUSE: 01  SCROLL: 1           MODE:  CONSOLE
```

Figure 11. VSE Console - Monitor Threshold Notification

In this figure, Database DB2VSE61 shows two agents, SQLMSTR and CMORGAN, in communication wait state. Although neither situation is necessarily a problem, you may want to investigate continued wait states.

Description of Monitor Options

Start the Monitor Kernel

You start the monitor kernel from the Monitor Menu. The Monitor Menu is invoked by selecting Option **09** Monitor Utilities from the Control Center Menu and is shown in Figure 12. You must define at least one monitor in order to use this function. Once started, the monitor kernel runs until you execute the Stop Monitor command. See “Add a Monitor” on page 40.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                    hh:mm:ss
*--------------------------     MONITOR MENU     ----------------------------*
|                                                                             |
|  DATABASE     => DB2VSE61                                                   |
|                                                                             |
|  ***************************************************************************|
|                                                                             |
|  OPTION       =>    01 START KERNEL             02 STOP KERNEL             |
|                     03 LIST MONITORS            04 ADD MONITOR             |
|                     05 MODIFY MONITOR           06 DELETE MONITOR          |
|                     07 DISPLAY MONITOR          08 VIEW DATA               |
|                     09 RESET DATA               10 PRINT REPORT            |
|                                                                             |
|  ***************************************************************************|
|                                                                             |
|  MONITOR      =>    01 SHOW ACTIVE              02 SHOW LOCK               |
|                     03 SHOW DBEXTENT            04 SHOW LOG                |
|                     05 SHOW CONNECT             06 SHOW DBSPACE 00001      |
|                     07 COUNTER *                                           |
|                                                                             |
*-------------------------------------------------------------- SQC40 ----*

PRESS ENTER TO DELETE MONITOR, PF3 TO CANCEL
ENTER F1=HELP F3=EXIT
```

Figure 12. Monitor Menu

Stop the Monitor Kernel

Lets you end the execution of a monitor, once monitor activity during that time period has been completed. When you stop the Monitor Kernel, you stop all the monitors.

List Monitors

Lists all the monitors defined for the specified database on your terminal. Figure 13 shows the monitors belonging to the DB2VSE61 database.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                      hh:mm:ss
*--------------------------     MONITOR LIST     ----------------------------*
| DATABASE     => DB2VSE61                                                    |
|                                                                             |
| MONITOR_____       DESCRIPTION_____________________________________________ |
| ACTIVITY           SHOW ACTIVE                                              |
| LOCKING            SHOW LOCK ACTIVE                                         |
| COUNTERS           COUNTER *                                                |
| LOG                SHOW LOG                                                 |
| POOL               SHOW POOL                                                |
| DBEXTENT           SHOW DBEXTENT                                            |
| DBSPACE00001       SHOW DBSPACE     1                                       |
| DBSPACE00002       SHOW DBSPACE     2                                       |
| DBSPACE00003       SHOW DBSPACE     3                                       |
| DBSPACE00004       SHOW DBSPACE     4                                       |
| DBSPACE00005       SHOW DBSPACE     5                                       |
| DBSPACE00006       SHOW DBSPACE     6                                       |
| DBSPACE00007       SHOW DBSPACE     7                                       |
| DBSPACE00010       SHOW DBSPACE    10                                       |
| DBSPACE00022       SHOW DBSPACE    22                                       |
*-------------------------------------------------------------- SQC43 ----*
F1=HELP F3=EXIT F4=TOP F5=BOT F7=BWD F8=FWD F12=CANCEL
```

Figure 13. Monitor List

Add a Monitor

Lets you define a monitor, the first step in using one. You specify the database you want to monitor, the type of monitor, the frequency, and the start and stop times. You also define the threshold at which notification will occur.

Modify a Monitor

Lets you view the monitor definition information at your workstation. You can choose to update the frequency, start and stop times, and the monitor thresholds from these screens.

Delete a Monitor

Lets you delete active monitors. Figure 12 on page 39 shows the Monitor Menu. To delete the SHOW LOG monitor from the DB2VSE61 database, for example, select Option **06**. Next select Monitor **04** and press ENTER.

Note that you can press PF3 to avoid deleting the monitor.

Display a Monitor

Lets you view the definition of a monitor.

View Data

Lets you view the data the monitor has collected. With this option, you can analyze the output and watch for significant database use or trends.

Reset Data

Deletes all rows of data collected by the monitor from the associated monitor table. Data keeps accumulating until you reset the monitor.

Print Report

Generates a monitor report and places it directly on the VSE/POWER List queue. The monitor report includes all the data that the monitor has captured. To Print a Monitor Report, select Option **10** from the main Monitor Menu, together with the desired monitor ID.

The figure that follows is where you include appropriate job submission parameters. The **RESET DATA?** option keeps the data or deletes it from the related monitor table. Specify "1" to refresh the table, or "2" to show cumulative data.

```
     mm/dd/yyyy               CONTROL CENTER V6.1                      hh:mm:ss
*----------------------------    MONITOR REPORT    ----------------------------*
|                                                                               |
|  DATABASE     => DB2VSE61_________              MONITOR     => DBSPACE00001  |
|                                                                               |
|  ***********************      REPORT PARAMETERS      ***********************  |
|                                                                               |
|  REPORT NAME       => ________                                                |
|                                                                               |
|  CLASS             => A                                                       |
|                                                                               |
|  PRI               => 3                                                       |
|                                                                               |
|  DISP              => D (D,H,L,K)                                             |
|                                                                               |
|  ***************************************************************************  |
|                                                                               |
|  RESET DATA?       => 1 (1=YES/2=NO)                                          |
|                                                                               |
|                                                                               |
*------------------------------------------------------------------ SQC45 -----*

PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT F12=CANCEL
```

Figure 14. Monitor Report Option

Types of Monitors

SHOW ACTIVE

Monitors active database users. You can also set this monitor to send a message to the VSE console when it detects an active checkpoint agent or a user agent in checkpoint, communication, or lock wait.

SHOW LOCK

Monitors database lock contention. In addition, you can set this monitor to send a message to the VSE console whenever it detects a lock holder not processing situation, due to a checkpoint or any lock contention at all.

SHOW DBEXTENT

Monitors page use for storage pools. You can also set this monitor to send a message to the VSE console if the percentage of the pages in use for any storage pool reaches the specified threshold.

SHOW LOG

Monitors database log use. You can also set this monitor to send a message to the VSE console if the log reaches or exceeds the percentage used that you specify.

SHOW CONNECT

Monitors users connected to the database. You can also set this monitor to send a message to the VSE console whenever it detects an active user not processing or inactive users.

SHOW DBSPACE

Monitors dbspace use. You must specify a dbspace number.

COUNTER *

Monitors the occurrence of key events in the database.

For more detailed information about the DB2 Server for VSE commands, refer to the *DB2 Server for VSE & VM Operation* manual.

Invocation

You start monitors from the Monitor Utility Menu. To reach this menu, choose Option **9** Monitor Utility, from Control Center's main menu.

How To Use the Monitors

To use the monitors, you:

1. Add a monitor. As part of adding the monitor, you define the conditions you want to set for the monitor.
2. Start the kernel that activates the monitor you have defined.
3. Work with the information that the monitors give you.

Adding A Monitor

To add a monitor, choose:

1. The name of the database you want to monitor,
2. Option 4 (Add Monitor), and
3. The monitor identifier (ID) that you want to add.

Figure 15 shows an example of adding the SHOW ACTIVE monitor for the DB2VSE61 database.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                    hh:mm:ss
*--------------------------     MONITOR UTILITY  ---------------------------*
|                                                                           |
|  DATABASE     => DB2VSE61                                                 |
|                                                                           |
|  *************************************************************************|
|                                                                           |
|  OPTION       => 04 01 START KERNEL             02 STOP KERNEL            |
|                     03 LIST MONITORS            04 ADD MONITOR            |
|                     05 MODIFY MONITOR           06 DELETE MONITOR         |
|                     07 DISPLAY MONITOR          08 VIEW DATA              |
|                     09 RESET DATA               10 PRINT REPORT           |
|                                                                           |
|  *************************************************************************|
|                                                                           |
|  MONITOR      => 01 01 SHOW ACTIVE              02 SHOW LOCK              |
|                     03 SHOW DBEXTENT            04 SHOW LOG               |
|                     05 SHOW CONNECT             06 SHOW DBSPACE 00001     |
|                     07 COUNTER *                                          |
|                                                                           |
*------------------------------------------------------------------ SQC40 ----*

PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT
```

Figure 15. How to Add a SHOW ACTIVE Monitor

When you press ENTER to process the choices you have made on the Monitor Menu, you see the Monitor Maintenance menu. The Monitor Maintenance menu shows you the status of the monitor on the current database. You use this menu to turn a specific monitor on and off. You also schedule when you want the monitor to run. You can choose to reset the data the monitor has collected as well as define and print reports from here.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                      hh:mm:ss
*------------------------    MONITOR MAINTENANCE    -------------------------*
|                                                                            |
|  DATABASE          => DB2VSE61            MONITOR           => ACTIVE      |
|                                                                            |
|  DESCRIPTION       => SHOW ACTIVE Monitor                                  |
|  ACTIVE?           => 1         (1=YES/2=NO)                               |
|                       1 2 3 4 5 6 7                                        |
|  RUN DAYS          => 2 1 1 1 1 1 2  (1=YES/2=NO)                          |
|  START TIME        => 1400      (HHMM)                                     |
|  STOP TIME         => 1600      (HHMM)                                     |
|  INTERVAL          => 0005      (HHMM)                                     |
|  RESET DATA?       => 2         (1=YES/2=NO)                               |
|  RESET DAY         => 1         (1-7)                                      |
|  PRINT REPORT?     => 1         (1=YES/2=NO)                               |
|                                                                            |
|  ************************     REPORT PARAMETERS     ********************** |
|                                                                            |
|  REPORT NAME       => ACTIVE    CLASS     => A                            |
|  PRI               => 3         DISP      => D (D,H,L,K)                  |
*---------------------------------------------------------------- SQC42 -----*

PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT F12=CANCEL
```

Figure 16. Adding a Monitor on the Monitor Maintenance menu

The monitor shown above is active and scheduled to run Monday-Friday (days 2-6), every five minutes, from 14:00 to 16:00. Additional options include printing the report and not resetting the data collected in the monitor table.

You can specify:

Field	Description
DATABASE	Displays the name of the database you specified on the Monitor Menu.
MONITOR	Shows the status of the monitor.
DESCRIPTION	Enter a brief description of the monitor.
ACTIVE?	Deactivates a monitor without having to delete the monitor completely. Use **1** to activate the monitor and **2** to deactivate it.
RUN DAYS	Indicates the days of the week on which you want the monitor to run. You can specify all days, specific days, or a range of days. Use 1 for YES and 2 for NO under the number for each day where 1=Sunday, 2=Monday, 3=Tuesday, and so on. The figure above shows monitor execution from Monday through Friday (days 2-6 are set to "1").
START TIME	Indicates when the monitor is to start for the day(s) specified.
STOP TIME	Indicates when the monitor is to terminate for the day(s) specified.
INTERVAL	Indicates how often the monitor will run during the time period between the start and stop times. You can schedule the monitors to run with frequencies from 2 minutes to several hours.

RESET DATA?	Deletes all rows of data collected by the monitor from the associated monitor table. Use **1** to delete all the rows and **2** to keep them.
RESET DAY	Used to indicate what day to RESET DATA.
PRINT REPORT?	Generates a monitor report and places it directly on the VSE/POWER List queue.
REPORT NAME	Specifies the name for the print monitor report job and its associated queue entries.
CLASS	Specifies the class or partition in which you want this job to run. The class defaults to A.
PRI	Specifies the priority that is to be assigned to the job. Specify a number from 0 to 9 where 9 is the highest priority. Default priority is 3.
DISP	Specifies how the job is to be handled in the reader queue. Disposition may be specified as: • D - Delete after processing • H - Hold until released • K - Keep after processing • L - Leave in the queue Disposition defaults to D.

Specifying Monitor Thresholds

When Adding a SHOW ACTIVE, SHOW CONNECT, SHOW DBEXTENT, SHOW LOCK or a SHOW LOG monitor, you can also specify a threshold for the monitor.

Figure 17 shows how to set the CHECKPOINT WAIT and USER WAIT thresholds for the SHOW ACTIVE monitor that belongs to database DB2VSE61.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                      hh:mm:ss
*------------------------     MONITOR THRESHOLDS    --------------------------*
|                                                                             |
|  DATABASE     => DB2VSE61                       MONITOR     => ACTIVITY     |
|                                                                             |
|  ***************************     SHOW ACTIVE     ***************************|
|  CHECKPOINT WAIT                        => 1 (1=YES/2=NO)                   |
|  USER WAIT                              => 1 (1=YES/2=NO)                   |
|                                                                             |
|  ***************************    SHOW CONNECT     ***************************|
|  AGENT NOT PROCESSING                   => 2 (1=YES/2=NO)                   |
|  INACTIVE                               => 2 (1=YES/2=NO)                   |
|                                                                             |
|  ****************************    SHOW LOCK    ****************************|
|  CHECKPOINT                             => 2 (1=YES/2=NO)                   |
|  ANY LOCKING                            => 2 (1=YES/2=NO)                   |
|                                                                             |
|  ***************         SHOW LOG AND SHOW DBEXTENT          ***************|
|  PERCENT USED                           => __ %                             |
|                                                                             |
*-------------------------------------------------------------------- SQC42 -----*
PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT F12=CANCEL
```

Figure 17. SHOW ACTIVE Threshold Specification

To activate threshold checking, for the SHOW ACTIVE, SHOW CONNECT, or SHOW LOCK monitors, specify "1" (YES). To deactivate threshold checking, specify "2" (NO).

To initiate SHOW LOG or SHOW DBEXTENT threshold checking, specify PERCENT USED as a number between 1 and 99. Figure 18 shows how to set the **SHOW LOG** monitor to issue a VSE console message if the monitor detects a value 75% or greater.

```
     mm/dd/yyyy               CONTROL CENTER V6.1                      hh:mm:ss
*------------------------     MONITOR THRESHOLDS    --------------------------*
| DATABASE     => DB2VSE61                       MONITOR     => LOG            |
|                                                                              |
|                                                                              |
| ***************         SHOW LOG AND SHOW DBEXTENT          *************** |
| PERCENT USED                                 => 75%                          |
*------------------------------------------------------------------ SQC42 -----*

PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT F12=CANCEL
```

Figure 18. SHOW LOG Threshold Specification

The fields displayed on the Monitor Threshold screen are:

Entry Field	Description
CHECKPOINT WAIT	The checkpoint agent is active and is in "wait" state.
USER WAIT	User is in a wait state caused by locking, checkpoint or communication wait.
AGENT NOT PROCESSING	Active agent that is not processing.
INACTIVE	User connected but inactive.
CHECKPOINT	Users locked because the database is taking a checkpoint.
ANY LOCKING	Users locked for any reason.
PERCENT USED	For the SHOW LOG monitor, log use has reached the percent specified. For the SHOW DBEXTENT monitor, log use has reached the percent specified.

Starting a Monitor

Once you have defined a monitor, you need to initiate it. Choose Option **1**, **Start Kernel** from the Monitor Utility menu.

Changing a Monitor

Once you are using a monitor, you may decide you want to change some or all of its settings. For example, you may decide you want the monitor to execute more frequently. If you want to modify a monitor, choose Option **5** on the Monitor Menu. You will see the Monitor Maintenance menu where you can make your changes to the monitor. For example, Figure 19 on page 47 shows how to enter **2** in the ACTIVE? field to deactivate the SHOW ACTIVE monitor.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                      hh:mm:ss
*-------------------------   MONITOR MAINTENANCE   -------------------------*
|                                                                            |
|  DATABASE          => DB2VSE61            MONITOR          => ACTIVE      |
|                                                                            |
|  DESCRIPTION       => SHOW ACTIVE Monitor                                  |
|  ACTIVE?           => 2                    (1=YES/2=NO)                    |
|                       1 2 3 4 5 6 7                                        |
|  RUN DAYS          => 2 1 1 1 1 1 2        (1=YES/2=NO)                    |
|  START TIME        => 1400      (HHMM)                                     |
|  STOP TIME         => 1600      (HHMM)                                     |
|  INTERVAL          => 0005      (HHMM)                                     |
|  RESET DATA?       => 2         (1=YES/2=NO)                               |
|  RESET DAY         => 1         (1-7)                                      |
|  PRINT REPORT?     => 1         (1=YES/2=NO)                               |
|                                                                            |
|  ************************     REPORT PARAMETERS     ********************** |
|                                                                            |
|  REPORT NAME       => ACTIVE    CLASS     => A                            |
|  PRI               => 3         DISP      => D (D,H,L,K)                  |
*---------------------------------------------------------------- SQC42 -----*

PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT F12=CANCEL
```

Figure 19. Updating (Deactivating) a Monitor

And Figure 20 shows how to enter **0002** to execute the SHOW ACTIVE monitor every two minutes instead of the five minutes originally defined in Figure 19.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                      hh:mm:ss
*-------------------------   MONITOR MAINTENANCE   -------------------------*
|                                                                            |
|  DATABASE          => DB2VSE61            MONITOR          => ACTIVE      |
|                                                                            |
|  DESCRIPTION       => SHOW ACTIVE Monitor                                  |
|  ACTIVE?           => 2                    (1=YES/2=NO)                    |
|                       1 2 3 4 5 6 7                                        |
|  RUN DAYS          => 2 1 1 1 1 1 2        (1=YES/2=NO)                    |
|  START TIME        => 1400      (HHMM)                                     |
|  STOP TIME         => 1600      (HHMM)                                     |
|  INTERVAL          => 0002(HHMM)                                           |
|  RESET DATA?       => 2         (1=YES/2=NO)                               |
|  RESET DAY         => 1         (1-7)                                      |
|  PRINT REPORT?     => 1         (1=YES/2=NO)                               |
|                                                                            |
|  ************************     REPORT PARAMETERS     ********************** |
|                                                                            |
|  REPORT NAME       => ACTIVE    CLASS     => A                            |
|  PRI               => 3         DISP      => D (D,H,L,K)                  |
*---------------------------------------------------------------- SQC42 -----*

PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT F12=CANCEL
```

Figure 20. Updating (Deactivating) a Monitor

Viewing Monitor Data

The View Data Option displays the selected monitor report at the user's terminal. An example of the View Data Option for the **SHOW ACTIVE** monitor is below. This information is useful in monitoring trends or resource consumption of your database.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                    hh:mm:ss
*----------------------------     MONITOR DATA     ---------------------------*
| DATABASE     => DB2VSE61                       MONITOR     => ACTIVE        |
|                                                                             |
|                                                 COMM LOCK CHKPT PAGE BLOCK  I/O
|    DATE      TIME   NACT NIW R/O R/W NEW WAIT WAIT  WAIT WAIT  WAIT WAIT
| ---------- -------- ---- --- --- --- --- ---- ---- ----- ---- ----- ----
| 1997-01-18 16.25.10    3   2   1   0   0    0    0     0    0     0    0
| 1997-01-18 17.06.56    3   2   1   0   0    0    0     0    0     0    0
| 1997-01-19 16.16.37    3   2   1   0   0    2    0     0    0     0    0
| 1997-01-19 16.17.05    3   2   1   0   0    2    0     0    0     0    0
| 1997-01-19 16.20.01    3   2   1   0   0    2    0     0    0     0    0
| 1997-01-19 16.22.01    3   0   3   0   0    2    0     0    0     0    0
| 1997-01-19 16.24.01    3   0   3   0   0    2    0     0    0     0    0
| 1997-01-19 16.26.01    3   1   2   0   0    2    0     0    0     0    0
| 1997-01-19 16.28.01    3   1   2   0   0    2    0     0    0     0    0
| 1997-01-19 16.49.18    3   2   1   0   0    2    0     0    0     0    0
| 1997-01-19 16.55.18    3   1   2   0   0    2    0     0    0     0    0
| 1997-01-19 17.01.18    3   1   2   0   0    2    0     0    0     0    0
*------------------------------------------------------------------ SQC44 ----*

F1=HELP F3=EXIT F4=TOP F5=BOT F7=BWD F8=FWD F10=LEFT F11=RIGHT F12=CANCEL
```

Figure 21. Monitor Report Data

To understand the monitor measurements, refer to the appropriate sections of the *DB2 Server for VSE & VM Operation* manual.

Stopping a Monitor

There are two methods that can be used to stop a monitor. You can disable **ALL** monitors by selecting the STOP KERNEL option from the Monitor Utility menu, or you can disable a single monitor by deactivating it from the Monitor Maintenance menu. To restart stopped monitors, select the START KERNEL from the Monitor Utility menu, or to restart a single monitor, reactivate it from the Monitor Maintenance menu.

Chapter 7. Group Authorization Tool

The Group Authorization tool helps DBAs manage access to database objects, simplifies the authorization process, and shortens the time needed to grant or revoke privileges. It lets DBAs issue authorizations to groups of users on groups of objects rather than one by one. You can associate individual users with defined User Groups, and you can associate database objects (such as tables, views, and packages) with defined Application Groups. Then you can use the Group Authorization menus to issue GRANTs and REVOKEs specifying a User Group (grantee) and an Application Group (on objects).

About the Group Authorization Tool

The Group Authorization Tool is a series of CICS transactions that operate under the SQM main transaction. The "SQLMSTR" ID grants all authorizations. The tool records and maintains all authorizations.

You can use the “LIST Functions” on page 51 to take advantage of the various reports that are available to help manage database access. These listings, or reports, show Application Groups and the objects they contain, such as tables, views and packages; User Groups and user IDs associated with specific User Groups; and authorities granted to User Groups.

The Group Authorization tool keeps all data about User and Application Groups, as well as authorization information, in database tables. You can query these tables to obtain authorization information. See “Special Considerations” on page 56 for an example.

The "SQLMSTR" ID owns five authorization tables that contain all the information about User, Application Groups and authorizations. These tables 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 unique combination of object (object owner, object name) and Application Group ID.
GROUP_AUTH_TAB	Group Authorization Table. This table records each group authorization made by SQLMSTR. It records the User Group ID, the Application Group ID and the specific privileges granted using the Group Authorization tool.

Users can belong to more than one User Group and can have the same privilege granted to an object through multiple User Groups. If you drop the user from one

User Group, the user does **not** automatically lose the privilege to the object since the user still has authority through the second User Group.

The same is true of an object that 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 automatically revoked from users who have authority on it through the second group.

You can choose to:

ADD a Group

Lets you create both User and Application Groups. The data created will be stored in the USER_GROUP_TAB and APPL_GROUP_TAB tables respectively.

DROP a Group

Deletes a group entry from the applicable tables. 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 that 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 if the user has been granted the same privileges through another group. If they do belong to a User Group with the same privileges, the user will not lose their privileges.
2. When dropping an Application Group that has had privileges granted on it, a check is made before revoking the privileges from each user to whom they were granted. If the privileges on the object have been granted to that user through another Application Group, the user will not lose their privileges.

Manage Group Objects and 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). Application Groups are 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. Each group type must have at least one member.

When you add an object to an Application Group, any privileges of existing User Groups 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 privileges to that object will be REVOKED for all users unless the user is a member of another group with similar privileges. (See the exceptions in “DROP a Group.”)

When you add a user to a User Group, the user is granted all authorities that the group currently holds. When you drop a user from a group, the user loses all authorities which the group has unless the user is a member of another group with the same authorities. (See exceptions “DROP a Group.”)

Manage Privileges

Lets you grant and revoke privileges to User Groups on individual database objects or on Application Groups (that is, on all objects defined in the group). You cannot grant column update privileges; however, you can create a view with the column updates and then grant update privilege on the view.

Use the Authorizations Menu to enter an individual object or a group of objects on which you want to grant or revoke privileges. An individual object is identified by its owner and object name. A group object is identified by its Application Group name. You use the SQLMSTR connect ID authorizations.

For example, in the SYSTEM.SYSTABAUTH table, SQLMSTR is always the grantor for group authorizations. An extra grant is done for the User Group name to facilitate the implementation of the tool. If you give SELECT privilege to User Group UGROUP1 on Application Group AGROUP1, the Group Authorization tool generates one extra GRANT SELECT to UGROUP1 on each object in AGROUP1. Likewise, when you use the REVOKE function, the SELECT privilege is revoked from the User Group.

LIST Functions

The Group Authorization tool provides the following on-line reports:

- 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

---

Using the Group Authorization Tool

To use the Group Authorization tool, you need to:

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 reach the Group Authorization Menu shown in Figure 22:

- Select Option **8** from the Control Center main menu, or
- Enter the **SQGA** transaction ID from a CICS screen.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                    hh:mm:ss
*--------------------      GROUP AUTHORIZATION MENU      --------------------*
| DATABASE  => DB2VSE61                                                      |
| OPTION    =>                                                               |
|                                                                            |
|  1 USER GROUP FUNCTION                                                     |
|  2 USER FUNCTION                                                           |
|  3 APPLICATION GROUP FUNCTION                                              |
|  4 GROUP AUTHORIZATIONS                                                    |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
*---------------------------------------------------------------- SQC19 ----*
PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT
```

Figure 22. Control Center Group Authorization Menu

Before using the Group Authorization tool, you need to analyze your current database authorization structure. Start by grouping tables and views according to some common function or element(s). For instance, perhaps you want all payroll, personnel, or accounting tables in their own specific groups. Once you have decided this, you can define Application Groups for these various functions.

Let's say you have 5 tables and 3 views that belong in the personnel organization and you might want to define three Application Groups for this organization. The Application Groups will contain the tables and views from the personnel organization.

The following steps show how to use the tool to define the application groups. Before starting, decide what tables will be in which group as shown in Application Group Definitions:

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 (PERSVIEW3).

Step 1: Define Application Groups

To define an Application Group, enter Option **3** from the Group Authorization Menu to get to the Application Group Menu.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                        hh:mm:ss
*------------------------- APPLICATION GROUP MENU ---------------------------*
|  DATABASE   => DB2VSE61                                                    |
|  GROUP      => AGROUP1                                                     |
|  OPTION     => 1                                                           |
|                                                                            |
|   1 ADD  GROUP                                                             |
|   2 DROP GROUP                                                             |
|   3 ADD/DROP OBJECTS IN A GROUP                                            |
|   4 LIST GROUPS                                                            |
|   5 LIST OBJECTS IN A GROUP                                                |
|   6 LIST GROUP AUTHORIZATIONS                                              |
|                                                                            |
|  ****************************************************************************
|                                                                            |
|  DESCRIPTION       => Personnel tables                                     |
|  GROUP TYPE        => T           (T OR P)                                 |
|                                                                            |
|                                                                            |
|                                                                            |
*---------------------------------------------------------------- SQC26-------*

PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT F12=RETURN
```

Figure 23. Control Center Application Group Menu

You enter the name of the Application Group you are adding in the Group field, choose Option **1**, Add Group, and enter the Group Type (use **T** for tables or views or **P** for packages). You can also choose to add a description of the group in the Group Description field. If you enter a Group Description, it is stored in the database record and you can view it in the Application Group list report. The database you are working with is shown at the top of the screen in the Database field. If you want to work with a different database, enter the name of the new database in the Database field.

The example shown in Figure 23 shows how to add the group AGROUP1. The Application Group type is **T** for tables or views and Personnel tables has been added as the Group description.

Step 2: Add Objects to the Application Group

Once you have defined the Application Group, you add the objects (that is, tables and views, or packages) to the group. On the Application Group Menu, choose Option **3** to add the objects to the Application Group. You add tables and views, or packages to the Application Group on the Object Functions menu. Figure 24 on page 54 shows how to add the three personnel tables and a view to the AGROUP1 Application Group.

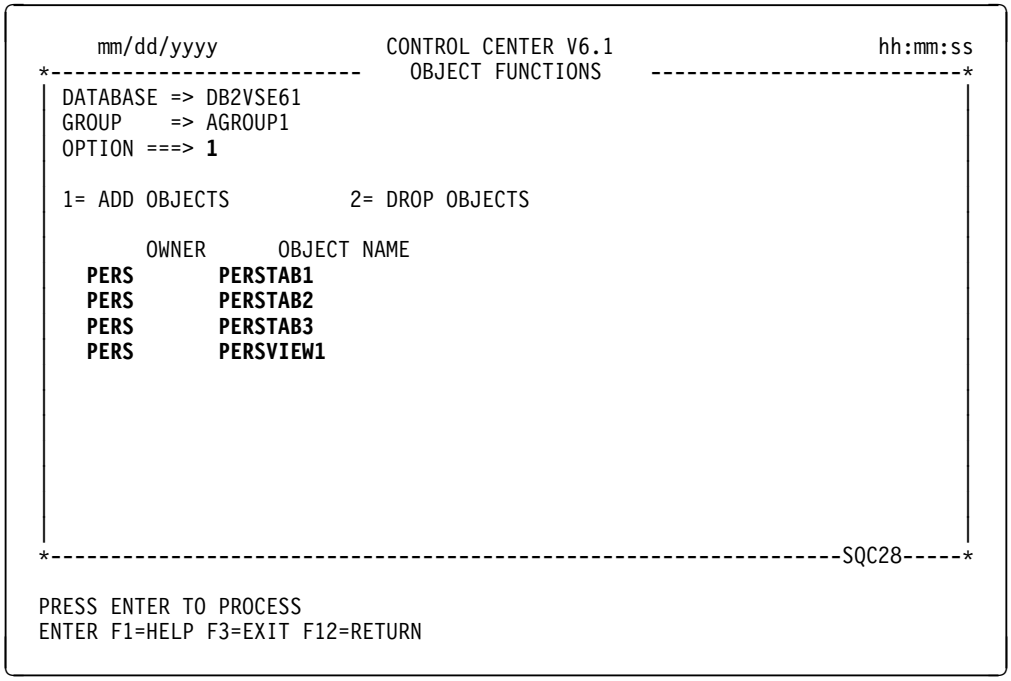

Figure 24. Adding Objects to Your Application Group

Step 3: Define User Groups

After you have established the Application Groups, you define the users of your database to whom you want to grant privileges. For example, let's assume you have ten users whose IDs are USER1, USER2, and so on through USER10. You can now group these users by the criteria you use to determine authorization. Our examples use the following three 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.

Use the User Group Menu to add or delete a User Group, to define the members of a User Group, and to list User Groups. To reach the User Group Menu: From the Group Authorization Menu, select Option **1** to reach the User Group Menu.

Figure 25 on page 55 show how to add the UGROUP1 User Group.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                      hh:mm:ss
*-------------------------     USER GROUP MENU       ------------------------*
| DATABASE   => DB2VSE61                                                      |
| GROUP      => UGROUP1                                                       |
| OPTION     => 1                                                             |
|                                                                             |
|  1 ADD  GROUP                                                               |
|  2 DROP GROUP                                                               |
|  3 ADD USERS TO A GROUP                                                     |
|  4 LIST GROUPS                                                              |
|  5 LIST USERS IN A GROUP                                                    |
|  6 LIST GROUP AUTHORIZATIONS                                                |
|                                                                             |
|                                                                             |
|                                                                             |
| ****************************************************************************|
|                                                                             |
| DESCRIPTION       => Executives                                             |
|                                                                             |
|                                                                             |
*-------------------------------------------------------------- SQC20 ----*

PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT F12=RETURN
```

Figure 25. Adding a User Group

Step 4: Add Users to the User Groups

Now choose Option **3** to add users to the User Group you have defined in Step 3. You enter the User Group member on the Add Users screen shown in Figure 26, or Option **2** from the Group Authorization Menu.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                      hh:mm:ss
*-------------------------        ADD USERS       ------------------------*
| DATABASE => DB2VSE61                                                        |
| GROUP    => UGROUP1                                                         |
| OPTION ===> 1                                                               |
|                                                                             |
| 1= ADD USER(S)                                                              |
|                                                                             |
|       USER                                                                  |
|   User1                                                                     |
|   User2                                                                     |
|   User3                                                                     |
|   User4                                                                     |
|   User5                                                                     |
|   User6                                                                     |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
*-------------------------------------------------------------SQC29------*

PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT F12=RETURN
```

Figure 26. Adding Users to UGROUP1

Step 5: Grant Authorities to the User Groups

Then in this step, you grant object authority to the User Group. Use the Authorization Menu to issue grants or revokes to the User Groups. Figure 27 shows how to grant Select and Update on AGROUP1 to UGROUP1.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                      hh:mm:ss
*-------------------------  AUTHORIZATIONS MENU  -------------------------*
| DATABASE => DB2VSE61                                                    |
|                                                                         |
| 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    )                    |
|                                                                         |
|                                                                         |
*------------------------------------------------------------------ SQC27----*

PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT F12=RETURN
```

Figure 27. Authorization Menu

The choices made in the example show in Figure 27, grant Select and Update privileges on tables PERSTAB1, PERSTAB2, PERSTAB3, and PERSVIEW1 to user IDs USER1 through USER6. To grant multiple privileges at the same time, enter the number corresponding to each privilege separated by a space.

Let's assume you have also granted Select and Update privileges on tables PERSTAB1, PERSTAB2, PERSTAB3, and PERSVIEW1 to UGROUP2 (that contains user IDs USER7 through USER10). Note that you have given USER1 Select and Update privileges on the same tables in AGROUP1. If you revoked the Update privilege from UGROUP1, all members of UGROUP1 would lose their Update privilege except USER1. USER1 would keep the UPDATE privilege because it is also a member of UGROUP2 that still has the UPDATE privilege.

Special Considerations

You grant authorizations to Groups, not to individual user ID's.

1. If you want to grant authorizations to an individual user ID, define a User Group with only one user ID in it. You can then grant authorizations to that user group which, in effect, let's you grant authorizations to a single user ID.
2. Authorizations can be granted to an individual table that 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 Application Group, the Group Authorization tool does not record the authorizations in its tables.

If any user ID in UGROUP1 were to later have its privilege to SQLDBA.ACTIVITY revoked due to the table being defined in an Application Group, the privilege would be deleted from SYSTEM.SYSTABAUTH since the Group Authorization tool does not know about the individually granted authority. If you want the tool to keep a record of the individual authorization, define the table in its own group. Just as you can define a single user in a User Group, you can also define a single object in its own Application Group.

3. You can grant or revoke the following authorizations:
   - SELECT, INSERT, UPDATE, DELETE and ALL on Tables and Views. The ALL authorization is equivalent to the SELECT, INSERT, UPDATE, and DELETE privileges. It does **NOT** include ALTER, INDEX, or REFERENCES.
   - EXECUTE on Packages
   - Use the GRANT option on all the above authorizations:
4. You execute all grants and revokes with the SQLMSTR connect ID.
5. When granting EXECUTE on packages, the "SQLMSTR" ID must have the RUN privilege with GRANT option in order to do the GRANT.
6. You cannot grant referential constraint privileges.
7. The tool will not affect any current authorizations recorded in SYSTEM.SYSTABAUTH and SYSTEM.SYSPROGAUTH, since the "SQLMSTR" ID grants all privileges. As you build your authorization scheme and issue grants using the tool, remove old or non-SQLMSTR grants as appropriate.
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 those users (and their group IDs) that 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   SQLMSTR.USERID_GROUP_TAB UG, SQLMSTR.USERID_TAB U,
       SQLMSTR.OBJECT_TAB O, SQLMSTR.GROUP_AUTH_TAB G,
       SQLMSTR.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 28. Query Using Join Statement

APPL GROUP NAME	OBJECT OWNER	OBJECT NAME	S AUTH	USERID	GROUP 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 29. 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.

Chapter 8. DBSPACE Reorganization Tool Tool

The DBSPACE Reorganization Tool tool makes it easy to manage your database servers. Databases are composed of many DBSPACES that are logical allocations of space. DBSPACES can contain one or more tables and their indexes. DBSPACE reorganizations are critical for providing optimum database performance because when you DROP and re-ACQUIRE a DBSPACE, all unused DBSPACE pages are returned to the storage pool for use elsewhere.

Control Center's ability to backup, copy, move, and migrate DBSPACES gives you control and flexibility in managing database growth. It also allows you to extract all of the Data Definition Language (DDL) statements needed to re-create a DBSPACE and everything in it. DBAs no longer have to manage huge libraries of DDL or struggle to produce "where-used" information because Control Center does it for them.

The DBSPACE Reorganization Tool tool operates in Multiple User Mode (MUM) or Single User Mode (SUM). You can choose the mode to run in. (MUM jobs run in one partition while the database is up and running in another, available for other users and applications. A SUM application starts the database. As soon as the database is up, the application program takes control, so both are running in the same partition. Other users cannot access the database until the SUM job ends and the database is restarted in MUM).

Reorganization jobs run in batch and consist of several job steps. Each job step is assigned a step number and description. DLBLs are generated for each step and are included in the JCL so that you know what files are being accessed.

The Control Center screen collects parameters needed by the batch programs. The DBSPACE Reorganization Submit screen, displayed when you press ENTER from the Reorganization screen, allows you to schedule jobs for execution immediately or at a later date and time.

When To Reorganize

Schedule DBSPACE reorganizations and RELOADS during non-peak hours to avoid locking contention with other database users. If you schedule these kinds of jobs during peak hours, against heavy multiple user sessions, you may encounter lock contention when the system catalogs are updated. Running more than one DBSPACE reorganization or RELOAD simultaneously against a single database can also result in catalog contention.

Schedule a DBSPACE reorganization whenever the database statistics indicate that the DBSPACE needs it. For example, when indexes are no longer clustered or when considerable delete activity has occurred, leaving holes of deleted data on DBSPACE pages.

You can also use the DBSPACE Reorganization tool when you need a larger DBSPACE due to growth in the volume of data in the DBSPACE. In addition, use the tool to move DBSPACES to less heavily occupied storage pools. Spreading the distribution of DBSPACES across storage pools helps improve performance.

Moving a DBSPACE can solve a short-on-storage problem and also eliminate the need to add a new dbextent to the database.

When you want to know the characteristics of the columns in a table, use the DBSPACE Reorganization tool's GENERATE DDL option. The generated DDL will show you how all of the objects in the DBSPACE are defined, what indexes exist, who has what authorizations, and what programs access what tables.

Features

The DBSPACE Reorganization tool allows you to:

- Extract and create all DDL required to re-create the DBSPACE and the objects it contains, including:
  - Tables
  - Data
  - Referential Integrity constraints
  - Unique column definitions
  - Indexes
  - Views
  - Grants
  - Table and Column Comments
  - Table and Column Labels
  - Packages (Access Modules)
- Unload DBSPACE data to tape or disk
- Free unused pages by dropping and re-acquiring the DBSPACE
- Load data in clustering index sequence
- Load data with freespace for future inserts
- Rebuild clustered indexes (where possible)
- Update Statistics
- Reprep invalidated access modules
- Reload a DBSPACE to a different database
- Reload a DBSPACE with a different owner
- Reload a DBSPACE with a different DBSPACE name
- Acquire a DBSPACE in a new storage pool
- Acquire a different size DBSPACE
- Change the number of DBSPACE header pages
- Change the free space percent
- Change the index percent
- Change the lock mode
- Run in Multiple or Single User Mode

How the DBSPACE Reorganization Tool Tool Works

When you choose full DBSPACE reorganization, Control Center:

1. Links and establishes communication to the target server.
2. Connects as user SQLREORG.
3. Verifies the availability of the new DBSPACE (if specified).
4. Gathers system catalog information about the specified DBSPACE and creates corresponding DDL statements in the Control Center Database Services Utility (DBSU) command file:
   a. Table create statements
   b. Table comments
   c. Column comments
   d. Table reload statements
   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. Package rebind statements
5. Unloads the DBSPACE data to the specified disk or tape.
6. Executes the SQLDBSU command file from the Database Services Utility to reorganize the DBSPACE and rebind any dependent packages.
7. Updates the SQLMAINT table with the date, time, and duration of the reorganization (see Chapter 9, “DBSPACE Analysis Tools” on page 73).

In order to retain hierarchical dependencies, Control Center issues all grants in the same chronological order in which they were originally issued.

In order to grant authority to an object, the grantor must first connect as the user who originally issued the grant. Therefore, the program must gather database connect passwords for all grantors. If a grantor does not have a connect password, a temporary password is assigned and later removed.

The database server does not remove grant information from the system catalogs when a user is removed from the SYSTEM.SYSUSERAUTH table. Consequently, the REORG job may need to connect as a nonexistent user in order to re-establish a grant. If this situation occurs, Control Center temporarily grants connect authority to the user and later revokes it.

Operational Note: In some cases (such as a reload failure), temporarily granted IDs will not be revoked from the database. You 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).

Using the DBSPACE Reorganization Utility Screen

To reach the DBSPACE Reorganization Utility menu shown in Figure 30, choose Option **2** on the Control Center Main Menu or enter the transaction ID **SQDR** on a CICS screen.

```
     mm/dd/yyyy             CONTROL CENTER V6.1                        hh:mm:ss
*----------------        DBSPACE REORGANIZATION UTILITY       ---------------*
| DATABASE  => DB2VSE61__________                                           |
| OWNER     => ________                                                     |
| DBSPACE   => ____________________                                         |
| FILE      => 1 (1-3)                                                      |
| OPTION    => 3 (1=GENERATE DDL                2=UNLOAD DBSPACE           )|
|               (3=REORGANIZE DBSPACE           4=RELOAD DBSPACE           )|
| ***********************     OPTIONAL PARAMETERS     ***********************|
| DATABASE => ____________________                                          |
| OWNER    => ________                                                      |
| DBSPACE  => ____________________                                          |
|                                                                           |
| PAGES    => _____________ NHEADER        => _ (1-8)      STORPOOL => ___  |
| PCTFREE  => __            ALTER PCTFREE  => __           PCTINDEX => __   |
| LOCK     => _______                                                       |
|                                                                           |
| REBIND PACKAGE => 1 (1=YES/2=NO)          UPDATE ALL STATISTICS => 2 (1/2)|
| COMMITCOUNT    => __________                                              |
| TLBL FILE-ID   => ____________________              DDL STATEMENTS => 1000__|
*-------------------------------------------------------------------- SQC05 ----*

PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT
```

Figure 30. DBSPACE Reorganization Utility Screen

You must enter the first three fields (DATABASE, OWNER, DBSPACEname) to identify the DBSPACE that you want to reorganize. The database specified must be either the default CICS region database or one to which the program may CONNECT. The OWNER and DBSPACEname parameters must identify a valid DBSPACE in the target database.

When you installed Control Center, you defined three SAM DDL files to hold extracted DDL. Provide the number of the file you want to use in the FILE large file.) The FILE number also determines what SAM data file to use if you have not entered a Tape File Name. You do not need to specify the file number if you choose Option **1**, because the DDL is written to the punch queue instead of to a file.

Enter the number of the option you want to execute in the Option field. You can choose to:

Option	**Description**
1 GENERATE DDL	This option extracts from the database, all of the DDL required to re-create a DBSPACE and the objects it contains. Places the DDL in the punch queue for inspection, alteration, or backup.
2 UNLOAD DBSPACE	This option extracts all DDL (as in Option **1**) and writes it to a VSAM file. Then, a DBSU UNLOAD DBSPACE step is executed that writes the

	DBSPACE data to a SAM or tape file. If SAM is selected, the file is REPRO'd to a VSAM file for more permanent retention. The unloaded data and extracted DDL can be used as the basis for a RELOAD DBSPACE (Option **4**) job. An example of an UNLOAD DBSPACE job created to do this is supplied in Figure 69 on page 144.
3 REORGANIZE DBSPACE	This option results in a full DBSPACE reorganization. A jobstream is created that captures the DDL, unloads the DBSPACE, drops, acquires, recreates, and reloads the DBSPACE. Error recovery logic is also included. An example of a REORGANIZE DBSPACE job is supplied in Figure 70 on page 145.
4 RELOAD DBSPACE	This option submits a job to recreate and reload a DBSPACE that has been unloaded from Option **2**. This is basically a DBSPACE recovery facility. An example of the job created to do this is supplied in Figure 71 on page 148.

Each of the options is discussed in more detail below and is accompanied by a sample JCL stream created by the DBSPACE Reorganization tool.

Optional Parameters

All parameters below the "Optional Parameters" line do not require entry.

Parameter	**Description**
DATABASE	Reloads the DBSPACE to a different database. Lets you migrate a DBSPACE from one database to another. For example, you can migrate a DBSPACE from a development database to a production database. Before you migrate the DBSPACE, you may want to ensure that the two databases are compatible so that all reload statements execute successfully. When you use the DATABASE parameter, the DBSPACE in the old database remains unchanged.
OWNER/DBSPACE	Specifies a new owner and/or a new DBSPACEname for the reloaded DBSPACE. When the source DBSPACE and the target DBSPACE are both PRIVATE, the old DBSPACE remains unchanged. If either or both DBSPACES are PUBLIC, the old DBSPACE is dropped prior to creation of the new DBSPACE.
PAGES	Defines a new DBSPACE page size for the reorganized DBSPACE. An empty (unacquired) DBSPACE of the indicated number of pages must be available in the database. If PAGES is not specified, a DBSPACE equal in size to the current DBSPACE is acquired.

NHEADER	Specifies 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 may fail. If you subscribe to the standard of one table per DBSPACE, one header page is sufficient.
STORPOOL	Used to specify a new storage pool for the acquired DBSPACE. This allows you to balance database I/O by spreading the most actively used DBSPACEs over multiple DASD volumes.
PCTFREE	Used to indicate the percentage of each DBSPACE page to be reserved for INSERTS or UPDATES that increase a table's row length. PCTFREE defaults to 10 percent. After the data is reloaded into the DBSPACE, PCTFREE can be altered to zero to make the freespace available.
ALTER PCTFREE	Used to indicate the value to which PCTFREE is to be altered, once the data has been reloaded into the DBSPACE. This value must be lower than the PCTFREE parameter value to have any positive effect.
PCTINDEX	Specifies the ratio of index pages to total DBSPACE pages. Use this parameter to maintain a balance between the number of occupied data and index pages. If not specified, the same ratio as the original DBSPACE will be used.
LOCK	Changes the lock mode of a DBSPACE. Valid values for PUBLIC DBSPACES are DBSPACE, PAGE, and ROW. Private DBSPACES are always locked at the DBSPACE level.
REBIND PACKAGE	Once a DBSPACE has been reloaded, DBSPACE Reorganization rebinds all access modules that are dependent on objects in the DBSPACE. To bypass package rebind processing, specify NO (2). The default value is YES (1).
UPDATE ALL STATISTICS	By default, UPDATE STATISTICS is issued for a DBSPACE once it has been successfully reloaded. UPDATE STATISTICS updates catalog statistics only for columns that appear as the first column in an index. To update catalog statistics for all columns, specify YES (1) for the UPDATE ALL STATISTICS parameter.
COMMITCOUNT	Used to specify the frequency of COMMITS during reload processing. Enter a number in the range 1 through 2147483647 to cause a COMMIT WORK to be executed after an equal number of input rows has been reloaded.

TLBL FILE-ID	Used to specify that data should be unloaded to tape instead of disk. The tape file must have been defined from the WORK FILE LABEL DEFINITION tool. This does not apply to DDL; DDL is ALWAYS unloaded to disk.
DDL STATEMENTS	Allows handling DBSPACES that contain an unusually large amount of DDL (lots of tables, indexes, views, comments). This parameter defaults to 1,000 records; that should be sufficient to handle the vast majority of DBSPACES.

After entering the desired REORG parameters, press ENTER to proceed to the DBSPACE Reorganization Submit screen.

Using the DBSPACE REORGANIZATION Tool

The DBSPACE Reorganization tool can be used in a variety of ways to achieve different goals. Each of the options is discussed in more detail followed by a sample JCL stream produced by the program:

Option 1 - GENERATE DDL

By reading the catalogs, this option generates the DDL necessary to re-create a DBSPACE and all of its associated objects. DDL is written to the VSE/POWER punch queue in the form of DBSU commands and can be used, as is, to redefine the DBSPACE. This option:

- Relieves DBAs from having to maintain large libraries of DDL
- Saves library disk space
- Solves the problem of who owns the "official" DDL
- Provides an easy way to determine table and index characteristics
- Provides authorization and "where-used" information

Figure 31 is an example of the jobstream produced by Control Center to generate DDL for the PUBLIC.SQMHELP DBSPACE.

```
* $$ JOB JNM=GENDDL,CLASS=0,DISP=D,PRI=9
* $$ LST PRI=3
* $$ PUN PRI=3
// JOB GENDDL MUM GENERATE DDL
// OPTION LOG
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0001 GENERATE DDL FOR "PUBLIC"."SQMHELP"
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS006,SYSPCH
// ASSGN SYS011,SYSLST
// EXEC SQB01,SIZE=AUTO
%%DB2VSE61             PUBLIC  SQMHELP           1 N
/*
/&
* $$ EOJ
```

Figure 31. DBSPACE Reorg Option 1 (Generate DDL) - Sample Jobstream

Option 2 - UNLOAD DBSPACE

This option writes the DDL necessary to recreate a DBSPACE to a VSAM file. It then unloads the DBSPACE to a SAM disk file (or a tape if that option was selected). The SAM disk data file is then REPRO'd to a VSAM-managed SAM file for more permanent retention. Data is unloaded in system-defined format; therefore, you must make sure that this data file is not altered prior to reloading the DBSPACE. This option is essentially a DBSPACE backup. Used in conjunction with a RELOAD DBSPACE (Option **4**), it provides the capability to recover from application errors.

Figure 66 on page 137 shows a jobstream that was generated by Control Center to unload the PUBLIC.SQMHELP DBSPACE.

Option 3 - REORGANIZE DBSPACE

This option schedules a full DBSPACE reorganization, including capturing all related DDL (DROP DBSPACE, ACQUIRE DBSPACE, CREATE TABLE, RELOAD DBSPACE) and executing it. In addition, depending on the optional parameters chosen, a DBSPACE can be migrated to another storage pool or another owner. A DBSPACE may also be changed from private to public, or vice versa. The DBSPACE can be moved to another database, as well as have its characteristics, number of pages, percent free space, and percent index changed. This is the most comprehensive option of the reorganization tool.

Figure 67 on page 139 is an example of a jobstream that was generated by Control Center to reorganize the PUBLIC.SQMHELP DBSPACE.

Option 4 - RELOAD DBSPACE

This option submits a job to reload a DBSPACE previously UNLOADED or REORGANIZED using Control Center. The previously created DDL and data files are used to re-create the DBSPACE in its entirety. This option is the recovery counterpart to the UNLOAD DBSPACE option, (Option **2**) and is the method of recovering from an error during a reorganization RELOAD step.

Figure 68 on page 142 shows a sample jobstream that was generated by Control Center to reload the PUBLIC.SQMHELP DBSPACE.

DBSPACE Reorganization Submit Screen

Figure 32 on page 67 shows the DBSPACE Reorganization Submit screen.

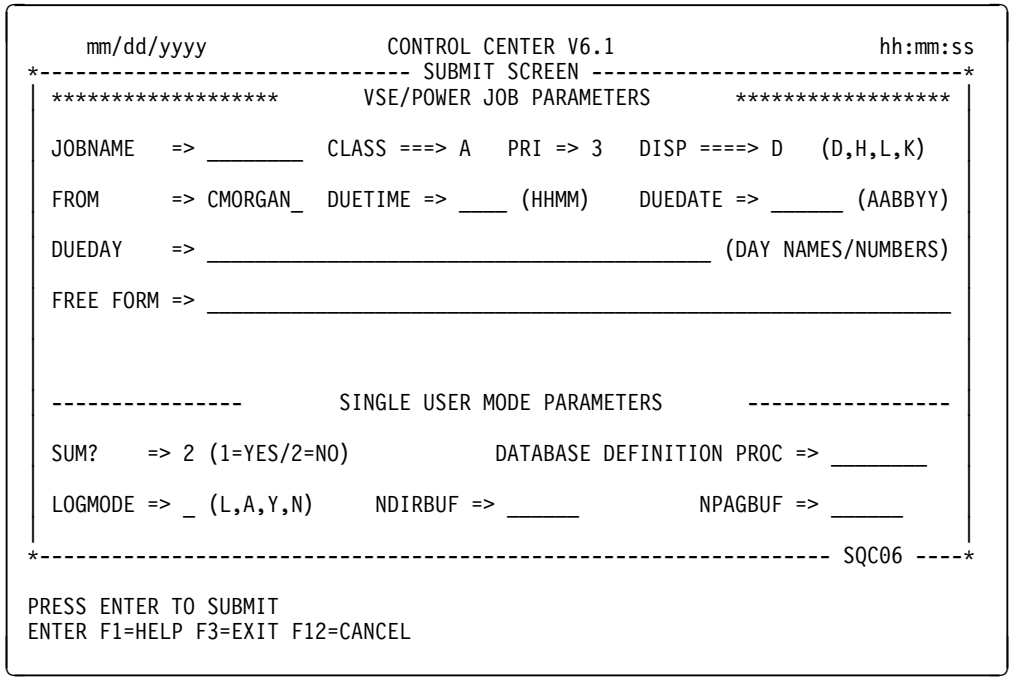

Figure 32. DBSPACE REORGANIZATION SUBMIT screen

To reach this screen, press ENTER from the Control Center DBSPACE REORGANIZATION screen.

The first parameter, JOBNAME, is the only one that is required.

Parameters

Parameter	Description
JOBNAME	Specifies the name by which the DBSPACE REORGANIZATION job and its associated queue entries is to be known.
CLASS	Specifies the class or partition in which you want this job to run. Class defaults to A.
PRI	Specifies the priority that is to be assigned to the job. Specify a number from 0 to 9 where 9 is the highest priority. Default priority is 3.
DISP	Specifies how the job is to be handled in the reader queue. Disposition may be specified as: • D - Delete after processing • H - Hold until released • K - Keep after processing • L - Leave in the queue Disposition defaults to D.
FROM	Specifies the ID of the user being allowed to manipulate or retrieve the job. Defaults to the CICS user ID.

DUETIME	Specifies the processing start time using hh for hour and mm for minute in 24-hour clock time.
DUEDATE	Specifies the processing date using YY for year. Depending on the format defined for your system, AA is month and BB is day, or AA is day and BB is month.
DUEDAY	Specifies the day(s) the job is to be scheduled. You may enter a day name abbreviation such as MON for Monday, or a list separated by commas and enclosed in quotes. You may also enter the day of the month or a list of day numbers separated by commas and enclosed in quotes. You may also specify DAILY to schedule the job every day of the year.
OTHER	The VSE/POWER * $$ JOB card offers many parameters that do not appear on the DBSPACE REORGANIZATION SUBMIT screen. Use this field to have Control Center include those parameters when the job is submitted.

After entering the desired submit parameters, press ENTER to submit the job to VSE/POWER. For more information on VSE/POWER jobs, refer to the *VSE/POWER Installation and Operations Guide*.

Figure 33 is a sample of the DBREORG Report.

```
SQB02                        CONTROL CENTER FOR VSE
hh:mm:ss                DBSPACE REORGANIZATION REPORT            mm/dd/yyyy
                   DBSPACE 1                    DBSPACE 2
                   ------------------           ------------------
DATABASE:          DB2VSE61                     DB2VSE61
OWNER:             PUBLIC                       PUBLIC
DBSPACENAME:       SQMHELP                      SQMHELP
            BEFORE REORG STATISTICS                   AFTER REORG STATISTICS
-----------------------------------     -----------------------------------
DBSPACENO:                          12                                    12
POOL:                                1                                     1
NPAGES:                            128                                   128
NRHEADER:                            1                                     1
PCTINDX:                            33                                    33
FREEPCT:                             0                                     0
LOCKMODE:                         PAGE                                  PAGE
NACTIVE:                            39                                    39
NTABS:                               1                                     1
            ELAPSED TIMES IN MINUTES
-----------------------------------
UNLOAD DBSPACE:             00:00:07
RELOAD DBSPACE:             00:00:08
TOTAL ELAPSED TIME:         00:00:15
SQLMAINT TABLE HAS BEEN SUCCESSFULLY UPDATED.
```

Figure 33. DBSPACE Reorganization Report

Single User Mode (SUM) DBSPACE Reorganization

You can choose to run a DBSPACE reorganization in Multiple User Mode (MUM) or in Single User Mode (SUM). In SUM, contention with other applications and users is eliminated. Storage used to support those users can be used to define additional directory or page buffers, resulting in better performance.

In SUM, you can bypass logging by specifying LOGMODE N. However, switching to logmode N will probably require an archive and a coldlog before the switch and another archive before switching back.

Before You Choose Single User Mode Execution

Review the *DB2 Server for VSE Database Administration* manual to understand Single User Mode database execution. Also, review the topics on choosing a logmode and switching logmodes. Control Center Single User Mode parameters are listed below:

Single User Mode Parameters

Parameter	Description
SUM?	Specify "1" (YES) to cause a Single User Mode job to be submitted. This parameter defaults to "2" (Multiple User Mode).
DATABASE DEFINITION PROC	Specifies the name of the procedure that contains the job control statements (DLBLs) required to access the database. This parameter need not be entered if the job control statements have been loaded into standard labels.
LOGMODE	Specifies the logmode you want Control Center to use during Single User Mode processing. You must enter a value. Valid values are: • A - All database changes are logged and regular database archives are maintained. • L - All database changes are logged and regular log archives are maintained. • N - No database changes are logged. • Y - All database changes are logged but no archives are maintained.
NDIRBUF	The number of 512-byte directory pages to be kept in storage. The bigger this value is, the better your database will perform until you run out of storage or cause excessive paging. NDIRBUF defaults to 14. This parameter is not required.
NPAGBUF	The number of 4096-byte data pages to be kept in storage. Again, bigger is better, within reason. NPAGBUF also defaults to 14. Entry of this parameter is not required.

SUM processing requires that the database be ended prior to execution. When a job step requires access to a database, the database is started and immediately branches to the application program. When the application program ends, control is passed back to the database server and the database is ended. The database remains down until it is restarted. Remember that changing logmode will probably force some combination of coldlogs, and log or database archives.

Figure 72 on page 150 is an example of a Single User Mode REORGANIZE DBSPACE (Option **3**).

DBSPACE Reorganization Tape Support

Unloading to Tape

When you specify a TLBL FILE-ID on the DBSPACE REORGANIZATION UTILITY screen, tape is used as the unload media. As a result, the jobstream that Control Center builds and submits is quite different. Figure 73 on page 153 is an example of a REORGANIZE DBSPACE (Option **3**) from tape.

Special Considerations

Repetitive Scheduling

If a DBSPACE reorganization job is scheduled to be run on a repetitive basis (such as each week on Thursday night), be aware that an SQMPARM file record is created when the REORG job is scheduled. This record contains parameters used by the REORG process. The same record will be used each time the DBSPACE is reorganized. If an intervening REORG job is scheduled from Control Center, a new SQMPARM record will be generated based upon the parameters chosen at that time. These may be different from the ones previously chosen for the scheduled job. This means that the new parm record will be used for all subsequent executions of the scheduled job. If this is not what you want, delete the scheduled job from the VSE/POWER reader queue and schedule a new one.

Failure Restart

The job listing from your Control Center jobs will indicate whether the job ended successfully. Return code checking and conditional JCL are used to support failure restart. If a DBSPACE reorganization fails prior to the reload step, the DBSPACE has not been changed and the job can be restarted from the beginning. If the failure occurs during the reload step, the function can be restarted using RELOAD DBSPACE (Option **4**).

In all cases, view the output job listing to determine the cause of the error and whether it requires fixing. In many cases, minor errors occur but the job is able to complete successfully.

Problem Analysis

During DDL generation, SQL statements are used to capture information from the database manager system catalogs. If a serious database error is encountered, a descriptive error message and all pertinent information from the SQL Communication Area is displayed on the job listing.

The DBSPACE REORGANIZATION tools use a DBSU command file to execute the UNLOAD DBSPACE portion of the job. Detailed output from the UNLOAD portion is displayed on the job listing. Examine the listing to determine the reason for failure.

During RELOAD processing, DBSPACE REORGANIZATION jobs invoke a DBSU RELOAD. Detailed output of this process is displayed in the job listing. If a failure occurs during the RELOAD, the listing can be examined to determine the cause of failure.

One common problem to be aware of is a possible LOG FULL condition that may occur during RELOAD processing. The DBSU RELOAD TABLE command executes as a single LUW, meaning that the entire RELOAD could be rolled back if an error occurs. The database server would then have to record the LUW in the LOG. If the target table is large, or the database LOG file was nearly full when the reload began, the possibility of a LOG FULL condition exists. Depending on logmode, the database server will attempt to perform a database archive, a log archive, or a checkpoint in the LOG. If the RELOAD process continues until the LOG is completely full, the database server will begin to ROLLBACK the entire RELOAD.

Since the DROP DBSPACE has already been COMMITTED, 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 was nearly full prior to the reload, you could perform a database archive, a log archive, or a coldlog (depending on whether you are using logmode A, L, or Y respectively). After this completes, you can complete the reload by initiating a RELOAD DBSPACE (Option **4**).

If the RELOAD LUW exceeds the LOG size, even when empty, you have two options:

1. Increase the size of the LOG file, then complete the reorganization.
2. Run the RELOAD in SUM with logmode N (no logging).

Chapter 9. DBSPACE Analysis Tools

About the DBSPACE Analysis Tools

When an application or user requests information from a database or server, the OPTIMIZER uses catalog statistics to choose the most efficient access path to the data. These statistics are not automatically updated every time a row is inserted, updated, or deleted because of the overhead that would be involved. However, the more accurate the statistics, the better able the optimizer is to choose an efficient method of retrieving data, thereby improving overall database performance. Every DBA needs to ensure that statistics are updated on a regular basis to maximize database performance.

Another common database problem that impacts performance is the gradual fragmentation and disorganization of data over time. After many inserts, updates, and deletes, the data in a database becomes fragmented - spread out over many physical pages, with many gaps and with a physical sequence much different than the logical sequence of data. Reorganization is required to reload the data onto a minimum number of physical pages in a physical sequence that optimizes logical data retrieval.

The DBSPACE Analysis tools help you to analyze DBSPACES and perform maintenance on them to improve performance. These tools are designed to allow you to specify all DBSPACES or a subset of the DBSPACES in the database. You are also given the opportunity to choose what criteria are to be used to select candidates for maintenance.

There are two basic DBSPACE Analysis tools: one for Update Statistics analysis and another for DBSPACE Reorganization analysis.

Before You Begin

You can analyze DBSPACES at any time since minimal calls are made to the database catalog. Based on the results of the analysis, the actual maintenance jobs can then be scheduled for off-hours, or as needed.

Whenever possible, schedule the actual maintenance during non-peak hours to prevent locking contention with other database users. Extensive updating of the database system catalogs occurs during UPDATE STATISTICS and DBSPACE reorganizations. During periods of high database usage, this can lead to lock contention.

How the DBSPACE Analysis Tools Work

The DBSPACE Analysis tools help keep DBSPACES tuned by keeping track of the UPDATE STATISTICS and reorganization activities at the DBSPACE level and by executing these functions where and when required based on specified execution parameters. This data is stored in a database table (SQLMSTR.SQLMAINT) that is created during Control Center installation.

When you select Option **2** from the DBSPACE Analysis Utilties screen (Figure 35 on page 76) to list UPDATE STATISTICS candidates or Option **3** to list REORG

candidates, a detailed analysis of each DBSPACE is conducted to determine the degree of need for maintenance. The results of this analysis are provided on a display screen for study by the DBA. The DBA can then optionally choose what DBSPACES to reorganize or to run UPDATE STATISTICS against.

The DBSPACE Analysis tool uses a database table (SQLMSTR.SQLMAINT) to maintain usage information for each DBSPACE in the database. During execution, each DBSPACE is considered for maintenance based on parameters specified by you. After execution, the information in the SQLMAINT table is updated to reflect the changes that have occurred.

When you invoke the DBSPACE Analysis tool, it:

1. Displays an analysis selection screen.
2. Optionally refreshes the SQLMAINT table with data from SYSTEM.SYSDBSPACES.
3. Selects DBSPACES that match the selection parameters for UPDATE STATISTICS or reorganization.
4. Displays a list of candidates chosen for maintenance.
5. Optionally submits maintenance jobs for the candidates selected.
6. Updates the SQLMAINT table for each DBSPACE that receives maintenance.

The selection process consists of 1 step for UPDATE STATISTICS analysis and 2 steps for the reorganization analysis. The steps are:

1. Select DBSPACES from the SQLMAINT table using a first set of selection parameters such as name, size, and time. (See “Selection Options” on page 77).
2. For reorganization, apply REORG CRITERIA to the list selected in step 1 above to come up with a list of candidates that need reorganization. (See “DBSPACE Reorganization Criteria (CRITERIA)” on page 77).

SQLMAINT Table

The product installation process creates SQLMAINT in a public DBSPACE in the target database as shown in Figure 34.

```
CREATE TABLE "SQLMSTR"."SQLMAINT"
( "OWNER"          CHAR(8),
  "DBSPACENAME"    CHAR(18),
  "DBSPACENO"      SMALLINT,
  "FREEPCT"        SMALLINT,
  "PCTINDX"        SMALLINT,
  "UPSTAT_DATE"    DATE,
  "UPSTAT_TIME"    TIME,
  "UPSTAT_ELAPSED" TIME,
  "REORG_DATE"     DATE,
  "REORG_TIME"     TIME,
  "REORG_ELAPSED"  TIME,
  "REORG_FREEPCT"  SMALLINT,
  "REORG_PCTINDX"  SMALLINT,
  "REORG_STATUS"   CHAR(2),
  "REORG_WEIGHT"   SMALLINT,
  "NPAGES"         INTEGER )
IN "PUBLIC"."SQLMAINT";
```

Figure 34. SQLMAINT Table Definition

SQLMAINT can be used as a basis for creating your own maintenance reports. You can also update it manually to further control the maintenance process (such as changing the reorganization date to prevent a large DBSPACE from being selected for reorganization).

The columns that make up SQLMSTR.SQLMAINT are described below:

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 job
REORG_DATE	Date of the last reorganization
REORG_TIME	Time of the last reorganization
REORG_ELAPSED	Elapsed time of the last reorganization job
REORG_FREEPCT	FREEPCT value in the last reorganization
REORG_PCTINDX	PCTINDX value in the last reorganization
REORG_STATUS	A two-character reorganization status indicator
REORG_WEIGHT	An integer weight of the need for reorganization
NPAGES	Size of the DBSPACE

Function 1 of the DBSPACE Analysis Tool (Initialize Control Table) inserts rows for all private and non-system owned public DBSPACES into the SQLMAINT table. All DATE columns are initialized to "0001-01-01". All TIME columns are set to "00:00:00". The table is then ready for use with the other DBSPACE Analysis tool functions.

DBSPACE Analysis Utility Screen

Figure 35 on page 76 shows the DBSPACE Analysis Utilities screen.

```
     mm/dd/yyyy              Control Center V6.1                      hh:mm:ss
*---------------------- DBSPACE ANALYSIS UTILITIES -------------------------*
| DATABASE => DB2VSE61
| OPTION ===> 3                        REFRESH         => 1 (1=YES 2=NO)
|
| 1=INITIALIZE CONTROL TABLE           2=UPDATE STATISTICS CANDIDATES
| 3=LIST REORG CANDIDATES
|
| *************************** SELECTION OPTIONS    ************************
|
| OWNER         ===> ALL               DBSPACENAME     => ____________________
|
| MIN(PAGES)     ==> 0                 MAX(PAGES)      => 9999999
|
| DAYS SINCE ======> 14
|
| CRITERIA       ==> 1
|
| 1=CLUSTERRATIO < 9999                2=UNCLUSTERED INDEX
| 3=NOVERFLOW    >  1 %                4=APPLY ALL CRITERIA
*------------------------------------------------------------------- SQC08-----*

PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT
```

Figure 35. DBSPACE Analysis Utilities Screen

Most fields are initialized to some value. The first two parameters (DATABASE and OPTION) are required. They identify the function you want to perform and what database the function will be performed upon. These fields are initialized to Option **3** (List REORG Candidates) and the database to which Control Center is currently connected.

Functions

The option field can have the following values:

Function	Description
1 INITIALIZE CONTROL TABLE	Rebuilds the SQLMAINT control table in real time; not as a scheduled job. All previous maintenance data will be erased and a row for each non-system owned DBSPACE will be inserted into the table.
2 UPDATE STATISTICS CANDIDATES	Lists the DBSPACES that need their statistics updated. The candidate list is displayed based upon the other selection options chosen.
3 LIST REORG CANDIDATES	This function lists candidate DBSPACES that require reorganization based on the specified criteria. It lists each candidate and advises the DBA of the need for reorganization. This is the default option when the screen is first displayed.

The REFRESH parameter (defaults to YES) updates the SQLMAINT table:

1. Adding rows for any DBSPACES that are not in the SQLMAINT table.
2. Deleting rows for any DBSPACES that are not in SYSTEM.SYSDBSPACES.

This is done in real time PRIOR to performing the analysis selected. Other DBSPACE entries are not affected and their maintenance data remains intact.

Selection Options

All parameters below the line labeled "SELECTION OPTIONS" allow you to control the maintenance activity by restricting the number and type of DBSPACES that will be selected for analysis. In the case of the Update Statistics tool, the selected DBSPACES will all be candidates. In the REORG tool, candidates retrieved from the SQLMAINT table then have the reorganization CRITERIA applied against them and a final status is presented that states whether the DBSPACE is a REORG candidate.

Parameter	Description
OWNER	Is used to specify whether PUBLIC, PRIVATE, ALL, or specific DBSPACE owner(s) will be selected for analysis. The DB2 Server for VSE wildcard character (%) can be used at the beginning and/or end of a specified DBSPACE OWNER to select "like" DBSPACE OWNER names.
DBSPACENAME	Is used to specify a single DBSPACE name or a group of similar DBSPACE names that should be selected for analysis. The DB2 Server for VSE wildcard character (%) can be used at the beginning and/or end of the specified DBSPACE name to select "like" DBSPACE names. This parameter defaults to choosing all DBSPACES owned by the specified DBSPACE OWNER.
PAGES	Is used to specify the MINIMUM and MAXIMUM DBSPACE sizes (expressed in pages) to be selected for analysis.
DAYS SINCE	Specifies the number of days that must have passed since the last UPDATE STATISTICS or reorganization before the DBSPACE can be considered for candidate analysis.

DBSPACE Reorganization Criteria (CRITERIA)

This parameter applies an additional test to selected DBSPACES to determine whether they are candidates for reorganization. After the selection options above have been used to select DBSPACE names from the SQLMAINT table, the reorganization CRITERIA chosen are checked against those DBSPACES and the results are displayed on a list screen (see Figure 37 on page 80). Those DBSPACES meeting the criteria should be considered candidates for reorganization. You can select them from the REORG CANDIDATES LIST screen for job submission and scheduling.

You can select one of four different methods of analyzing the need for reorganization in the list of retrieved DBSPACES. They are:

Criteria	Description
1 CLUSTERRATIO < nnnn	If the cluster ratio of any index in a DBSPACE is less than the stated value, the DBSPACE is considered a candidate for reorganization, regardless of the value of the CLUSTER field. You may specify a value from 1 to 9999.

2 UNCLUSTERED INDEX	If there are any unclustered indexes in the tables in the DBSPACE, the DBSPACE is identified as a candidate.
3 NOVERFLOW ROWS > n%	If the number of overflow rows for any of the tables in the DBSPACE exceeds the value you entered, the DBSPACE is selected as a candidate.
4 APPLY ALL CRITERIA	If any of the above three criteria apply, the DBSPACE is chosen as a candidate for reorganization.

CLUSTER RATIO < 9999 is the default criterion.

Update Statistics Analysis Tool

The Update Statistics Analysis tool can be selected from the DBSPACE ANALYSIS UTILITIES screen by choosing Option **2** (Update Statistics Candidates).

The purpose of this tool is to list the DBSPACES that need to have their statistics updated and to optionally submit batch jobs to update them.

An example of the list generated by the Update Statistics Analysis Tool is shown in Figure 36 on page 79. The list was created by specifying:

```
OPTION ===> 2

DBSPACE OWNER ===> PUBLIC__

PAGES: MINIMUM ==> 0______            MAXIMUM ========> 512____

DAYS SINCE ======> 14_
```

```
     mm/dd/yyyy              CONTROL CENTER V6.1                      hh:mm:ss
*------------------------ UPDATE STATISTICS LIST ----------------------------*
| DATABASE => DB2VSE61                                                       |
|                                                                            |
|   OWNER    DBSPACENAME        LAST DATE   ELAPSED   NPAGES                |
|                                                                            |
| _ PUBLIC   ANDY               1996-11-19  00:01:00     128                 |
| _ PUBLIC   CONNIE             1996-11-18  00:02:30     128                 |
| _ PUBLIC   FRED               1996-10-30  00:01:05     128                 |
| _ PUBLIC   MARTIN             1996-11-19  00:00:50     128                 |
| _ PUBLIC   SMITH              1996-11-17  00:03:14     256                 |
| _ PUBLIC   RAY                1996-11-17  00:02:45     512                 |
| _ PUBLIC   TAYLOR             1996-11-18  00:01:30     512                 |
| _ PUBLIC   SQLMSTR            1996-11-18  00:00:45     128                 |
| _ PUBLIC   SQMHELP            1996-10-29  00:01:20     256                 |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
*---------------------------------------------------------------- SQC08 ----*

USE 'X' TO SELECT DBSPACE(S); PRESS ENTER TO SUBMIT
ENTER F1=HELP F3=EXIT F12=CANCEL
```

Figure 36. Update Statistics Analysis List Screen

Note that on this screen the following are displayed:

1. The DBSPACE OWNER and NAME.
2. The DATE of the last Update Statistics for this DBSPACE.
3. The ELAPSED time (in hours:minutes:seconds) of the last Update Statistics job.
4. The size of the DBSPACE.

From this screen, you can enter an "X" next to the DBSPACE(s) for which an UPDATE STATISTICS job is desired. Then, by pressing ENTER, the DBSPACE ANALYSIS SUBMIT screen is displayed and a job can be scheduled for execution. When multiple DBSPACES are selected, multiple jobs are submitted, each having the same job name and parameters. (See “DBSPACE Analysis Submit Screen” on page 81).

Pressing F12 will take you back to the previous screen (DBSPACE Analysis Utilities). Pressing F3 will take you back to the Control Center main menu.

DBSPACE Reorganization Analysis Tool

This function lists the DBSPACES that need to be reorganized. The selection process is conducted by evaluating information in the database catalogs and comparing it to the criteria selected by you.

There are three criteria that are evaluated by the program, any of which may cause a DBSPACE to be selected for reorganization. You can choose any one or all three to be used by the Analyzer tool:

1. The existence of unclustered primary indexes on tables within the DBSPACE. This will typically occur when many inserts and deletes have been performed. It greatly reduces the performance of the database.
2. Overflow pages greater than n% for any table in the DBSPACE. This condition indicates a great deal of fragmentation, caused by updates that have increased the row length. The number of I/O operations required to return selected rows is greatly increased.
3. A CLUSTER RATIO value below the threshold specified by the user.
4. All of the criteria above.

An example of the list generated by the Reorganization Analysis Tool is shown in Figure 37. The list was created by specifying:

```
OPTION ===> 3

DBSPACE OWNER ===> PRIVATE_

PAGES: MINIMUM ==> 0_____            MAXIMUM =========> 8192___

DAYS SINCE ======> 14_

REORG CRITERIA ==> 4
```

```
     mm/dd/yyyy              Control Center V6.1                      hh:mm:ss
*------------------------ REORG CANDIDATES LIST -----------------------------*
| DATABASE => DB2VSE61                                                       |
|                                                                            |
|   OWNER    DBSPACENAME        LAST DATE   ELAPSED   NPAGES STATUS         |
|                                                                            |
| _ ANDY     TAYLOR             0001-01-01  00:00:00     128                |
| _ CONNIE   SMITH              1996-11-18  00:01:35     128 NO REORG NEEDED|
| _ FRED     TAYLOR             1996-11-10  00:01:38     128 OVERFLOW ROWS  |
| _ MARTIN   MARTIN             1996-11-17  00:02:40     128 NO REORG NEEDED|
| _ MASALI   MASALI             1996-11-17  00:03:00    5120 CLUSTER RATIO  |
| _ RAY      MARTIN             1996-11-17  00:02:15    1024 UNCLUSTERED    |
| _ TAYLORA  TAYLOR             1996-11-12  00:01:50     128 NO REORG NEEDED|
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
|                                                                            |
*---------------------------------------------------------------- SQC08 ----*

USE 'X' TO SELECT DBSPACE(S); PRESS ENTER TO SUBMIT
ENTER F1=HELP F3=EXIT F12=CANCEL
```

Figure 37. DBSPACE Reorganization Analysis List Screen

Figure 37 is an example of the Reorganization Analysis List created by the tool. Note that on this screen the following are displayed:

- the DBSPACE OWNER and NAME.
- the DATE of the last reorganization run for the DBSPACE.

- the ELAPSED time (in hours:minutes:seconds) of the last reorganization job.
- the size of the DBSPACE.
- the result of the analysis.

From this screen, you can place an "X" next to the DBSPACE(s) for which reorganization is desired. Then, by pressing ENTER, the DBSPACE ANALYSIS SUBMIT screen is displayed and a reorganization job can be scheduled for execution. Only one DBSPACE is reorganized in a single batch job. (See Figure 32 on page 67 for details concerning the job submission screen).

If more than one DBSPACE is selected, the SUBMIT screen is displayed for the first DBSPACE selected. Once a job is submitted, the LIST screen is re-displayed, with that DBSPACE identified with an asterisk "*" in the select field. If ENTER is pressed again, the SUBMIT screen is displayed for the next DBSPACE with an "X" in the select field. This process is repeated until a reorganization job has been submitted for every DBSPACE selected (unless, of course, F3, F12, or a scroll function key is chosen by you).

In the sample display, seven DBSPACES were analyzed and three were selected for reorganization. Four DBSPACES were discounted because they did not meet the criteria selected. Note that one DBSPACE has never been reorganized. Its date and elapsed time fields have initial values. ANDY.TAYLOR is a new entry in the SQLMAINT table.

By pressing F12, you are returned to the previous screen (DBSPACE Analysis Utilities). F3 returns you to the main menu.

DBSPACE Analysis Submit Screen

Figure 38 on page 82 shows the DBSPACE Analysis SUBMIT screen.

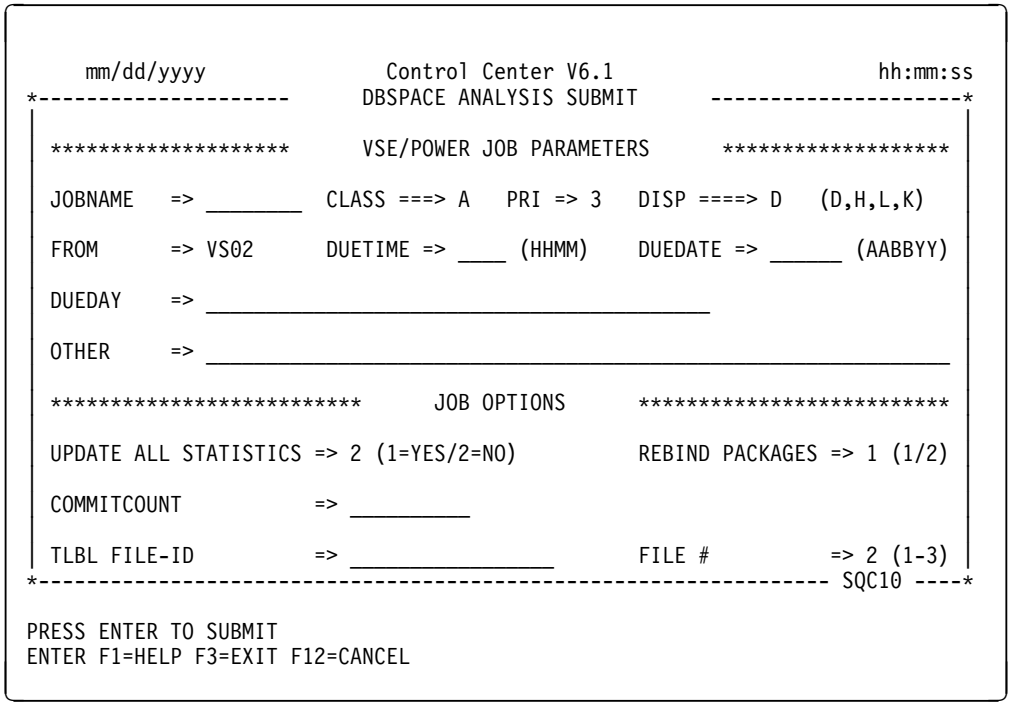

Figure 38. DBSPACE Analysis Submit Screen

Refer to the *IBM VSE/POWER Installation and Operations Guide* for use of the job submit parameters. The parameters in the JOB OPTIONS section are for use by the individual Update Statistics or reorganization jobs. These are explained below. Note that only the UPDATE ALL STATISTICS parameter is applicable to the Update Statistics function.

Job Options

Parameter	Description
JOBNAME	Specifies the name by which the DBSPACE ANALYSIS job and its associated queue entries is to be known.
CLASS	Specifies the class or partition in which you want this job to run. Class defaults to A.
PRI	Specifies the priority that is to be assigned to the job. Specify a number from 0 to 9 where 9 is the highest priority. Default priority is 3.
DISP	Specifies how the job is to be handled in the reader queue. Disposition may be specified as: • D - Delete after processing • H - Hold until released • K - Keep after processing • L - Leave in the queue Disposition defaults to D.

FROM	Specifies the ID of the user being allowed to manipulate or retrieve the job. Defaults to the CICS user ID.
DUETIME	Specifies the processing start time using hh for hour and mm for minute in 24-hour clock time.
DUEDATE	Specifies the processing date using YY for year. Depending on the format defined for your system, AA is month and BB is day, or AA is day and BB is month.
DUEDAY	Specifies the day(s) the job is to be scheduled. You may enter a day name abbreviation such as MON for Monday, or a list separated by commas and enclosed in quotes. You may also enter the day of the month or a list of day numbers separated by commas and enclosed in quotes. You may also specify DAILY to schedule the job every day of the year.
OTHER	The VSE/POWER * $$ JOB card offers many parameters that do not appear on the DBSPACE ANALYSIS SUBMIT screen. Use this field to have Control Center include those parameters when the job is submitted.
UPDATE ALL STATISTICS	This parameter is used to specify whether an UPDATE ALL STATISTICS command should be executed instead of merely an UPDATE STATISTICS. It applies to Update Statistics jobs as well as the reorganization jobs. The default value is NO.
REBIND PACKAGES	Enter 2 (NO) if the DBSPACE Reorganization tool should *not* rebind packages as part of a DBSPACE reorganization. The default is 1 (YES), which means rebinding will occur.
TAPE FILE NAME	This parameter identifies a tape file name that is to receive the output of the UNLOAD, rather than a disk file. The file name should have been defined to Control Center/VSE using the Work File Label Definition function.
FILE #	This parameter defines which of the previously defined data and DDL files (See “How the Work File Label Definition Tool Works” on page 94) is to be used for the unloaded data and DDL. Valid values are 1 to 3.

Additional Topics

Initial Execution

Since the SQLMAINT table initially contains no maintenance history, all DBSPACES will be selected when analysis is done the first time. If it is desired to limit the number of DBSPACES returned in the Analysis Display List, the last date field for either Update Statistics or REORG (or both) can be modified to inhibit selection of those DBSPACES for candidate processing. This can be done manually using ISQL or DBSU.

For example, the REORG_DATE and UPSTAT_DATE columns in the SQLMAINT table can initially be set to different values so that a single Analysis run will only consider a portion of the DBSPACES. You can execute database command such as:

```
UPDATE SQLMSTR.SQLMAINT SET UPSTAT_DATE = '1998-12-31', REORG_DATE
= '1998-12-31' WHERE OWNER < 'M'
```

This may 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.

Reorganization Work Space

The same considerations for TAPE FILE NAME and FILE # apply whether the reorganization is accomplished using the DBSPACE Reorganization or the Analysis Tool.

Chapter 10. Package Utility

Introduction

The Control Center Package Utility assists DBAs by automating four tasks very often associated with managing database packages. A package is a control structure containing SQL statements in executable form. It is produced as part of the preparation process before a program is compiled. Packages are stored in the database.

When a program is migrated from test to production, very often it is not recompiled. Instead, the executable PHASE is simply copied from the test to the production library. If the program contains SQL statements, the associated package must also be copied to the production database.

The Control Center Package Utility helps in this effort by allowing you to unload a package from one database and reload it into another. The unload/reload utilities are also handy for migrating packages in a distributed processing environment.

Whenever an object such as a DBSPACE, table, or index is dropped, all packages dependent upon that object are marked invalid. Before an invalid package can be executed, the database must first rebind it. This can result in poor response time for the first person attempting to execute the program. The Control Center Package Utility allows DBAs to rebind packages ahead of time so that their users are not impacted.

Many times, DBAs are called upon to help application developers identify poor performing SQL statements in their programs and suggest ways the SQL can be recoded to improve performance. The Control Center Package Utility helps in this area by producing a package report which, among other things, lists every SQL statement in the program. The Control Center package report can be used to tune the SQL in the program and becomes an important part of the documentation about the program.

Package Utility Functions

The Package Utility offers four main functions:

- Unload Package
- Reload Package
- Rebind Package
- View Package

The Unload, Reload, Rebind, and View Package Utilities all use DBSU to accomplish the desired process. For each utility, Control Center generates the appropriate DBSU commands and includes the JCL to manage their execution. The Unload and Reload functions use LIBR, the VSE librarian, for storage and retrieval of unloaded packages. The View Package Utility uses a REXX/VSE program to produce a printed package report.

Package Function Descriptions

Unload Package

The Unload Package function consists of three steps:

- Collect package parameter
- Collect job submission parameters and submits job
- Execute the DBSU to Unload the Package

In the first step, you identify the package to be unloaded by specifying the owner and package name. Optionally, a server name other than the one that is displayed can be specified. If the package does not exist, you will be notified.

The second step is where the VSE/POWER job parameters are built and a two step batch jobstream is submitted.

The third step executes DBSU to unload the package to the VSE punch queue. It then catalogs the punch file containing the package into a library where it can be accessed for reloading.

Reload Package

The Reload Package function consists of three steps:

- Collect package parameter
- Collect job submission parameters and submits job
- Execute the DBSU to Reload the Package

In the first step, you identify the package to be reloaded by specifying an owner and package name. You many specify a server name other than the one that is displayed. Options such as whether an existing package is to be replaced and whether to keep or revoke existing run privileges are supported.

The next step collects VSE/POWER job parameters and builds and submits a two step batch jobstream.

The last step punches the package to the VSE reader queue from the VSE library where it was cataloged. It then executes DBSU to reload the package from the VSE reader queue.

Rebind Package

The Rebind Package function consists of three steps:

- Collect package parameter
- Collect job submission parameters and submits job
- Execute the DBSU to Rebind the Package

In the first step, you identify the package to be rebound by specifying an owner and package name. You may specify a server name other than the one that is displayed. Optionally, you may specify that the package is only to be rebound if it is already marked as invalid.

The second step collects VSE/POWER job parameters and builds and submits a single step batch jobstream.

The final step executes DBSU to rebind the package.

View Package

The View Package function consists of three steps:

- Collect package parameter
- Collect job submission parameters and submits job
- Copy, Unload and Display the Package Contents

Initially, you identify the package to be viewed by specifying an owner and package name. You may specify a server name other than the one that is displayed.

Next, the VSE/POWER job parameters are collected and a three-step batch jobstream is built and submitted.

In the last step, the package messages are copied to a SAM file where they are accessible to the REXX/VSE package report program. A DBSU job is then executed to unload the package to a SAM file, followed by the execution of a REXX/VSE program to produce the package report.

Package Migration

For package migration, both the Unload and Reload Package functions previously described are used. Be sure to specify the **Server** name field (Figure 39 on page 88) when invoking the Reload function. During the job submission step, the parameter **DEST** must be specified. See Package Utility Parameters and “Package Utility Job Submit Parameters” on page 89.

Invocation

There are two ways to invoke the Package Utility:

- From the main menu, by selecting Option **7** (PACKAGE UTILITY), or
- Directly from CICS, by typing the Package Utility transaction ID, SQPM

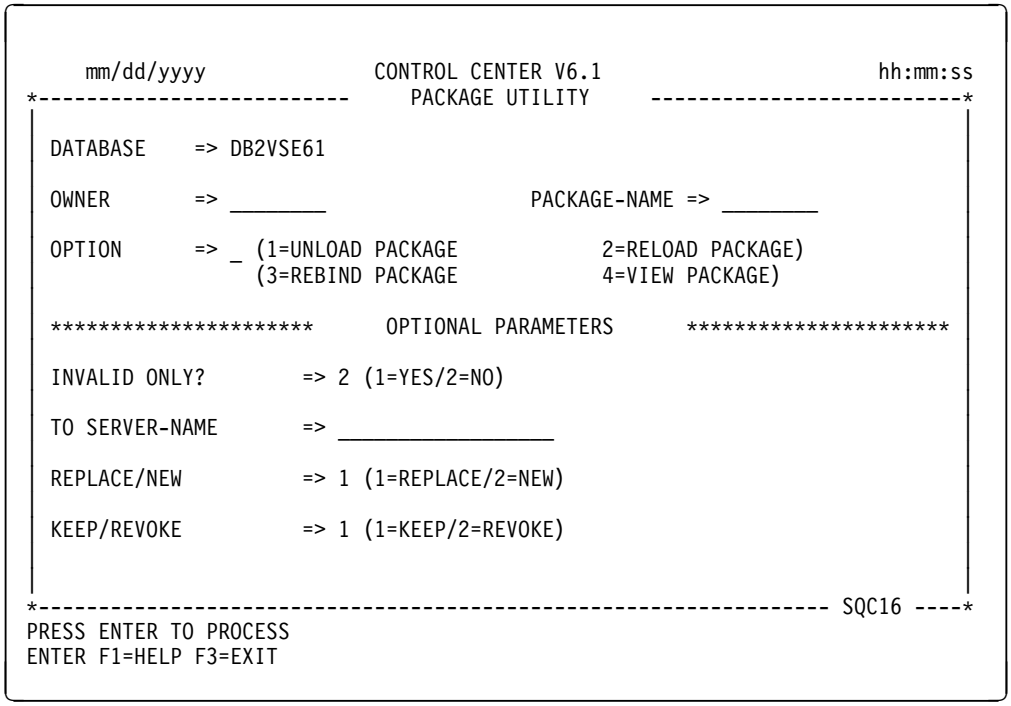

Figure 39. Package Utility Screen

Package Utility Parameters

The Package Utility parameters are described below.

Entry Field	Description
DATABASE	the name of the default or last used application server. If you enter the name of another server, you will be connected to it before proceeding.
OWNER	the ID of the person who created this package or preprocessed the program associated with it.
PACKAGE-NAME	the name of the package.
OPTION	a subfunction list
INVALID ONLY?	You may not want to rebind a package that is already valid. If so, specify YES. The Package Utility will check to make sure the package is invalid before allowing a rebind job to be submitted. The default value is NO (2).
TO SERVER-NAME	Use this parameter to reload a package to a different application server. DBSU will connect to the application server and reload the package. This parameter applies only to reload package.
REPLACE/NEW	Specify REPLACE if an existing package is to be replaced by the reload. If the package does not exist, a new package will be created. NEW causes the reload to fail if a package already exists. REPLACE is the default.

KEEP/REVOKE	KEEP specifies that run authorizations are to remain in effect when the package is reloaded. REVOKE causes all existing run authorizations to be revoked. KEEP/REVOKE are only allowed with REPLACE. KEEP is the default.

Using the Package Utility

The examples below will illustrate how to use the Unload and View Package functions.

How to Unload a Package

To Unload a package, select Option **1** on the Package Utility screen (Figure 39 on page 88). Identify the package to be unloaded by specifying the owner and package name. Optionally, you may specify a server name other than the one that is displayed. If the package does not exist, you will receive a message in the lower left hand corner of menu.

After you press ENTER while viewing the Package Utility screen, the Package Job Submit screen, Figure 40, is displayed. This is where you specify the parameters for job submission to the VSE/POWER queue.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                        hh:mm:ss
*-------------------       PACKAGE JOB SUBMIT SCREEN       -------------------*
|                                                                             |
|  JOBNAME   => ________                CLASS   => A                          |
|                                                                             |
|  PRI       => 3                       DISP    => D (D,H,L,K)               |
|                                                                             |
|  FROM      => VS02____                                                      |
|                                                                             |
|  DUETIME   => ____ (HHMM)             DUEDATE => ______ (AABBYY)           |
|                                                                             |
|  DUEDAY    => ___________________________________________                   |
|                                                                             |
|  OTHER     => ____________________________________________________________  |
|                                                                             |
|  LST CLASS => A                                                             |
|                                                                             |
|  DEST      => ________                                                      |
|                                                                             |
*-------------------------------------------------------------------- SQC17 ----*

PRESS ENTER TO SUBMIT
ENTER F1=HELP F3=EXIT F12=CANCEL
```

Figure 40. Package Job Submit Menu

Package Utility Job Submit Parameters

The job submission parameters are:

Entry Field	Description
JOBNAME	the name by which the job and its associated queue entries are to be known. Specify a name consisting of 2 to 8 alphameric characters.

CLASS	the class to be assigned to the job in the input queue. Valid values are A through Z, 0 through 9
PRI	priority to be assigned to the job in the input/output queues. Specify a number 0 through 9.
DISP	disposition that specifies how the job is to be handled in the reader queue. Valid values are • D = Delete after processing. • H = Hold. Job remains in reader queue until it is released or altered to another class. • L = Leave in queue. Job is not processed until disposition is altered or job is released. • K = Keep after processing. After job completes, remains in the reader queue with disposition set to L.
FROM	specifies the ID of the user allowed to manipulate or retrieve the job.
DUETIME	specifies the processing time in 24-hour clock time where HH = hour and MM = minute. According to the format defined for your system, AA is month and BB is day, or AA is day and BB is month. Use this operand when job is to be run just once.
DUEDATE/DUEDAY	specifies the days on which a job is to be scheduled. DUEDATE and DUEDAY are mutually exclusive. If a list is specified, it must be enclosed in parentheses. They are specified as: • DAILY - job is scheduled every day • day-list - 'MON,WED,FRI' • number-list - '1,15,28-31'
OTHER	enter any VSE/POWER job parameters not specifically listed elsewhere on the screen.
LST CLASS	defines the class to be assigned to printed output. Specify any letter of the alphabet, or any number 0 through 9.
DEST	specifies the node-id where VSE/POWER is to route the RELOAD PACKAGE job for processing. Specify the name of an applicable node as defined to VSE/POWER with the PNODE macro. This parameter allows you to migrate a package from a local application server to an application server running on a remote system.

How to View a Package

To view a package, specify the OWNER and PACKAGE_NAME fields on the Package Utility screen (Figure 39 on page 88), and select Option **4**. Provide appropriate parameters on the Package Job Submit screen. The View Package tool executes DBSU to unload the desired package to a SAM file. It then generates a package report that lets you view the package contents which include:

- Preprocessing Information
- Each SQL statement used in the associated program
- Information about its corresponding host variables

This data is quite helpful and can be used to analyze performance problems.

The example in Figure 41 shows a report generated from the View Package option.

```
Date: dd Mmm yyyy                                                    Page: 1
                        Control Center Package Report
                        _____________________________

 Package: SQLMSTR.SQC05
 Database: DB2VSE61

 First Create Release: 6.1
 Last Create Release: 6.1
 Charname: INTERNATIONAL
 Sections: 6

 Preprocessing Characteristics:
 ______________________________

     NOGRAPHIC, BLOCK, NOMODIFY, NODESCRIBE

 Options Specified at Prep time:
 _______________________________

     BLock
     ISOLation(RR)

 DEFAULT Options at Prep time:
 _____________________________

     RELease(COMMIT), EXPLAIN(NO), KEEP, REPLACE, NOEXIST
     NOCHECK, PERiod, APOST
     PREPname=SQC05, CTOKEN(NO), LABEL( )

 Static SQL Statements in Package:
 _________________________________

 SELECT NPAGES, POOL INTO :H, :H FROM SYSTEM.SYSDBSPACES WHERE OWNER
 :H AND DBSPACENAME = :H

 SELECT OWNER FROM SYSTEM.SYSDBSPACES WHERE OWNER = :H AND DBSPACETYPE =
 :H AND NPAGES = :H AND POOL IN (:H,:H)

 SELECT OWNER INTO :H FROM SYSTEM.SYSDBSPACES WHERE OWNER = :H AND
 DBSPACENAME = :H

 SELECT VALUE INTO :H FROM SYSTEM.SYSOPTIONS WHERE SQLOPTION = 'RELEASE'


 SELECT CURRENT SERVER INTO :H FROM SYSTEM.SYSOPTIONS WHERE SQLOPTION =
 'RELEASE'
```

Figure 41. Example Output from View Package Option

Chapter 11. Work File Label Definition Tool

About the Work File Label Definition Tool

The Work File Label Definition tool is an easy-to-use interface for DBAs to use to set up the Job Control Language (JCL) label definition statements needed by Control Center for the DBSPACE Reorganization and Analysis utilities. The JCL is stored in the SQMWORK file and is used during job submission to create the JCL for the batch job.

Note that work files are not actually defined until the first time they are used in a Control Center job. If another application uses an area first, the Control Center job will be cancelled.

Work File Label Definition Screen

When Option **4** is chosen from the Control Center main menu, the screen shown in Figure 42 is displayed. This is also the screen that is presented when the CICS transaction SQFM is entered directly.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                      hh:mm:ss
*------------------------ WORK FILE LABEL DEFINITION ------------------------*
|                                                                             |
|  OPTION => 1(DEFINE WORK FILE         )                                     |
|                                                                             |
|  ***************************************************************************|
|                                                                             |
|  FILE TYPE    ==> _ (1=DATA        2=DDL           )                        |
|                     (3=PACKAGES    4=MESSAGES      )                        |
|                                                                             |
|  FILE NUMBER  ==> _                                                         |
|                                                                             |
|                                                                             |
|  TLBL FILE-ID ==> ____________________                                      |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
*------------------------------------------------------------------ SQC03 ----*

PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT
```

Figure 42. Work File Label Definition Screen

This screen allows you to enter the parameters that identify the work file labels you want to define or update. There is one option available - Define Work File.

How the Work File Label Definition Tool Works

There are 2 types of work files that may be defined:

1. Data, and
2. DDL

Data work files are used by the DBSPACE Reorganization and DBSPACE Analysis tools to hold data unloaded from the DBSPACE using the DBSU UNLOAD DBSPACE command. Data work files may be defined as either:

1. Disk, or
2. Tape

DDL work files are used by the DBSPACE Reorganization and DBSPACE Analysis tools to hold DDL generated from the database by SQB01, the Control Center batch DDL generation program. DDL work files are always stored on disk.

To define a tape work file label, leave FILE TYPE and FILE NUMBER BLANK and only enter a 1 to 17 character FILE ID and press ENTER. The TAPE WORK FILE LABEL DEFINITION will be displayed. Remember that tape may only be used for data work files.

Defining a disk work file is a bit more complicated. Since a disk work file can be used for either Data or DDL, you must indicate which by entering the appropriate parameter in the DISK FILE TYPE field (1=DATA/2=DDL).

The DISK FILE NUMBER parameter is used to indicate file size. Control Center limits the amount of work file definition. For disk data, you need to define 3 files depending on size.

1. Small DBSPACES
2. Medium DBSPACES
3. Large DBSPACES

You indicate what file you want to define by the value you enter in the DISK FILE NUMBER parameter. Valid values are 1 through 3.

DDL work files are defined the same way. You enter a DISK FILE NUMBER from 1 to 3 to indicate whether you are defining the small, medium or large DDL work file.

Based on the FILE TYPE and FILE NUMBER entered, Control Center generates a file name and ID. If you enter FILE TYPE 1 and FILE NUMBER 2, Control Center generates FILENAME "SQMDAT2" and FILE ID "SQLMSTR.DATA.FILE2". These are displayed on the DISK WORK FILE DEFINITION screen and may not be changed.

Disk Work File Label Definition Screen

Enter a FILE TYPE and FILE NUMBER and press ENTER to display the DISK WORK FILE LABEL DEFINITION screen as shown in Figure 43 on page 95.

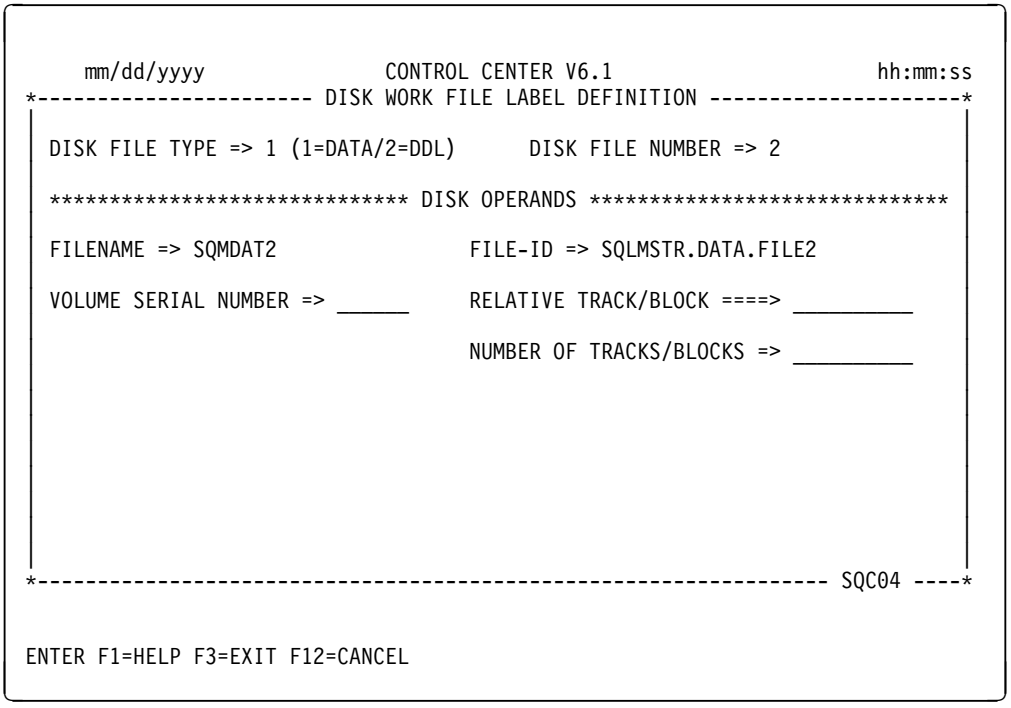

Figure 43. Disk Work File Label Definition Screen

This screen contains the parameters required to define the location and size of a disk work file. The values entered are used to create DLBL, ASSGN, and EXTENT cards that define the file. When you press ENTER, Control Center writes the JCL statements to the SQMWORK file, from which they are available for the job submission routines. Each of the parameters is discussed below.

Disk Work File Label Definition Fields

DISK FILE TYPE and DISK FILE NUMBER are displayed as they were entered on the previous screen. FILENAME and FILE-ID are displayed as generated by Control Center. The following required parameters are explained below. They are used in building the EXTENT JCL statement.

Parameter	**Description**
VOLUME SERIAL NUMBER	This is the serial number parameter on the EXTENT JCL card. It is a 1 to 6 character field indicating the serial number of the volume on which this file is to be located.
RELATIVE TRACK/BLOCK	This is the starting location of the file. For CKD (Count-Key-Data) devices, specify the 1 to 5 digit sequential track number where the file is to begin. For FBA devices, specify the physical block number, from 2 to 2147483645, where the file is to begin.
NUMBER OF TRACKS/BLOCKS	This specifies the size of the work file. For CKD devices, specify the 1 to 5 digit number of tracks to be allocated to the file. For FBA devices, specify the number of physical blocks that should be

allocated to the file. Valid values are 1 to 2147483645.

The actual size of each file depends on the size of the DBSPACES in your database. Refer to Installation Step 3: Allocate DASD for Control Center SAM Work Files on page 22 for sample allocations. Then, make sure the allocations you have made can accommodate your data and DDL.

After entering all parameters, press ENTER to complete processing. Control Center will write DLBL, ASSGN, and EXTENT statements to the SQMWORK file.

F12 will return you to the main WORK FILE LABEL DEFINITION screen. F3 will return you to the Control Center Main Menu.

Note that if the labels for the chosen file are already defined, the currently defined values from the SQMWORK file will be displayed on the DISK WORK FILE DEFINITION screen. When the ENTER key is pressed, the records in the SQMWORK file will be UPDATED with any new values entered.

Tape Work File Definition Screen

When defining a tape file, the screen shown in Figure 44 is presented. This screen contains the parameters used to create TLBL and ASSGN statements for the specified file. When you press ENTER, Control Center writes the JCL to the SQMWORK file from which they are accessed by the job submission routines.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                      hh:mm:ss

*---------------------- TAPE WORK FILE LABEL DEFINITION ---------------------*
|                                                                            |
|  FILENAME => SQMTAPE                TAPE FILE-ID => SQMHELP3               |
|                                                                            |
|  ***************************** TAPE OPERANDS ****************************** |
|                                                                            |
|  VOLUME SERIAL NUMBER => ______     VOLUME SEQUENCE NUMBER => ____         |
|                                                                            |
|  FILE SEQUENCE NUMBER => ____            GENERATION NUMBER => ____         |
|                                                                            |
|        VERSION NUMBER => __                            DATE => ________   |
|                                                                            |
|          DEVICE CLASS => 1                              (YYYY/DDD OR 0-9999) |
|                                                                            |
|           (1=CARTRIDGE/2=TAPE)                         MODE => __         |
|                                                                            |
|                                                                            |
|                                                                            |
*---------------------------------------------------------------- SQC04 ----*

ENTER F1=HELP F3=EXIT F12=CANCEL
```

Figure 44. Tape Work File Label Definition Screen

Each of the parameters is discussed in more detail below.

Tape Work File Label Definition Fields

FILENAME is pre-filled and is always set to "SQMTAPE". TAPE FILE-ID is displayed as it was entered on the WORK FILE LABEL DEFINITION screen.

These parameters are all optional except DEVICE CLASS. They are used in building the TLBL and ASSGN JCL statements that define the file.

Parameter	Description
VOLUME SERIAL NUMBER	This is the file serial number parameter on the TLBL statement. It is a 1 to 6 character field indicating the volume serial number of the first (or only) reel of the file.
VOLUME SEQUENCE NUMBER	This is a 1 to 4-digit number specifying the volume of a multi-volume file at which you wish to start processing.
FILE SEQUENCE NUMBER	This is a 1 to 4-digit number specifying the file of a multi-file volume at which you wish to start processing.
GENERATION NUMBER	This is a 1 to 4-digit number specifying the generation number of the file to be processed.
VERSION NUMBER	This is a 1 or 2-digit number specifying the version of the file to be processed.
DATE	This is the expiration date of the output file expressed either in YYYY/DDD format (absolute expiration date as a Julian date) or as a retention period in days expressed as a 1 to 5-digit number from 0 to 99999.
DEVICE CLASS	This is used in building the ASSGN statement. Valid values are "1" for CARTRIDGE or "2" for TAPE. This is the only required parameter on this screen.
MODE	This specifies density when a device class supports more than 1.

After entering all required parameters, press ENTER to complete processing. Control Center will write a TLBL and an ASSGN card to the SQMWORK file.

F12 returns you to the main WORK FILE LABEL DEFINITION screen. F3 returns you to the Control Center Main Menu.

Note that if the file has already been defined, the current values from the SQMWORK file are displayed on the TAPE WORK FILE LABEL DEFINITION screen. When you press ENTER, the records in the SQMWORK file will be UPDATED with any new values entered.

JCL Default Values

If any of the above optional parameters are not entered, default values in accordance with the *VSE/ESA System Control Statements* manual will be used.

Special Considerations

Size of Defined Files

Depending on the size and type of DBSPACES in your database, definition of the DDL and DATA files need not follow the examples. For instance, a database may have some large DBSPACES (many rows). This would mean a large DATA file would have to be defined. However, the corresponding DDL may in fact be rather small (say a PUBLIC DBSPACE with SELECT granted to all users, only one table, a few indexes, and no referential integrity). The DDL could fit in a small file. In this case, you might define the number 3 DATA file to be much larger and the associated number 3 DDL file to be much smaller.

Chapter 12. CICS Report Controller Interface Tool

About the CICS Report Controller Interface Tool

The CICS Report Controller Interface tool provides a means of transferring from Control Center to the CICS Report Controller so that you can manage the VSE/POWER queue entries associated with the Control Center jobs you have submitted. The CICS Report Controller allows you to release, delete, change, and browse queue entries. When you exit the CICS Report Controller facility, you are returned to the main menu.

Option **5** from the main menu causes Control Center to start the CEOS CICS transaction that displays the initial Report Controller screen. From there, you navigate through the screens and are presented with different options depending on whether you are working with report listings or jobs in the reader queue. To return to Control Center, repeatedly press F3 until the main menu is displayed.

A Sample CICS Report Controller Session

In this sample session, assume that you have submitted a Multiple User mode job to reorganize the PUBLIC.SQMHELP DBSPACE.

You enter Option 5 from the main menu and are presented with the CICS Report Controller main menu as depicted in Figure 45.

```
CEMS:  1      CICS REPORT CONTROLLER

Select one of the following options:

               1  Report selection.
               2  Printer selection.
               3  JCL report (job) selection.
               4  Transient data queue selection.




               Selection ==>









PF1=Help PF3=End ENTER=Continue.
```

Figure 45. CICS Report Controller Main Menu Screen

You enter selection **1** (Report selection) and are presented with the Report Selection screen as shown in Figure 46 on page 100.

```
CEMS: 11                   REPORT SELECTION

      You may list all reports or only those reports that match
      your selection criteria.

      Type your selection criteria:
        Report number  ==>
        Report name    ==>  M*
        Destination    ==>
        Forms          ==>
        Class          ==>               A-Z / 0-9
        From date      ==>               mm/dd/yy
        To date        ==>               mm/dd/yy

      To select Status type a Y against one or more of the following:
        In use         ==>               Held (L)       ==>
        Resume         ==>               Held (H)       ==>
        Error creating ==>               Ready (K)      ==>
        Error printing ==>               Ready (D)      ==>


PF1=Help PF3=End ENTER=Continue.
```

Figure 46. CICS Report Controller Report Selection Screen

You tab down to the Report name field, enter "**M***", and are presented with the Report List screen as depicted in Figure 47.

```
CEMS: 111                  REPORT LIST                    1 TO     1 OF     1

 Enter Options (1=See/Change Characteristics 3=Print 5=Delete 6=Hold 8=Browse)
OPT  NAME     USERDATA         NUMBER STATUS   FORMS CLS PAGES CPY PRI DEST
     MUMREORG                   17471 READY(D)        A     16   1  3  SYSCICSA
```

Figure 47. CICS Report Controller Report List Screen

You enter Option **8** (Browse) and are presented with the Report Browse screen. By pressing F8 (Scr Fwd) and F11 (Page+1), you can view the output of the MUMREORG job, including the portion shown in Figure 48 on page 101.

```
CEMS: 1114        Browsing: MUMREORG                    PAGE 2     OF    16
Search for ==>                                                    More:   - +

09:14:50 Gathering table create DDL...
09:14:51 Gathering deactive primary key DDL...
09:14:51 Gathering inactive unique constraints ..
09:14:51 Gathering table comment DDL...
09:14:51 Gathering column comment DDL...
09:14:51 Gathering primary clustering keys DDL...
09:14:51 Gathering clustering unique constraints DDL...
09:14:51 Gathering index create DDL...
09:14:51 Gathering primary keys DDL...
09:14:51 Gathering foreign keys DDL...
09:14:51 Gathering active unique constraints DDL...
09:14:51 Gathering table grants DDL...
09:14:51 Gathering column grants DDL...
09:14:51 Gathering views and grants DDL...
09:14:51 Gathering package rebind/reload DDL...
09:14:51 Gathering temporary password revokes DDL...
DDL CREATE SUCCESSFUL!

PF1=Help      PF2=Scr Cursor  PF3=Quit   PF4=Scr Left PF5=Scr Right  PF6=Home
PF7=Scr Back  PF8=Scr Fwd     PF9=Repeat PF10=Page-1  PF11=Page+1    PF12=Scal
```

Figure 48. CICS Report Controller Browse Screen

For detailed instruction on using the CICS Report Controller, refer to the IBM *CICS/VSE® Report Controller User's Guide*.

Chapter 13. Control Center Help Facility

About the Help Facility

The Help Facility tool provides information on Control Center menus, functions, and other topics that you may find useful.

You can reach the Help Menu shown in Figure 49 by selecting Option **6** from Control Center's Main Menu.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                        hh:mm:ss
*----------------------------    HELP FACILITY    ----------------------------*
| DATABASE    => SQLDS350                                                     |
| SUBJECT     =>                                                              |
| **************************    HELP SUBJECTS    **************************  |
|  1 APPLICATION GROUP OBJECTS MENU                                           |
|  2 APPLICATION GROUPS LIST                                                  |
|  3 CICS REPORT CONTROLLER                                                   |
|  4 DBSPACE ANALYSIS SUBMIT SCREEN                                           |
|  5 DBSPACE ANALYSIS UTILITY SCREEN                                          |
|  6 DBSPACE REORGANIZATION SCREEN                                            |
|  7 DBSPACE REORGANIZATION SUBMIT SCREEN                                     |
|  8 DISK WORK FILE LABEL DEFINITION SCREEN                                   |
|  9 GROUP AUTHORIZATION - USER GROUP FUNCTION SCREEN                         |
| 10 GROUP AUTHORIZATION ADD USERS SCREEN                                     |
| 11 GROUP AUTHORIZATION APPLICATION GROUP MENU                               |
| 12 GROUP AUTHORIZATION GRANT/REVOKE MENU                                    |
| 13 GROUP AUTHORIZATION LIST                                                 |
| 14 GROUP AUTHORIZATION OBJECT LIST                                          |
| 15 GROUP AUTHORIZATION USER LIST                                            |
*-------------------------------------------------------------- SQC07 ----*

ENTER F1=HELP F3=EXIT F5=BOT F8=FWD
```

Figure 49. Control Center Help Menu

The name of the server you are currently working with is displayed in the DATABASE field in the upper left corner of the screen.

A scrollable list of Help Subjects available is displayed in the body of the screen. By typing the associated subject number in the TOPIC field and pressing ENTER, a display of information on that subject is presented for viewing. Backward and forward as well as top and bottom scrolling are available where applicable.

Text for the Help Facility is located in the database in the PUBLIC.SQMHELP DBSPACE in the SQLMSTR.SQMHELP table. Help records are numbered 0001-9998 for each subject. The 9999 record, for each subject, contains the subject title displayed on the HELP FACILITY Menu Screen.

In general, the text provided is identical to that displayed when F1 (HELP) is entered from many of the Control Center screens.

You can find additional information about Control Center in the DB2 Server for VSE technical library on the World Wide Web at:
http://www.software.ibm.com/data/db2/vse-vm/.

Chapter 14. Control Center Table Utility

The Table Utility provides an easy way for you to view a list of tables stored in a DB2 Server for VSE database, and select one (or to directly specify a single table) and then do these DBA-oriented operations on it:

- Drop (delete) a table.
- Reorganize a table (recreating all associated DDL).
  - Generate DDL.
  - Unload a table to tape or disk in a DBSU internal format.
  - Perform a full reorganization on a table.
  - Reload a table from tape or disk generated by an Unload.
- Create a table.
- Update statistics.

The Table Utility consists of a full screen, interactive interface, written in COBOL and CICS, a batch DDL generator program written in COBOL, and batch jobstreams that execute Database Services Utility (DBSU) commands. VSE/POWER job scheduling support allows a table to be unloaded from one application server and reloaded into another, as well as complete job scheduling of the other functions.

Based on your selections, the Table Utility will:

- generate the DDL required to recreate a table and the objects related to it, including:
  - Table Definition
  - Data
  - Referential Integrity constraints
  - Unique column definitions
  - Indexes
  - Views
  - Grants
  - Table and Column Comments
  - Table and Column Labels
  - Packages
- load data in clustering index sequence
- update statistics
- reprep invalidated packages
- support:
  - Commitcount use
  - table backup (image copy)
  - copying a table to a different database
  - copying a table to a different owner and/or name
  - moving table to a different DBSPACE
  - reloading data into a different table
- drop a table
- provide a screen interface to define (create) a table.

Invocation

There are two ways to invoke the Table Utility:

1. on the Control Center Main Menu - select option 10 (TABLE UTILITY) and press ENTER.
2. directly from CICS, by typing the Table Utility transaction ID "SQTU" and pressing ENTER.

When either of these is done, the Table Utility main menu is displayed.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                        hh:mm:ss
*-------------------------     TABLE UTILITY      -------------------------*
|                                                                           |
| DATABASE     => SQLDS610                                                  |
|                                                                           |
| ************************************************************************* |
|                                                                           |
| OPTION       => _ 1 LIST TABLES                                           |
|                   2 DROP TABLE                                            |
|                   3 REORGANIZE TABLE                                      |
|                   4 CREATE TABLE                                          |
|                   5 UPDATE STATISTICS                                     |
|                                                                           |
| ************************************************************************* |
|                                                                           |
| CREATOR      => ________                                                  |
|                                                                           |
| TABLE        => __________________                                        |
|                                                                           |
*---------------------------------------------------------------- SQC60 ----*

PRESS ENTER TO SELECT FUNCTION
ENTER F1=HELP F3=EXIT
```

Figure 50. Table Utility Main Menu

There are 5 major functions you can select from this menu. They are described in:

Section “LIST TABLES” on page 107
Section “DROP TABLE” on page 108
Section “REORGANIZE TABLE” on page 109
Section “CREATE TABLE” on page 119
Section “UPDATE STATISTICS” on page 124

If Option 1 is selected, several choices exist:

- If TABLE and CREATOR are left blank, then **all** tables in the database will be listed.
- If one of the TABLE or CREATOR fields is blank or specified as "ALL", all tables with the matching other field will be listed.

  Both, or each of the CREATOR and TABLE field values can have a trailing percent sign (%) to represent a string of zero or more characters in the same way one would use the LIKE predicate in an SQL statement.

If Option 2, 3, or 5, is selected, an existing database table name (TABLE) and creator (CREATOR) must be entered (or selected by using Option 1, List TABLES).

If Option 4 is selected, then a CREATOR and non-existing TABLE name must be entered.

Pressing PF12, RETURN, when available, returns control to the invoking screen. Pressing PF3, EXIT, returns control to the point from where the Table Utility was started, for example, the Control Center main menu or CICS.

LIST TABLES

The LIST TABLES function is used to select a table to be operated on by the other functions of the Table Utility.

Using the List Tables function you can search a database for a specific table, several tables with common NAMEs or CREATORs (using “wildcard” specification) or all tables. Tables meeting the NAME or CREATOR search criteria are displayed in the TABLE LIST menu. A specific table can then be selected and its name and creator will be returned to the Table Utility main menu and displayed on it.

The Table List menu is displayed when the Table List criteria you specify on the Table Utility main menu are satisfied. If no tables match the search criteria, a message is displayed indicating that no such tables can be found.

Additional information (described below) is displayed about the tables. Note that these statistics may be correct only as of the last statistics update or table reorganization.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                        hh:mm:ss
*-------------------------      TABLE  LIST         --------------------------*
| DATABASE => SQLDS610                                                        |
|                                                                             |
|   CREATOR  TNAME              AVGROWLEN ROWCOUNT  NPAGES PCTPAGES NOVERFLOW |
|                                                                             |
| _ SQLMSTR  APPL_GROUP_TAB            39        1       1      100          |
| _ SQLMSTR  COUNTER                    0        0       0        0          |
| _ SQLMSTR  GROUP_AUTH_TAB            18        1       1      100          |
| _ SQLMSTR  MONITOR_CONTROL          142        1       1      100          |
| _ SQLMSTR  OBJECT_TAB                36        1       1      100          |
| _ SQLMSTR  RAY_TEST                  13        1       1      100          |
| _ SQLMSTR  RAY_TEST2                 13        1       1      100          |
| _ SQLMSTR  SHOW_ACTIVE                0        0       0        0          |
| _ SQLMSTR  SHOW_CONNECT               0        0       0        0          |
| _ SQLMSTR  SHOW_DBEXTENT              0        0       0        0          |
| _ SQLMSTR  SHOW_DBSPACE               0        0       0        0          |
| _ SQLMSTR  SHOW_LOCK                  0        0       0        0          |
| _ SQLMSTR  SHOW_LOG                   0        0       0        0          |
| _ SQLMSTR  SQLMAINT                  85       16       1      100          |
*-------------------------------------------------------------------- SQC63 ----*

SELECT A TABLE AND PRESS ENTER TO RETURN
F1=HELP F3=EXIT F4=TOP F5=BOT F7=BWD F8=FWD F12=CANCEL
```

Figure 51. Table List Screen

From this menu you can scroll forward or backward, and to the top or bottom, if there are multiple menus. You can choose one table by placing an "X" (or any non-blank character) next to the table name and then pressing ENTER to return to the Table Utility main menu. The chosen table NAME and CREATOR will be placed

in the respective fields of the main menu for use with Table Utility options 2 through 5. For option 4, Create Table, you can use the List function to check that the planned new table's name does *not* exist.

If you press F3, control returns to the Control Center main menu, SQC01. If you press F12, control returns to the Table Utility main menu, SQC60, without changing anything that was originally on that menu.

The Table List menu (Figure 51 on page 107) displays these table attributes from SYSTEM.SYSCATALOG:

Column Label	Content
AVGROWLEN	the average length of the rows in this table
ROWCOUNT	the number of rows in this table as of the last statistics update
NPAGES	the number of pages in the DBSPACE in which rows of this table appear.
PCTPAGES	the approximate percentage of the active pages in the DBSPACE that have rows from this table in them
NOVERFLOW	the number of rows that have overflowed from their original page in storage to another page

List Tables Processing Flow

When you choose the LIST TABLES function, Control Center reads the system catalogs to build a list of tables. Values you specify as CREATOR and TABLE name are passed to SQL to use as filters.

DROP TABLE

This option provides you with an easy method of dropping (deleting) a table.

On the Table Utility main menu, you can directly specify the table to be dropped or you can use the Table List function, option 1, to identify the table. Then, select option 2, Drop Table. You will be prompted with a message asking for confirmation of the DROP TABLE request as shown in Figure 52 on page 109.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                        hh:mm:ss
*-------------------------     TABLE UTILITY      -------------------------*
|                                                                           |
|  DATABASE     => SQLDS610                                                 |
|                                                                           |
|  ************************************************************************ |
|                                                                           |
|  OPTION       => 2 1 LIST TABLES                                          |
|                    2 DROP TABLE                                           |
|                    3 REORGANIZE TABLE                                     |
|                    4 CREATE TABLE                                         |
|                    5 UPDATE STATISTICS                                    |
|                                                                           |
|  ************************************************************************ |
|                                                                           |
|  CREATOR      => SQLMSTR                                                  |
|                                                                           |
|  TABLE        => COUNTER                                                  |
|                                                                           |
*---------------------------------------------------------------- SQC60 ----*

PRESS F10 TO VERIFY DROP OR F12 TO CANCEL
ENTER F1=HELP F3=EXIT
```

Figure 52. Table Utility Main Menu with Prompt for Verifying a Drop Request

You can then cancel (F12) the request or proceed with the drop (F10).

Drop Table Processing Flow

When you choose the drop table option and have entered a table's CREATOR and NAME, Control Center issues a SQL DROP TABLE statement for immediate execution. If the job is successful, message SQM0036, REQUEST SUCCESSFULLY PROCESSED, will be displayed; if not, an SQL error message will be displayed.

REORGANIZE TABLE

The Reorganize Table function is used to simplify your work for backup and restore, table reorganization, and saving table structure information. The function provides four options; see Figure 53 on page 110.

You can choose to:

Option	Description
1 - GENERATE DDL	Generates from the database all of the DDL required to recreate the table a nd the objects it contains, including indexes, views, and grants. The DDL is placed in the punch queue for inspection, alteration, or backup.
2 - UNLOAD TABLE	Generates DDL (as in Option 1) and writes it to a VSAM file. Then, a DBSU UNLOAD TABLE step is executed that writes the table data to a SAM or tape file. If SAM is selected, the file is REPRO'd to a VSAM file for more permanent retention. The unloaded data and generated DDL can be used as the basis for a subsequent

RELOAD TABLE (Option 4) job. The table is not dropped. An example of an UNLOAD TABLE job created to do this is in Figure 69 on page 144.

3 - REORGANIZE TABLE
Results in a full table reorganization. A jobstream is created that generates the DDL, UNLOADS the table, DROPs, recreates, and RELOADS the TABLE. Error recovery logic is also included. An example of a REORGANIZE TABLE job is in Figure 70 on page 145.

4 - RELOAD TABLE
Submits a job to recreate and reload a table that has been unloaded from Option 2. This is basically a table recovery facility. An example of the job created to do this is in Figure 71 on page 148.

Each of the options is discussed in detail below and is accompanied by a sample JCL stream created by the Table Utility.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                        hh:mm:ss
*----------------------     TABLE REORGANIZATION     -------------------------*
| DATABASE  => SQLDS610__________                                             |
| CREATOR   => SQLDBA__                                                       |
| TABLE     => ACTIVITY__________                                             |
| FILE      => 1 (1-3)                                                        |
| OPTION    => 3 (1 GENERATE DDL                2 UNLOAD TABLE)               |
|                (3 REORGANIZE TABLE            4 RELOAD TABLE)              |
| ***********************     OPTIONAL PARAMETERS     ******************      |
| DATABASE  => ____________________                                           |
| OWNER     => ________                                                       |
| DBSPACE   => ____________________                                           |
| CREATOR   => ________                                                       |
| TABLE     =>                                                                |
|                                                                             |
| REBIND PACKAGE => 1 (1=YES/2=NO)         UPDATE ALL STATISTICS => :         |
| COMMITCOUNT    => __________            DATA CAPTURE          => :         |
|                                                                             |
| TLBL FILE-ID   => ___________________                                       |
*-------------------------------------------------------------------- SQC61 ----*

PRESS ENTER TO PROCESS
F1=HELP F3=EXIT
```

Figure 53. Table Reorganization Screen

The following is a summary of the various processing options:

- Generate DDL

  If the Generate DDL option is specified, the utility will generate the table create statement and all associated database objects, including indexes, views, and grants.

- Table Reorganization

  If only the source database and table (at the top of the menu) are specified (meaning that this is not a migration), the utility will perform a table reorganization by generating a job to:

1. generate the DDL for recreating the table and its objects,
2. unload the data to a VSAM file,
3. copy the VSAM file to a SAM file,
4. drop the table,
5. recreate the table,
6. reload the table,
7. and generate the table objects, such as, indexes.

When data and DDL are both generated for a reorganization, the program has DBA authority for copying **all** dependent objects within the database (including those of other users). Control Center will copy the data using UNLOAD TABLE, will DROP and CREATE the table, RELOAD TABLE to reload the data, and then recreate all dependent objects (indexes, views, and grants).

The presence of optional parameters affects the ultimate type of reorganization selected. These are:

– Table Migration

  The utility will migrate data or table DDL between databases if the optional DATABASE parameter is specified. The source table will not be dropped or modified in any way.

– Table Copy

  The utility will copy the DDL and data if the “optional” CREATOR and/or TABLE parameter are specified. The target table will be dropped before being defined. The source table will not be dropped or modified in any way. It is important to note that if the target CREATOR is the same as the source table CREATOR, the INDEX and VIEW create names will be identical to the existing ones in the old table; the create statements will fail. A warning message will be displayed to this effect (SQM0730) and then you can confirm whether to proceed or cancel processing.

– Table Move

  The utility will move a table to a new DBSPACE if the optional OWNER and/or DBSPACE parameters are specified. Note that the old table will be dropped only if the new CREATOR and TABLE name remain the same as the old CREATOR and TABLE.

  If you are moving a database created prior to DB2 Version 5.1:

  - Control Center cannot access the passwords on that database. This means that for any grants for which a password is needed, the password from the old database (at the top of the menu) will be used if it exists. (This may cause the connect in the new data base to fail if the password is different.) If there is no password on the old database, one is generated for you. This would be executed on the new database, thereby changing your password if it existed on the new database.
  - If the TABLE NAME or OWNER change, any view text referencing the old table will **not** change. This means that the view create may fail if the table name is used in the view and it does not exist in the new database.

- Unload Table

  This option creates a backup copy of a table using the UNLOAD TABLE DBSU command. The DDL will also be generated for backup purposes. The table will not be dropped.

- Reload Table

  This option reloads a table from a previous backup made with the UNLOAD TABLE function. The unloaded table resulting from a reorganization function that failed, can also be used as input to this option.

Reorganize Table Processing Flow

When you submit a table reorganization job, Control Center:

1. Links and establishes communication to the target server.
2. Connects as user SQLREORG.
3. Verifies the availability of the new DBSPACE (if specified).
4. Gathers system catalog information about the specified table and creates corresponding DDL statements in the Control Center Database Services Utility (DBSU) command file:
   a. Table create statements
   b. Table comments
   c. Column comments
   d. Table reload statements
   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. Package rebind statements
5. Unloads the table data to the specified disk or tape.
6. Executes the SQLDBSU command file from the Database Services Utility to reorganize the table and rebind any dependent packages.

Special Processing Considerations:

1. In order to retain hierarchical dependencies, Control Center issues all grants in the same chronological order in which they were originally issued.
2. In order to grant authority to an object, the grantor must first connect as the user who originally issued the grant. Therefore, the program must gather database connect passwords for all grantors. If a grantor does not have a connect password, a temporary password is assigned and later removed.
3. The database server does not remove grant information from the system catalogs when a user is removed from the SYSTEM.SYSUSERAUTH table. Consequently, the REORGANIZATION job may need to connect as a nonexistent user in order to re-establish a grant. If this situation occurs, Control Center temporarily grants connect authority to you and later revokes it.
4. Key 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.

Table Reorganization Menu Required Parameters

To reach the Table Reorganization menu shown in Figure 53 on page 110, choose Option **3** on the Table Utility main menu.

When the menu is displayed, the DATABASE, CREATOR and TABLE fields at the top of the menu will be filled in from the Table Utility main menu.

When you installed Control Center, you defined three SAM DDL files to hold generated DDL. Specify the number of the file you want to use in the FILE field. (1 selects the small file, 2 selects the medium file, 3 selects the large file.) The number also indicates what SAM data file to use if you have not entered a Tape File Name. You do not need to specify the file number if you choose Option 1 or 2, because the DDL is written to the punch queue instead of to a file.

Enter the number of the option you want to execute in the Option field.

Table Reorganization Menu Optional Parameters

Parameters below the "OPTIONAL PARAMETERS" line are not required.

Parameter	Description
DATABASE	The name of a different database into which the table is reloaded. Use of this parameter lets you migrate a table from one database to another. For example, you can migrate a table from a development database to a production database. Before you migrate the table, you may want to ensure that the two databases are compatible so that all reload statements execute successfully. When you use the optional DATABASE parameter, the table in the old database remains unchanged.
OWNER/DBSPACE	If used, these two parameters must be used as a pair; however, if you omit one one of the two, Control Center provides a default value. They indicate that you want to specify a new DBSPACE for the table.

	If the new CREATOR/TABLE option is used, a copy will be performed. If a new CREATOR/TABLE is not specified, the table will be moved from the current DBSPACE to the new DBSPACE.
CREATOR/TABLE	If used, these two parameters must be used as a pair; however, if you omit one of the two, Control Center provides a default value. They indicate that you want to specify a new TABLE name (and CREATOR) for the DDL and/or data from the old table. Use of this option pair will cause the source table to be copied, (when using options other than 1 - Generate DDL) with the source table remaining unchanged. **Note:** The table view names and index names will not be changed, causing failure of the DDL CREATE statements if the original table CREATOR is the same as the new CREATOR.
REBIND PACKAGE	Once a table has been reloaded, the Table Utility rebinds all PACKAGES that are dependent on that table. To bypass package rebind processing, specify 2 (NO). The default value is 1 (YES). This option is not valid when the OWNER name is changed.
UPDATE ALL STATISTICS	By default, UPDATE STATISTICS is done for a table during reload and during index create. Statistics (including the number of rows and number of unique values in a column) are, by default, gathered only for columns that appear as the first column in an index. To update the statistics for all the columns in all of your indexes, specify 1 (YES) for this parameter. **Note:** To perform periodic statistics updates, use the Table Utility main menu Option 5, Update Statistics. When this parameter is set to NO, the statistics are reset only for the first column in each index. When set to YES, the statistics are reset for all columns in all indexes. The statistics include such things as the number of rows in a table and the number of unique values in an index column. They are used by SQL to optimize retrievals. Since this is can be a lengthy and resource consuming process, it is recommended that you do not reset all of the statistics during normal work hours. If you are reloading a table, this is an optimal time to reset ALL of the statistics.
COMMITCOUNT	Used to specify the frequency of COMMITS during reload processing. Enter a number in the range 1 through 2147483647 to cause a COMMIT WORK to

be executed after that number of input rows has been reloaded.

TAPE FILE NAME (TLBL FILE-ID)
: Used to specify that data should be unloaded to tape instead of disk. The tape file must have been defined by the WORK FILE LABEL DEFINITION tool. This field does not apply to DDL because DDL is **always** unloaded to disk.

DATA CAPTURE
: Allows changing or adding the attribute to a table for Release 5.1 databases. The default is to use what is currently in the catalog for the table. The value “CHANGES” can be used to change the current value held in the catalog. This affects how much information is retained in the log when a table is changed.

After entering the desired parameters, press ENTER to proceed to the Table Reorganization Submit Screen.

Using the TABLE REORGANIZATION Option

The Table Reorganization option can be used in a variety of ways to achieve different goals. Each of the options is discussed in more detail.

Option 1 - GENERATE DDL

By reading the catalogs, this option generates the DDL necessary to recreate a table and all of its associated objects. DDL is written to the VSE/POWER punch queue in the form of DBSU commands and can be used, as is, to recreate the table. This option:

- relieves DBAs from having to maintain large libraries of DDL.
- saves library disk space.
- solves the problem of who owns the "official" DDL.
- provides an easy way to determine table and index characteristics.
- provides authorization and "where-used" information.

Figure 54 is an example of the jobstream produced by Control Center to generate DDL for the SQLDBA.ACTIVITY table.

```
* $$ JOB JNM=TABLEDDL,CLASS=A,PRI=3,DISP=D
* $$ LST PRI=3
* $$ PUN PRI=3
// JOB TABLEDDL GENERATE DDL
// OPTION LOG
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0001 GENERATE DDL FOR "SQLDBA"."ACTIVITY"
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMTPRM,'SQLMSTR.TABLE.PARMS',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS006,SYSPCH
// ASSGN SYS011,SYSLST
// EXEC SQB60,SIZE=AUTO
%%SQLDS610          SQLDBA  ACTIVITY          1
/*
/&
* $$ EOJ
```

Figure 54. Table Reorg Option 1 (Generate DDL) - Sample Jobstream

Option 2 - UNLOAD TABLE

This option generates the DDL necessary to recreate a table and writes it to a VSAM file. It then unloads the table to a SAM disk file (or a tape if a tape label was specified). The SAM data file is then REPRO'd to a VSAM-managed SAM file for more permanent retention. Data is unloaded in system-defined format; you must make sure that this data file is not altered prior to reloading the table. This option is essentially a table backup. Used in conjunction with a RELOAD TABLE (Option 4), it provides the capability to recover from application errors.

Figure 69 on page 144 shows a jobstream generated by Control Center to unload the SQLDBA.ACTIVITY TABLE.

Option 3 - REORGANIZE TABLE

This is the most comprehensive option of the reorganization tool. It schedules a full table reorganization, including generating all related DDL and executing it. Depending on the optional parameters chosen, a table can be migrated to another DBSPACE or another owner, the table name can be changed, and the table can be moved to another database.

Figure 70 on page 145 shows a jobstream generated by Control Center to reorganize the SQLDBA.ACTIVITY table.

Option 4 - RELOAD TABLE

This option generates a job to reload a table previously unloaded or reorganized using Control Center. The previously created DDL and data files are used to recreate the table in its entirety. This option is the recovery counterpart to the UNLOAD TABLE option, (Option 2), and is the method of recovering from an error during a reorganization reload step.

Figure 71 on page 148 shows a jobstream generated by Control Center to reload the SQLDBA.ACTIVITY table.

Table Reorganization Submit Screen

Figure 55 on page 117 shows the Table Reorganization Submit Screen.

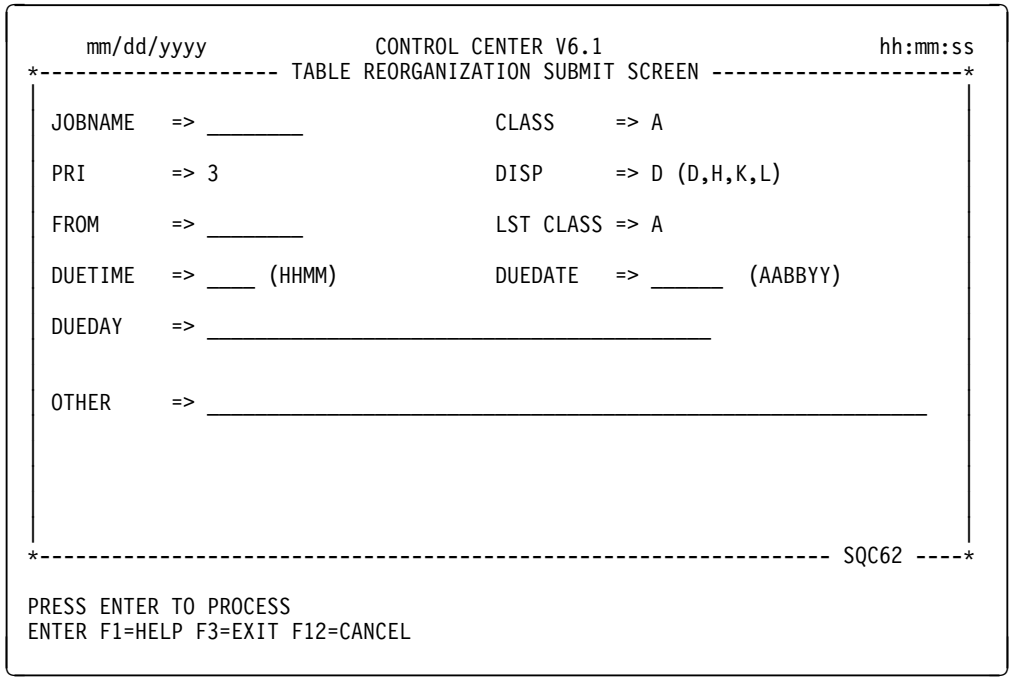

Figure 55. Table Reorganization Submit Screen

To reach this menu, press ENTER from the Control Center Table Reorganization menu.

Job Submission Screen Required Parameters

On each of the job submission screens (Table Reorganization, Create Table, and Update Statistics), there are parameters that are required, that are “required” but have defaults, and some that are optional. The following parameter descriptions apply to **ALL** of the Table Utility job submission screens.

Parameter	Description
JOBNAME	Specifies the job name for the Table Reorganization job and its associated queue entries. The JOBNAME parameter is the only parameter that you must enter because there are default values for the other parameters that are required.
CLASS	Specifies the class or partition in which you want this job to run. This parameter is required; its default value is **A**.
PRI	Specifies the priority to be assigned to the job. Specify a number from 0 to 9 where 9 is the highest priority. This parameter is required; its default value is **3**.
DISP	Specifies how the job is to be handled in the reader queue. DISPosition may be specified as: • **D** - Delete after processing • **H** - Hold until released • **K** - Keep after processing • **L** - Leave in the queue This parameter is required; its default value is **D**.

Note: If you enter data for any of the Table Utility functions that cause job submission and the job fails, you can easily resubmit the job. However, this is not completely true when creating a table because you may have entered a lot of data to define the table. There is no way to "reload" that data entry menu, but, there **is** a way to recover.

If you use **DISP=K**, then the job is kept in the VSE/POWER queue, after the job ends, whether it is successful or not. If it is not successful, you can edit the job, fix the problem, and resubmit the job. If you use **DISP=K**, be sure to delete the job from the queue when you are done.

FROM — Specifies the ID of the user being allowed to manipulate or retrieve the job. This parameter is required. It defaults to the CICS User ID, even though that is **not displayed** on the screen.

LST CLASS — Defines the class to be assigned to printed output. The value of this parameter is used on a $$ LST card. Specify any letter of the alphabet, or any number 0 through 9. This parameter is required; its default value is **A**.

Job Submission Screen Optional Parameters

Parameter	Description
DUETIME	Specifies the job processing start time using **HH** for hour and **MM** for minute in 24-hour clock time.
DUEDATE	Specifies the date on which the job is to be run, using **YY** for the year. Depending on the format defined for your system, **AA** is the month and **BB** is the day, or **AA** is the day and **BB** is the month.
DUEDAY	Specifies the day(s) the job is to be scheduled. You can enter a day name or abbreviation such as **MON** for Monday, or a list of these separated by commas and enclosed in parentheses. You can enter the day of the month or a list of day numbers separated by commas and enclosed in parentheses, and can also specify DAILY to schedule the job every day of the year. Certain combinations of the date and time parameters specify repeated processing. The *VSE/POWER Installation and Operations Guide* manual explains the logic used to control repeated processing.
OTHER	The VSE/POWER * $$ JOB card offers many parameters that do not appear on the Table Utility job submission screens. Use this field to have Control Center include those parameters when the job is submitted.

After entering the desired submit parameters, press ENTER to submit the job to VSE/POWER. For more information on VSE/POWER jobs, see the *VSE/POWER Installation and Operations Guide*. After submitting any Table Utility job, control always returns to the screen from which job submission was selected.

CREATE TABLE

The Create Table function provides you with an easy, interactive, interface for defining SQL tables and having them created by the Table Utility. You are presented with a Create Table screen (see Figure 56).

```
     mm/dd/yyyy              CONTROL CENTER V6.1                        hh:mm:ss
*--------------------------     CREATE  TABLE      --------------------------*
| DATABASE    => SQLDS610
| CREATOR     => SQLDBA                   OWNER       => ________
| TABLE       => ACTIVITY2                DBSPACE     => ____________________
|
|   COL CNAME              COLTYPE         LENGTH NULLS (1/2)
|   001 ____________________ _______________ _____   _
|   002 ____________________ _______________ _____   _
|   003 ____________________ _______________ _____   _
|   004 ____________________ _______________ _____   _
|   005 ____________________ _______________ _____   _
|   006 ____________________ _______________ _____   _
|   007 ____________________ _______________ _____   _
|   008 ____________________ _______________ _____   _
|   009 ____________________ _______________ _____   _
|   010 ____________________ _______________ _____   _
|  ***********************     OPTIONAL PARAMETERS     ******************
|  GRANT SELECT TO PUBLIC => 2 (1=YES/2=NO)            DATA CAPTURE => NONE___
|  COMMENT ON TABLE => ____________________________________________________
*------------------------------------------------------------------ SQC65 ----*

ENTER DATA AND A VALID FKEY TO CONTINUE PROCESSING
ENTER F1=HELP F3=EXIT F8=FWD F12=CANCEL
```

Figure 56. Create Table Data Entry Screen

Create Table Processing Flow

Screen rows are scanned from the top to the bottom (including the DATABASE, CREATOR, OWNER, TABLE, and DBSPACE). If any errors are detected, an error message is issued; when processing resumes, the top-to-bottom scan restarts at the top of the screen.

A data column must have a name, column type, and if required by the column type, a length (default lengths are described just below). An embedded blank row (column definition) is not allowed. When you press ENTER, the SQL statements corresponding to the table structure are generated and made ready for job submission.

Using the Create Table Function

Entering a Column's Data

Note: In the remainder of the Table Create discussion, the term **row** refers to a row on the screen; this is synonymous with **table column**.

Each data column to be defined for the table is specified on a row on the CREATE TABLE screen. Each row on the screen must be completed (at a minimum, have a column name, type, and length if required) before you can scroll down to the next screen. No intermediate, empty, rows are allowed. Each CREATE TABLE entry

screen has space to enter data for 10 data columns; when 10 entries are made, you can scroll down to make the next 10 entries. Before you can scroll down, the entries you have made must be correct for the fields you have entered. When you scroll away from a screen on which you have entered data, Control Center, if you have not entered a length, will fill in the length field with the SQL default length length for the specific data type, such as, character fields default to length 1.

Note: SQL has a limitation on the length of a row in a database table. The CREATE TABLE statement syntax (in the *DB2 Server for VSE & VM SQL Reference* manual) describes the limitation. The limit is approximately 4K bytes, however, variable length, and large character strings, each require only a few bytes of the 4K, so this limit is usually not significant.

There are many infrequently used SQL options that you may want to specify when defining a table column, such as UNIQUE. Control Center cannot directly support these due to their number and complexity; however, there **is** a way for you to specify them. In the description of the job submission DISP parameter on page 117, note that you can specify that after a job is submitted, it should be held rather than run. When you create a table and you need to include specifications that Control Center does not directly support, use DISP=HOLD. Then, you can copy the job from the VSE/POWER queue to the ICCF library, edit it, add the additional specifications, and then resubmit the job.

Control Center does not directly support specifying a primary key, although you **can** do this by using DISP=HOLD as described immediately above. The designers of Control Center assume that if you are creating a table, you will usually follow that with a bulk load. Existence of a primary key when a table is initially loaded can cause serious performance degradation. It is much more efficient to do the load first and then specify the primary key.

When You Think You're Done

DO NOT PRESS ENTER UNTIL YOU HAVE ENTERED **ALL** OF THE COLUMN DATA YOU PLAN TO ENTER FOR ALL OF THE ROWS.

Pressing the ENTER key indicates that you have completed entering the table's definition. If no errors are found, the CREATE TABLE job submission screen (see Figure 58 on page 123) is displayed. *If you accidentally press ENTER, when the CREATE TABLE job submission screen is displayed, press PF12 to cancel the job submission and return to the CREATE TABLE screen.*

Inserting and Deleting Data Columns

If you have not yet submitted the table create job, you can change, insert, and delete rows. Suppose that you have specified columns 1 through 10, used PF8 to scroll to a second screen, and have specified additional columns. At this time, you decide to make changes to rows specified on the first screen. Press PF7 to return to the previous screen. If there are no errors in the specification of the rows on the current screen, the previous screen will be displayed. If there are errors, they must be fixed before you can scroll.

You will now notice that a new field, with no heading, has been added to the screen. Located to the left of the column number (COLNO) field, this field is used for indicating that you want to insert or delete a column. The field has no meaningful use when an empty CREATE TABLE screen is first presented, so it does not appear the first time a screen is displayed for column data entry. You

must scroll away from a screen and then back to it for the new field to be displayed.

You can use this field to insert or delete a row. To delete a row, type a **2** in the new field at the left end of the row. To insert a new row **ahead** of another, type a **1** in the new field and then overtype the row's data with the new information; the original row's data will not be changed. In addition to inserts and deletes, you can also overtype existing fields that you want to change, and, finish any partially complete entries; this type of change is automatically recognized by Control Center.

Note: Note that when you do an insert, under some conditions, a row of data that is pushed down onto the next set of ten entries may be marked internally as not being processed; if this happens, a message is displayed identifying the unprocessed row (data column). To clear the message and proceed, scroll down until the unprocessed row is visible on the screen. Then continue what you were doing. If the message was displayed because you pressed ENTER, press it again. If there is another unprocessed row, the message will be displayed again. Scroll down until it is displayed and continue what you were doing. Scrolling off a screen with such a row will cause it to be processed.

When you have completed all of the changes on a screen, press a scroll key, or if you are finished with the table definition, press ENTER to actuate the inserts, deletes, and changes. Control Center will process the new information on the screen, inserting and deleting rows, and changing any other data you have entered. Error checking is performed as usual and the screen is re-displayed with deleted rows removed, inserted rows added, and any other changes made. Inserts and deletes cause automatic renumbering of the rows. Make these types of changes carefully; there is **no** way to undo a change other than retyping the original data, or using the DISP=HOLD technique.

OK, You Really Are Done

When you have completed all of the column definitions, press ENTER to display the CREATE TABLE job submission screen. When the job is submitted, Control Center creates the JCL and DDL statements to define the table and submits the job to the VSE/POWER queue for execution.

Create Table Parameters

The DATABASE, CREATOR and TABLE fields are required and are automatically filled in from the Table Utility main menu. The COLNO parameter is initialized to column numbers 1 - 10 on the first screen displayed. When the screen is scrolled forward, columns 11 - 20 will be displayed (and so on with any further screen entries). A maximum of 255 column entries is allowed.

The OWNER and DBSPACE parameters are required because they are needed in the CREATE TABLE DDL statement generated by Control Center that processes the screen input.

Parameters below the "OPTIONAL PARAMETERS" line are **not** required.

These fields are used to define the table to be created:

Parameter	Description
CNAME	This is the column name.

COLTYPE The datatype of the column. Valid types are those defined in the CREATE TABLE statement in the *DB2 Server for VSE & VM SQL Reference* manual. Abbreviations allowed by SQL are also allowed here. Note that the *DB2 Server for VSE & VM SQL Reference* manual does not show that **DEC** is a valid abbreviation for DECIMAL, although Control Center accepts it.

LENGTH A LENGTH is required for VARCHAR and VARGRAPHIC data types only. All other data types have a default length (or precision) specified in the *DB2 Server for VSE & VM SQL Reference* manual. Unless you specify a LENGTH value, the SQL defaults will be used.

You can add additional column specification parameters if necessary. To do this, when you submit the Create Table job, use DISP=H as the disposition parameter on the Create Table Submit Screen. Then, you can edit the job in the VSE/POWER queue, add the additional parameters, and release the job for execution.

NULL Specify "1" if the column allows NULLS, "2" if the column may not be NULL. This field defaults to "1".

These are the "OPTIONAL PARAMETERS":

Parameter	Description
GRANT SELECT TO PUBLIC	Specifying a 1 (YES) for this parameter will cause the statement to be generated in the DDL following the CREATE TABLE.
DATA CAPTURE	This field causes the table to be defined with DATA CAPTURE NONE (the default), or DATA CAPTURE CHANGES if CHANGES is specified. It affects how much information is retained in the log when a table is changed.
COMMENT ON TABLE	This field allows the you to add a comment to the generated table. The COMMENT ON TABLE statement will be generated in the DDL. The comment may not contain apostrophes or quotation marks.

Figure 57 on page 123 is an example of a filled in Create Table screen.

After entering the desired parameters and column descriptions, press F8 to scroll forward for entry of more column definitions or press ENTER to proceed to the Create Table Submit screen.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                        hh:mm:ss
*-------------------------     CREATE  TABLE       -------------------------*
| DATABASE    => SQLDS610
| CREATOR     => SQLDBA                    OWNER       => owner
| TABLE       => ACTIVITY2                 DBSPACE     => dbspacename
|
|   COL CNAME              COLTYPE         LENGTH NULLS (1/2)           (1/2)
|   001 name-last          char             20
|   002 name-first         char             25
|   003 name-middle        char             25     1
|   004 title              char             10
|   005 address 1          char             25
|   006 address 2          char             25     1
|   007 address 3          char             25     1
|   008 city               char             25
|   009 state              char             2
|   010 zip-5              char             5
| **********************     OPTIONAL PARAMETERS     ******************
| GRANT SELECT TO PUBLIC => 2 (1=YES/2=NO)            DATA CAPTURE => NONE
| COMMENT ON TABLE =>
*------------------------------------------------------------------- SQC65 ----*

ENTER DATA AND A VALID FKEY TO CONTINUE PROCESSING
ENTER F1=HELP F3=EXIT F4=TOP F5=BOT F7=BWD F8=FWD F12=CANCEL
```

Figure 57. Create Table Data Entry Screen

When you press ENTER, the Create Table Submit screen is displayed; see Figure 58.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                        hh:mm:ss
*----------------------- CREATE TABLE SUBMIT SCREEN -------------------------*
|
| JOBNAME   => ________                CLASS     => A
|
| PRI       => 3                       DISP      => D (D,H,K,L)
|
| FROM      => ________                LST CLASS => A
|
| DUETIME   => ____ (HHMM)             DUEDATE   => ______
|
| DUEDAY    => _____________________________________________
|
|
| OTHER     => ____________________________________________________________
|
|
|
|
*------------------------------------------------------------------- SQC66 ----*

PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT F12=CANCEL
```

Figure 58. Create Table Submit screen

The first parameter, JOBNAME, is the only one that is required.

For a detailed description of the job submission parameters, see the description of the job submission parameters on page 117.

Note: In the discussion of the job submission DISP parameter on page 117, be sure to note that using DISP as a means of recovering from some of the errors you might make when creating a table.

The description of DUEDAY on page 118 explains that you can specify repetitive processing for a job. When you are creating a table, this option may not be appropriate, so Control Center issues a warning message as shown at the bottom of Figure 59. You can change the repetitive specification or keep it.

```
     mm/dd/yyyy              CONTROL CENTER V6.1                      hh:mm:ss
*------------------------ CREATE TABLE SUBMIT SCREEN -------------------------*
|                                                                             |
|  JOBNAME   => MAKETABL               CLASS     => A                         |
|                                                                             |
|  PRI       => 3                      DISP      => D (D,H,K,L)               |
|                                                                             |
|  FROM      => ________               LST CLASS => A                         |
|                                                                             |
|  DUETIME   => ____ (HHMM)            DUEDATE   => ______                    |
|                                                                             |
|  DUEDAY    => DAILY_____________________________________                    |
|                                                                             |
|                                                                             |
|  OTHER     => ____________________________________________________________  |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
*------------------------------------------------------------------- SQC66 ----*
WARNING: YOU HAVE REQUESTED REPETITIVE SCHEDULING FOR CREATING A TABLE.
PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT F12=CANCEL
```

Figure 59. Create Table Submit Screen With Time Scheduling Warning

UPDATE STATISTICS

This function provides you with the capability of issuing the UPDATE STATISTICS or UPDATE ALL STATISTICS commands in a background SQLDBSU job.

When you select Option 5, Update Statistics, on the Control Center main menu, the Update Statistics Submit screen is displayed (Figure 60 on page 125).

```
     mm/dd/yyyy              CONTROL CENTER V6.1                      hh:mm:ss
*---------------------- UPDATE STATISTICS SUBMIT SCREEN ----------------------*
|                                                                             |
| ********************      VSE/POWER JOB PARAMETERS      ******************* |
|                                                                             |
|  JOBNAME   => ________  CLASS   => 'A'  PRI => '3'  DISP ====> 'D'  (D,H,L,K)|
|                                                                             |
|  FROM      => VS52____  DUETIME => ____ (HHMM)    DUEDATE => ______         |
|                                                                             |
|  DUEDAY    => ____________________________________________                  |
|                                                                             |
|  LST CLASS => A                                                             |
|                                                                             |
|  OTHER     => ____________________________________________________________  |
|                                                                             |
|  **************************      JOB OPTIONS      ************************** |
|                                                                             |
|  UPDATE ALL STATISTICS => 2 (1=YES/2=NO)                                    |
|                                                                             |
|                                                                             |
*------------------------------------------------------------------- SQC64 ----*

PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT F12=CANCEL
```

Figure 60. Update Statistics Submit Screen

Update Statistics Submit Screen Required Parameters

The first parameter, JOBNAME, is the only one that is required. For a detailed description of the job submission parameters, see job submission parameters on page 117.

Update Statistics Submit Screen Optional Parameters

Parameter	Description
UPDATE ALL STATISTICS	Specifies whether ALL should be added to the UPDATE STATISTICS command. 1 indicates YES, 2 indicates NO. When this parameter is set to NO, the statistics are reset for the first column in each index. When set to YES, the statistics are reset for all columns in all indexes. The statistics include such things as the number of rows in a table and the number of unique values in an index column. They are used by SQL to optimize retrievals. Since this is can be a lengthy and resource consuming process, it is recommended that you do not reset all of the statistics during normal work hours. **Note:** You should update all statistics on a periodic basis for tables that are subject to frequent change.

After entering the desired submit parameters, press ENTER to submit the job to VSE/POWER. For more information on VSE/POWER jobs, see the *VSE/POWER Installation and Operations Guide*.

Appendix A. Installation JCL

The Control Center distribution library contains all of the JCL necessary to install the product. You will need to punch these members and import them into your editor so that you can customize them as necessary before submitting them. Listed below are the members that contain JCL and a brief description of their function. In parenthesis, is the page on which you can find the referenced figure.

1. SQLMAINT — Maintenance Report Template (74)
2. SQMCRGRP.Z — DBSU Job Defining Group Authorization Tables (169)
3. SQMCRHLP.Z — Create and load the Control Center help table (135)
4. SQMCRMNT.Z — Create the Control Center maintenance table (163)
5. SQMCRMON.Z — DBSU Job Defining Monitor Tables (164)
6. SQMCSDUP.Z — Update the CICS System Definition file in offline mode (130)
7. SQMDELET.Z — Remove Unnecessary Languages (14 , (25)
8. SQMFCT.A — Control Center FCT Macros (133)
9. SQMGRANT.Z — Grant DBA authority to SQLMSTR (16)
10. SQMLDMSG.Z — Load the Control Center error messages (15)
11. SQMLIBDF.Z — Control Center Package Library Definition (134)
12. SQMMSHPI.Z — Load Control Center into your production library (10)
13. SQMRENAM.Z — Select a Language (14)
14. SQMRLDPK.Z — Load Control Center Packages (158)
15. SQMSTD.Z — Define the Control Center standard labels (12)
16. SQMTSCAN.Z — Scan the Control Center distribution tape (8)
17. SQMVSAM.Z — Define the Control Center VSAM environment (128)

```
$ $$ JOB JNM=SQMVSAM,CLASS=0,DISP=D,PRI=9
$ $$ LST CLASS=Q
// JOB SQMVSAM SETUP CONTROL CENTER VSAM ENVIRONMENT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* GLOBALLY REPLACE '$ $$' WITH '* $$'     EX: CH/$ $$/* $$/ * G
* GLOBALLY REPLACE '#'    WITH '/'        EX: CH|#|/| * G
* GLOBALLY REPLACE XXXXXX WITH YOUR VOLID EX: CH/XXXXXX/DATA12/ * G
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// EXEC IDCAMS,SIZE=AUTO

     DEFINE UCAT                      /* DEFINE USER CATALOG */     -
          (NAME(SQLMSTR.USER.CATALOG)                                -
           CYL(1)                                                    -
           ORIGIN(YYYY)          /* <=== CATALOG ORIGIN HERE */     -
           VOL(XXXXXX))

     DEFINE SPACE                           /* DEFINE SPACE */     -
          (CYL(200)                                                  -
           ORIGIN(ZZZZ)            /* <=== SPACE ORIGIN HERE */     -
           VOL(XXXXXX))                                              -
     CAT(SQLMSTR.USER.CATALOG)

     DEFINE CLUSTER                  /* DEFINE DEFAULT MODEL */     -
          (NAME(DEFAULT.MODEL.ESDS.SAM)                              -
           NOALLOCATION                                              -
           NONINDEXED                                                -
           RECORDFORMAT(UNDEF)                                       -
           RECORDS(1000 100)                                         -
           RECORDSIZE(2000 2000)                                     -
           REUSE                                                     -
           SPEED                                                     -
           VOLUMES(XXXXXX))                                          -
     CATALOG(SQLMSTR.USER.CATALOG)

     DEFINE CLUSTER                       /* DEFINE SQMTPRM */     -
          (NAME(SQLMSTR.TABLE.PARMS)                                 -
           CYL(2 2)                                                  -
           FSPC(15 7)                                                -
           IXD                                                       -
           RECSZ(160 160)                                            -
           REUSE                                                     -
           SHR(4)                                                    -
           VOL(XXXXXX))                                              -
     DATA                                                            -
          (NAME(SQLMSTR.TABLE.PARMS.DATA)                            -
           CISZ(4096)                                                -
           KEYS(45 0)                                                -
     INDEX                                                           -
          (NAME(SQLMSTR.TABLE.PARMS.INDEX))                          -
     CATALOG(SQLMSTR.USER.CATALOG)

     DEFINE CLUSTER                       /* DEFINE SQMMESG */     -
          (NAME(SQLMSTR.MESSAGES)                                    -
           CYL(2 2)                                                  -
           FSPC(0 0)                                                 -
           IXD                                                       -
           RECSZ(80 80)                                              -
           REUSE                                                     -
           SHR(2)                                                    -
           VOL(XXXXXX))                                              -
     DATA                                                            -
          (NAME(SQLMSTR.MESSAGES.DATA)                               -
           KEYS(4 0))                                                -
```

Figure 61 (Part 1 of 2). Control Center VSAM Definitions (SQMVSAM.Z)

```
     INDEX                                                          -
          (NAME(SQLMSTR.MESSAGES.INDEX))                            -
     CATALOG(SQLMSTR.USER.CATALOG)

     DEFINE CLUSTER                        /* DEFINE SQMPARM */     -
          (NAME(SQLMSTR.REORG.PARMS)                                -
           CYL(2 2)                                                 -
           FSPC(15 7)                                               -
           IXD                                                      -
           RECSZ(165 165)                                           -
           REUSE                                                    -
           SHR(4)                                                   -
           VOL(XXXXXX))                                             -
     DATA                                                           -
          (NAME(SQLMSTR.REORG.PARMS.DATA)                           -
           CISZ(4096)                                               -
           KEYS(45 0))                                              -
     INDEX                                                          -
          (NAME(SQLMSTR.REORG.PARMS.INDEX))                         -
     CATALOG(SQLMSTR.USER.CATALOG)

     DEFINE CLUSTER                        /* DEFINE SQMRDAT */     -
          (NAME(SQLMSTR.REORG.DATA)                                 -
           CYL(2 2)                                                 -
           FSPC(15 7)                                               -
           IXD                                                      -
           RECSZ(87 87)                                             -
           REUSE                                                    -
           SHR(4)                                                   -
           VOL(XXXXXX))                                             -
     DATA                                                           -
          (NAME(SQLMSTR.REORG.DATA.DATA)                            -
           CISZ(4096)                                               -
           KEYS(44 0))                                              -
     INDEX                                                          -
          (NAME(SQLMSTR.REORG.DATA.INDEX))                          -
     CATALOG(SQLMSTR.USER.CATALOG)

     DEFINE CLUSTER                        /* DEFINE SQMWORK */     -
          (NAME(SQLMSTR.WORK.FILES)                                 -
           CYL(2 2)                                                 -
           FSPC(15 7)                                               -
           IXD                                                      -
           RECSZ(90 90)                                             -
           REUSE                                                    -
           SHR(2)                                                   -
           VOL(XXXXXX))                                             -
     DATA                                                           -
          (NAME(SQLMSTR.WORK.FILES.DATA)                            -
           CISZ(4096)                                               -
           KEYS(18 0))                                              -
     INDEX                                                          -
          (NAME(SQLMSTR.WORK.FILES.INDEX))                          -
     CATALOG(SQLMSTR.USER.CATALOG)
#*
#&
$ $$ EOJ
```

Figure 61 (Part 2 of 2). Control Center VSAM Definitions (SQMVSAM.Z)

```
$ $$ JOB JNM=SQMCSDUP,DISP=D,CLASS=0
$ $$ LST CLASS=Q
// JOB SQMCSDUP DEFINE CONTROL CENTER PROGRAMS/TRANSACTIONS TO CICS
// DLBL DFHCSD,'CICS.CSD',,VSAM,CAT=VSESPUC
// LIBDEF *,SEARCH=PRD1.BASE
// EXEC DFHCSDUP,SIZE=AUTO
ADD GROUP(SQM) LIST(VSELIST)
DEFINE PROGRAM(SQC01)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC02)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC03)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC04)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC05)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC06)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC07)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC08)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC09)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC10)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC11)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC12)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC16)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC17)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC19)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC20)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC21)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC22)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC23)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC24)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC25)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC26)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC27)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC28)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC29)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC40)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC41)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC42)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC43)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC44)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC45)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC46)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC47)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC48)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC49)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC50)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC51)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC52)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC53)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC54)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC60)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC61)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC62)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC63)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC64)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC65)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQC66)    LANGUAGE(COBOL)     GROUP(SQM)
DEFINE PROGRAM(SQM01)    LANGUAGE(ASSEMBLER) GROUP(SQM)
```

Figure 62 (Part 1 of 3). Define Control Center Programs and Transactions to CICS (SQMCSDUP.Z)

```
DEFINE PROGRAM(SQM02)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM03)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM04)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM05)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM06)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM07)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM08)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM09)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM10)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM11)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM12)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM14)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM16)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM17)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM19)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM20)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM21)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM22)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM23)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM24)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM25)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM26)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM27)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM28)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM29)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM40)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM41)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM42)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM43)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM44)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM45)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM60)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM61)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM62)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM63)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM64)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM65)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM66)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE PROGRAM(SQM99)    LANGUAGE(ASSEMBLER) GROUP(SQM)
DEFINE TRANSACTION(SQM)  PROGRAM(SQC01)      GROUP(SQM)
DEFINE TRANSACTION(SQHD) PROGRAM(SQC02)      GROUP(SQM)
DEFINE TRANSACTION(SQFM) PROGRAM(SQC03)      GROUP(SQM)
DEFINE TRANSACTION(SQFD) PROGRAM(SQC04)      GROUP(SQM)
DEFINE TRANSACTION(SQDR) PROGRAM(SQC05)      GROUP(SQM)
DEFINE TRANSACTION(SQDS) PROGRAM(SQC06)      GROUP(SQM)
DEFINE TRANSACTION(SQHM) PROGRAM(SQC07)      GROUP(SQM)
DEFINE TRANSACTION(SQMM) PROGRAM(SQC08)      GROUP(SQM)
DEFINE TRANSACTION(SQML) PROGRAM(SQC09)      GROUP(SQM)
DEFINE TRANSACTION(SQMS) PROGRAM(SQC10)      GROUP(SQM)
DEFINE TRANSACTION(SQOM) PROGRAM(SQC11)      GROUP(SQM)
DEFINE TRANSACTION(SQOD) PROGRAM(SQC12)      GROUP(SQM)
DEFINE TRANSACTION(SQPM) PROGRAM(SQC16)      GROUP(SQM)
DEFINE TRANSACTION(SQPS) PROGRAM(SQC17)      GROUP(SQM)
DEFINE TRANSACTION(SQGA) PROGRAM(SQC19)      GROUP(SQM)
DEFINE TRANSACTION(SQUG) PROGRAM(SQC20)      GROUP(SQM)
```

Figure 62 (Part 2 of 3). Define Control Center Programs and Transactions to CICS (SQMCSDUP.Z)

```
DEFINE TRANSACTION(SQL1) PROGRAM(SQC21)      GROUP(SQM)
DEFINE TRANSACTION(SQL2) PROGRAM(SQC22)      GROUP(SQM)
DEFINE TRANSACTION(SQL3) PROGRAM(SQC23)      GROUP(SQM)
DEFINE TRANSACTION(SQL4) PROGRAM(SQC24)      GROUP(SQM)
DEFINE TRANSACTION(SQUF) PROGRAM(SQC25)      GROUP(SQM)
DEFINE TRANSACTION(SQAG) PROGRAM(SQC26)      GROUP(SQM)
DEFINE TRANSACTION(SQAU) PROGRAM(SQC27)      GROUP(SQM)
DEFINE TRANSACTION(SQAO) PROGRAM(SQC28)      GROUP(SQM)
DEFINE TRANSACTION(SQAD) PROGRAM(SQC29)      GROUP(SQM)
DEFINE TRANSACTION(SQRU) PROGRAM(SQC40)      GROUP(SQM)
DEFINE TRANSACTION(SQRM) PROGRAM(SQC41)      GROUP(SQM)
DEFINE TRANSACTION(SQRP) PROGRAM(SQC42)      GROUP(SQM)
DEFINE TRANSACTION(SQRL) PROGRAM(SQC43)      GROUP(SQM)
DEFINE TRANSACTION(SQRV) PROGRAM(SQC44)      GROUP(SQM)
DEFINE TRANSACTION(SQRR) PROGRAM(SQC45)      GROUP(SQM)
DEFINE TRANSACTION(SQRK) PROGRAM(SQC46)      GROUP(SQM)
DEFINE TRANSACTION(SQR1) PROGRAM(SQC47)      GROUP(SQM)
DEFINE TRANSACTION(SQR2) PROGRAM(SQC48)      GROUP(SQM)
DEFINE TRANSACTION(SQR3) PROGRAM(SQC49)      GROUP(SQM)
DEFINE TRANSACTION(SQR4) PROGRAM(SQC50)      GROUP(SQM)
DEFINE TRANSACTION(SQR5) PROGRAM(SQC51)      GROUP(SQM)
DEFINE TRANSACTION(SQR6) PROGRAM(SQC52)      GROUP(SQM)
DEFINE TRANSACTION(SQR7) PROGRAM(SQC53)      GROUP(SQM)
DEFINE TRANSACTION(SQR8) PROGRAM(SQC54)      GROUP(SQM)
DEFINE TRANSACTION(SQTU) PROGRAM(SQC60)      GROUP(SQM)
DEFINE TRANSACTION(SQTR) PROGRAM(SQC61)      GROUP(SQM)
DEFINE TRANSACTION(SQTS) PROGRAM(SQC62)      GROUP(SQM)
DEFINE TRANSACTION(SQTL) PROGRAM(SQC63)      GROUP(SQM)
DEFINE TRANSACTION(SQS1) PROGRAM(SQC64)      GROUP(SQM)
DEFINE TRANSACTION(SQTC) PROGRAM(SQC65)      GROUP(SQM)
DEFINE TRANSACTION(SQS2) PROGRAM(SQC66)      GROUP(SQM)
#*
#&
$ $$ EOJ
```

Figure 62 (Part 3 of 3). Define Control Center Programs and Transactions to CICS (SQMCSDUP.Z)

```
*********************************************************************
*                    CONTROL CENTER VSE V6.1 FCT ENTRIES (4)        *
*********************************************************************
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                    (1) CONTROL CENTER ERROR MESSAGE FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
SQMMESG  DFHFCT TYPE=FILE,                                            X
               ACCMETH=(VSAM,KSDS),                                   X
               FILE=SQMMESG,                                          X
               FILSTAT=(ENABLED,OPENED),                              X
               SERVREQ=(BROWSE),                                      X
               STRNO=2
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                    (2) CONTROL CENTER DBSPACE PARAMETER FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
SQMPARM  DFHFCT TYPE=FILE,                                            X
               ACCMETH=(VSAM,KSDS),                                   X
               FILE=SQMPARM,                                          X
               FILSTAT=(ENABLED,OPENED),                              X
               SERVREQ=(ADD,BROWSE,DELETE,UPDATE),                    X
               STRNO=2
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                    (3) CONTROL CENTER WORK FILE LABEL FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
SQMWORK  DFHFCT TYPE=FILE,                                            X
               ACCMETH=(VSAM,KSDS),                                   X
               FILE=SQMWORK,                                          X
               FILSTAT=(ENABLED,OPENED),                              X
               SERVREQ=(ADD,BROWSE,DELETE,UPDATE),                    X
               STRNO=2
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                    (4) CONTROL CENTER TABLE PARAMETER FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
SQMTPRM  DFHFCT TYPE=FILE,                                            X
               ACCMETH=(VSAM,KSDS),                                   X
               FILE=SQMTPRM,                                          X
               FILSTAT=(ENABLED,OPENED),                              X
               SERVREQ=(ADD,BROWSE,DELETE,UPDATE),                    X
               STRNO=2
```

Figure 63. Control Center FCT Entries (SQMFCT.A)

```
$ $$ JOB JNM=SQMLIBDF,CLASS=0,DISP=D,PRI=3
$ $$ LST CLASS=Q
// JOB SQMLIBDF
// EXEC IDCAMS,SIZE=AUTO
 DEFINE CLUSTER                              -
       (NAME (SQLMSTR.LIBRARY)               -
        CYL(10 1)                            -
        SHAREOPTIONS (3)                     -
        RECORDFORMAT (NOCIFORMAT)            -
        VOLUMES (XXXXXX)                     -
        NOREUSE                              -
        NONINDEXED                           -
        TO (99366))                          -
        DATA (NAME (SQLMSTR.LIBRARY.DATA ) ) -
        CATALOG (SQLMSTR.USER.CATALOG)
  IF LASTCC NE 0 THEN CANCEL JOB
/*
// OPTION STDLABEL=ADD
// DLBL SQLMSTR,'SQLMSTR.LIBRARY',,VSAM,CAT=SQMCAT,DISP=(OLD,KEEP)
/*
// EXEC IESVCLUP,SIZE=AUTO
A SQLMSTR.LIBRARY                              SQLMSTR SQMCAT
/*
// EXEC LIBR,PARM='MSHP'
DEFINE LIB=SQLMSTR    REPLACE=YES
DEFINE SUBLIB=SQLMSTR.PACKAGE     REPLACE=YES
#*
#&
$ $$ EOJ
```

Figure 64. Define the Control Center Package Library (SQMLIBDF.Z)

```
$ $$ JOB JNM=SQMCRHLP,CLASS=0,DISP=D,PRI=9
$ $$ LST CLASS=Q
// JOB SQMCRHLP
// LIBDEF *,SEARCH=PRD2.DB2610
// EXEC ARIDBS,SIZE=AUTO,PARM='D(DB2VSE61)'
READ MEMBER SQMCONN.C NOCONT
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*                                                       *'
COMMENT '*  CREATE AND LOAD THE CONTROL CENTER SQMHELP TABLE     *'
COMMENT '*                                                       *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
SET AUTOCOMMIT (ON)
SET ERRORMODE CONTINUE
COMMENT '* * * * * * *          Drop DBSPACE           * * * * *'
DROP DBSPACE PUBLIC.SQMHELP;
COMMENT '* * * * * * *          Acquire DBSPACE        * * * * *'
ACQUIRE PUBLIC DBSPACE NAMED PUBLIC.SQMHELP
(PAGES    = 128,
 PCTINDEX = 33,
 PCTFREE  = 0,
 NHEADER  = 1,
 STORPOOL = 1,
 LOCK     = PAGE);
COMMENT '* * * * * * *          Create TABLE           * * * * *'
CREATE TABLE SQLMSTR.SQMHELP
(SUBJECT            CHAR(8)     NOT NULL,
 LINE_NO            SMALLINT    NOT NULL,
 TEXT               CHAR(68)    NOT NULL)
IN PUBLIC.SQMHELP;
COMMENT '* * * * * * *          Set Automatic Upstats Off  * * *'
SET UPDATE STATISTICS (OFF)
COMMENT '* * * * * * *          Dataload TABLE         * * * * *'
DATALOAD TABLE (SQLMSTR.SQMHELP)
     SUBJECT     01-08
     LINE_NO     09-12
     TEXT        13-80
INFILE(*)
$ $$ SLI MEM=SQMHLPTX.Z,S=PRD2.SQL610
ENDDATA
COMMENT '* * * * * * *          Primary Key            * * * * *'
CREATE UNIQUE INDEX SQLMSTR.SQMHELP_INDX1
ON SQLMSTR.SQMHELP
(SUBJECT            ASC,
 LINE_NO            ASC)
PCTFREE = 0;
COMMENT '* * * * * * *          Update All Statistics  * * * * *'
UPDATE ALL STATISTICS FOR DBSPACE
PUBLIC.SQMHELP;
COMMENT '* * * * * * *          Table Grants           * * * * *'
GRANT SELECT ON SQLMSTR.SQMHELP
TO PUBLIC;
#*
#&
$ $$ EOJ
```

Figure 65. Define and Load the Control Center Help Table (SQMCRHLP.Z)

Appendix B. Reorganization Job Streams

```
* $$ JOB JNM=UNLOAD,CLASS=0,DISP=D,NTFY=YES
* $$ LST PRI=3
// JOB UNLOAD MUM UNLOAD DBSPACE TO DISK
// OPTION LOG
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0001 RECORD TIME BEFORE DDL GENERATION
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%DB2VSE51             PUBLIC  SQMHELP           2N 1
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0002 GENERATE DDL FOR "PUBLIC"."SQMHELP"
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMDDL,'L.DB2VSE51.PUBLIC.SQMHELP',0,VSAM,                    X
               RECORDS=001000,RECSIZE=80,DISP=(NEW,KEEP),CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB01,SIZE=AUTO
%%DB2VSE51             PUBLIC  SQMHELP           2 N
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0003 UNLOAD DBSPACE "PUBLIC"."SQMHELP"
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,SYS302,1,0,02985,150
// ASSGN SYS007,DISK,VOL=SYS302,SHR
// EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(DB2VSE51)'
READ MEMBER CONNECT.C NOCONT
COMMENT '***** UNLOAD DBSPACE "PUBLIC"."SQMHELP" *****'
SET UPDATE STATISTICS OFF;
UNLOAD DBSPACE ("PUBLIC"."SQMHELP")
OUTFILE(SQMDAT1 BLKSZ(02048) PDEV(DASD))
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
```

Figure 66 (Part 1 of 2). DBSPACE Reorg Option 2 (Unload DBSPACE) - Sample Jobstream

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0004 RECORD TIME AFTER UNLOAD
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%DB2VSE51             PUBLIC  SQMHELP           2N 2
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0005 REPRO SAM DATA TO VSAM
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,SYS302,1,0,02985,150
// ASSGN SYS007,DISK,VOL=SYS302,SHR
// DLBL SQMDAT,'D.DB2VSE51.PUBLIC.SQMHELP',0,VSAM,                     X
               RECORDS=000100,RECSIZE=8240,DISP=(NEW,KEEP),CAT=SQMCAT
// EXEC IDCAMS,SIZE=AUTO
   REPRO INFILE(SQMDAT1  -
                ENV(RECFM(SB) -
                    BLKSZ(2048) -
                    RECSZ(8240))) -
         OUTFILE(SQMDAT  -
                ENV(RECFM(VB) -
                    BLKSZ(8248) -
                    RECSZ(8240)))
/*
/&
* $$ EOJ
```

Figure 66 (Part 2 of 2). DBSPACE Reorg Option 2 (Unload DBSPACE) - Sample Jobstream

```
* $$ JOB JNM=DISKREOR,CLASS=0,DISP=D,NTFY=YES
* $$ LST PRI=3
// JOB DISKREOR MUM REORG DBSPACE VIA DISK
// OPTION LOG
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0001 RECORD TIME BEFORE DDL GENERATION
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%DB2VSE51             PUBLIC  SQMHELP           3N 1
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0002 GENERATE DDL FOR "PUBLIC"."SQMHELP"
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMDDL,'L.DB2VSE51.PUBLIC.SQMHELP',0,VSAM,                    X
               RECORDS=001000,RECSIZE=80,DISP=(NEW,KEEP),CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB01,SIZE=AUTO
%%DB2VSE51             PUBLIC  SQMHELP           3 N
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0003 UNLOAD DBSPACE "PUBLIC"."SQMHELP"
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,SYS302,1,0,02985,150
// ASSGN SYS007,DISK,VOL=SYS302,SHR
// EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(DB2VSE51)'
READ MEMBER SQMCONN.C NOCONT
COMMENT '***** UNLOAD DBSPACE "PUBLIC"."SQMHELP" *****'
SET UPDATE STATISTICS OFF;
UNLOAD DBSPACE ("PUBLIC"."SQMHELP")
OUTFILE(SQMDAT1 BLKSZ(02048) PDEV(DASD))
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0004 RECORD TIME AFTER UNLOAD
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%DB2VSE51             PUBLIC  SQMHELP           3N 2
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
```

Figure 67 (Part 1 of 3). DBSPACE Reorg Option 3 (Reorganize DBSPACE) - Sample Jobstream

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0005 REPRO VSAM DDL TO SAM
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL VSAMIN,'L.DB2VSE51.PUBLIC.SQMHELP',,VSAM,                     X
              CAT=SQMCAT,DISP=(OLD,KEEP)
// DLBL SQMDDL1,'SQLMSTR.DDL.FILE1',0,SD
// EXTENT SYS008,SYS302,1,0,03835,100
// ASSGN SYS008,DISK,VOL=SYS302,SHR
// EXEC IDCAMS,SIZE=AUTO
   REPRO INFILE(VSAMIN ) -
        OUTFILE(SQMDDL1 -
                ENV(RECFM(F ) -
                    BLKSZ(0080) -
                    RECSZ(0080)))
   IF LASTCC > 0000 -
        THEN CANCEL JOB
/*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0006 RECORD TIME BEFORE RELOAD
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%DB2VSE51             PUBLIC  SQMHELP           3N 3
/*
 // IF $RC > 0000 THEN
 // GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0007 RELOAD DBSPACE "PUBLIC"."SQMHELP"
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,SYS302,1,0,02985,150
// ASSGN SYS007,DISK,VOL=SYS302,SHR
// ASSGN SYS004,SYS005
// ASSGN SYS020,SYSIPT
// DLBL IJSYSIN,'SQLMSTR.DDL.FILE1',0,SD
// EXTENT SYSIPT,SYS302,1,0,03835,100
ASSGN SYSIPT,DISK,VOL=SYS302,SHR
// EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(DB2VSE51)'
/*
// IF $RC > 0006 THEN
// GOTO CLOSEIPT
CLOSE SYSIPT,SYS020
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0008 RECORD TIME AFTER RELOAD
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%DB2VSE51             PUBLIC  SQMHELP           3N 4
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
```

Figure 67 (Part 2 of 3). DBSPACE Reorg Option 3 (Reorganize DBSPACE) - Sample Jobstream

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0009 DELETE VSAM DDL FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// ASSGN SYSLST,IGN
// DLBL FILEIN,'L.DB2VSE51.PUBLIC.SQMHELP',,VSAM,                      X
               CAT=SQMCAT,DISP=(OLD,DELETE)
// EXEC IDCAMS,SIZE=AUTO
   PRINT INFILE(FILEIN) -
        COUNT(1)
/*
// RESET SYSLST
// GOTO THEEND
/. CLOSEIPT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0010 CLOSE SYSIPT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CLOSE SYSIPT,SYS020
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0011 REPRO SAM DATA TO VSAM
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,SYS302,1,0,02985,150
// ASSGN SYS007,DISK,VOL=SYS302,SHR
// DLBL SQMDAT,'D.DB2VSE51.PUBLIC.SQMHELP',0,VSAM,                     X
               RECORDS=000100,RECSIZE=8240,DISP=(NEW,KEEP),CAT=SQMCAT
// EXEC IDCAMS,SIZE=AUTO
   REPRO INFILE(SQMDAT1  -
                ENV(RECFM(SB) -
                    BLKSZ(2048) -
                    RECSZ(8240))) -
        OUTFILE(SQMDAT  -
                ENV(RECFM(VB) -
                    BLKSZ(8248) -
                    RECSZ(8240)))
   IF LASTCC > 0000 -
        THEN CANCEL JOB
/*
// GOTO $EOJ
/. THEEND
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0012 THE END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/*
/&
* $$ EOJ
```

Figure 67 (Part 3 of 3). DBSPACE Reorg Option 3 (Reorganize DBSPACE) - Sample Jobstream

```
* $$ JOB JNM=RELOAD,CLASS=0,DISP=D,NTFY=YES
* $$ LST PRI=3
// JOB RELOAD MUM RELOAD DBSPACE FROM DISK
// OPTION LOG
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0001 CHECK FOR DATA FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// ASSGN SYSLST,IGN
// DLBL FILEIN,'D.DB2VSE51.PUBLIC.SQMHELP',,VSAM,                     X
               CAT=SQMCAT
// EXEC IDCAMS,SIZE=AUTO
   PRINT INFILE(FILEIN) -
        COUNT(1)
/*
// IF $RC > 0000 THEN
// GOTO NODAT
// RESET SYSLST
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0002 CHECK FOR DDL FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// ASSGN SYSLST,IGN
// DLBL FILEIN,'L.DB2VSE51.PUBLIC.SQMHELP',,VSAM,                     X
               CAT=SQMCAT
// EXEC IDCAMS,SIZE=AUTO
   PRINT INFILE(FILEIN) -
        COUNT(1)
/*
// IF $RC > 0000 THEN
// GOTO NODDL
// RESET SYSLST
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0003 REPRO VSAM DATA TO SAM
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL VSAMIN,'D.DB2VSE51.PUBLIC.SQMHELP',,VSAM,                     X
               CAT=SQMCAT,DISP=(OLD,KEEP)
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,SYS302,1,0,02985,150
// ASSGN SYS007,DISK,VOL=SYS302,SHR
// EXEC IDCAMS,SIZE=AUTO
   REPRO INFILE(VSAMIN   -
                ENV(RECFM(VB) -
                    BLKSZ(8248) -
                    RECSZ(8240))) -
        OUTFILE(SQMDAT1 -
                ENV(RECFM(SB) -
                    BLKSZ(2048) -
                    RECSZ(8240)))
   IF LASTCC > 0000 -
        THEN CANCEL JOB
/*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0004 RECORD TIME BEFORE RELOAD
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%DB2VSE51            PUBLIC  SQMHELP          4N 3
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
```

Figure 68 (Part 1 of 3). DBSPACE Reorg Option 4 (Reload DBSPACE) - Sample Jobstream

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0005 REPRO VSAM DDL TO SAM
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL VSAMIN,'L.DB2VSE51.PUBLIC.SQMHELP',,VSAM,                     X
              CAT=SQMCAT,DISP=(OLD,KEEP)
// DLBL SQMDDL1,'SQLMSTR.DDL.FILE1',0,SD
// EXTENT SYS008,SYS302,1,0,03835,100
// ASSGN SYS008,DISK,VOL=SYS302,SHR
// EXEC IDCAMS,SIZE=AUTO
   REPRO INFILE(VSAMIN ) -
        OUTFILE(SQMDDL1 -
                ENV(RECFM(F ) -
                    BLKSZ(0080) -
                    RECSZ(0080)))
   IF LASTCC > 0000 -
        THEN CANCEL JOB
/*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0006 RELOAD DBSPACE "PUBLIC"."SQMHELP"
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,SYS302,1,0,02985,150
// ASSGN SYS007,DISK,VOL=SYS302,SHR
// ASSGN SYS006,SYS007
// ASSGN SYS020,SYSIPT
// DLBL IJSYSIN,'SQLMSTR.DDL.FILE1',0,SD
// EXTENT SYSIPT,SYS302,1,0,03835,100
ASSGN SYSIPT,DISK,VOL=SYS302,SHR
// EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(DB2VSE51)'
/*
// IF $RC > 0000 THEN
// GOTO CLOSEIPT
CLOSE SYSIPT,SYS020
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0007 RECORD TIME AFTER RELOAD
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%DB2VSE51             PUBLIC  SQMHELP           4N 4
/*
// GOTO THEEND
/. CLOSEIPT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0008 CLOSE SYSIPT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CLOSE SYSIPT,SYS020
// GOTO $EOJ
/. NODAT
// RESET SYSLST
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0009 NO DATA FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* RELOAD ABORTED: MISSING FILE = D.DB2VSE51.PUBLIC.SQMHELP
// GOTO $EOJ
/. NODDL
// RESET SYSLST
```

Figure 68 (Part 2 of 3). DBSPACE Reorg Option 4 (Reload DBSPACE) - Sample Jobstream

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0010 NO DDL FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* RELOAD ABORTED: MISSING FILE = L.DB2VSE51.PUBLIC.SQMHELP
/. THEEND
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0011 THE END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/*
/&
```

Figure 68 (Part 3 of 3). DBSPACE Reorg Option 4 (Reload DBSPACE) - Sample Jobstream

```
* $$ JOB JNM=TABLEUNL,CLASS=0,DISP=D,NTFY=YES
* $$ LST PRI=3
// JOB TABLEUNL UNLOAD TABLE TO DISK
// OPTION LOG
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0001 GENERATE DDL FOR "SQLDBA"."ACTIVITY"
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMTPRM,'SQLMSTR.TABLE.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMDDL,'L.SQLDS610.SQLDBA.ACTIVITY',0,VSAM,                    X
               RECORDS=001000,RECSIZE=80,DISP=(NEW,KEEP),CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB60,SIZE=AUTO
%%SQLDS610          SQLDBA  ACTIVITY          2
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0002 UNLOAD TABLE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,'VSEPK3',1,0,0000201438,06300
// ASSGN SYS007,DISK,VOL='VSEPK3',SHR
// EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(SQLDS510)'
READ MEMBER SQMCONN.C NOCONT
COMMENT '***** UNLOAD TABLE "SQLDBA"."ACTIVITY" *****'
SET UPDATE STATISTICS OFF;
UNLOAD TABLE ("SQLDBA"."ACTIVITY")
OUTFILE(SQMDAT1 BLKSZ(02048) PDEV(DASD))
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0003 REPRO SAM DATA TO VSAM
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,'VSEPK3',1,0,0000201438,06300
// ASSGN SYS007,DISK,VOL='VSEPK3',SHR
// DLBL SQMDAT,'D.SQLDS510.SQLDBA.ACTIVITY',0,VSAM,                    X
               RECORDS=000100,RECSIZE=8240,DISP=(NEW,KEEP),CAT=SQMCAT
// EXEC IDCAMS,SIZE=AUTO
   REPRO INFILE(SQMDAT1           -
                ENV(RECFM(SB)     -
                    BLKSZ(2048)   -
                    RECSZ(8240))) -
         OUTFILE(SQMDAT           -
                 ENV(RECFM(VB)    -
                     BLKSZ(8248)  -
                     RECSZ(8240)))
/*
/&
* $$ EOJ
```

Figure 69. TABLE Reorg Option 2 (Unload TABLE) - Sample Jobstream

```
* $$ JOB JNM=TABREORG,CLASS=A,PRI=3,DISP=D
* $$ LST PRI=3
// JOB TABREORG REORG TABLE - DISK
// OPTION LOG
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0001 GENERATE DDL FOR "SQLDBA"."ACTIVITY"
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMTPRM,'SQLMSTR.TABLE.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMDDL,'L.SQLDS610.SQLDBA.ACTIVITY',0,VSAM,                    X
               RECORDS=001000,RECSIZE=80,DISP=(NEW,KEEP),CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB60,SIZE=AUTO
%%SQLDS610          SQLDBA  ACTIVITY          3NN
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0002 UNLOAD DATA
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,'VSEPK3',1,0,0000201438,06300
// ASSGN SYS007,DISK,VOL='VSEPK3',SHR
// EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(SQLDS510)'
READ MEMBER SQMCONN.C NOCONT
COMMENT '***** UNLOAD TABLE "SQLDBA"."ACTIVITY" *****'
SET UPDATE STATISTICS OFF;
UNLOAD TABLE ("SQLDBA"."ACTIVITY")
OUTFILE(SQMDAT1 BLKSZ(02048) PDEV(DASD))
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0003 REPRO VSAM DDL TO SAM
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL VSAMIN,'L.SQLDS610.SQLDBA.ACTIVITY',,VSAM,                     X
               CAT=SQMCAT,DISP=(OLD,KEEP)
// DLBL SQMDDL1,'SQLMSTR.DDL.FILE1',0,SD
// EXTENT SYS008,'VSEPK3',1,0,0000239238,00900
// ASSGN SYS008,DISK,VOL='VSEPK3',SHR
// EXEC IDCAMS,SIZE=AUTO
   REPRO INFILE(VSAMIN )          -
        OUTFILE(SQMDDL1           -
                ENV(RECFM(F )     -
                    BLKSZ(0080)   -
                    RECSZ(0080)))
   IF LASTCC > 0000 -
        THEN CANCEL JOB
/*
```

Figure 70 (Part 1 of 3). TABLE Reorg Option 3 (Reorganize TABLE) - Sample Jobstream

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0004 RELOAD TABLE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,'VSEPK3',1,0,0000201438,06300
// ASSGN SYS007,DISK,VOL='VSEPK3',SHR
// ASSGN SYS004,SYS005
// ASSGN SYS020,SYSIPT
// DLBL IJSYSIN,'SQLMSTR.DDL.FILE1',0,SD
// EXTENT SYSIPT,'VSEPK3',1,0,0000239238,00900
// ASSGN SYSIPT,DISK,VOL='VSEPK3',SHR
// ON $ABEND GOTO CLOSEIPT
// ON $CANCEL GOTO CLOSEIPT
// EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(SQLDS610)'
/*
// IF $RC > 0006 THEN
// GOTO CLOSEIPT
CLOSE SYSIPT,SYS020
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0005 DELETE PARAMETER RECORD
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMTPRM,'SQLMSTR.TABLE.PARMS',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB62,SIZE=AUTO
%%SQLDS510          SQLDBA  ACTIVITY
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0006 DELETE VSAM DDL FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// ASSGN SYSLST,IGN
// DLBL FILEIN,'L.SQLDS610.SQLDBA.ACTIVITY',,VSAM,                    X
               CAT=SQMCAT,DISP=(OLD,DELETE)
// EXEC IDCAMS,SIZE=AUTO
   PRINT INFILE(FILEIN) -
         COUNT(1)
/*
// RESET SYSLST
// GOTO THEEND
/. CLOSEIPT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0007 CLOSE SYSIPT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CLOSE SYSIPT,SYS020
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0008 REPRO SAM DATA TO VSAM
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Figure 70 (Part 2 of 3). TABLE Reorg Option 3 (Reorganize TABLE) - Sample Jobstream

```
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,'VSEPK3',1,0,0000201438,06300
// ASSGN SYS007,DISK,VOL='VSEPK3',SHR
// DLBL SQMDAT,'D.SQLDS610.SQLDBA.ACTIVITY',0,VSAM,                    X
              RECORDS=000100,RECSIZE=8240,DISP=(NEW,KEEP),CAT=SQMCAT
// EXEC IDCAMS,SIZE=AUTO
   REPRO INFILE(SQMDAT1           -
                ENV(RECFM(SB)     -
                    BLKSZ(2048)   -
                    RECSZ(8240))) -
         OUTFILE(SQMDAT           -
                ENV(RECFM(VB)     -
                    BLKSZ(8248)   -
                    RECSZ(8240)))
   IF LASTCC > 0000               -
        THEN CANCEL JOB
/*
// GOTO $EOJ
/. THEEND
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0009 THE END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/*
/&
* $$ EOJ
```

Figure 70 (Part 3 of 3). TABLE Reorg Option 3 (Reorganize TABLE) - Sample Jobstream

```
* $$ JOB JNM=TABLERLD,CLASS=A,PRI=3,DISP=D
* $$ LST PRI=3
// JOB TABLERLD RELOAD TABLE DISK
// OPTION LOG
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0001 CHECK FOR DATA FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// ASSGN SYSLST,IGN
// DLBL FILEIN,'D.SQLDS610.SQLDBA.ACTIVITY',,VSAM,                    X
               CAT=SQMCAT
// EXEC IDCAMS,SIZE=AUTO
   PRINT INFILE(FILEIN) -
        COUNT(1)
/*
// IF $RC > 0000 THEN
// GOTO NODAT
// RESET SYSLST
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0002 CHECK FOR DDL FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// ASSGN SYSLST,IGN
// DLBL FILEIN,'L.SQLDS610.SQLDBA.ACTIVITY',,VSAM,                    X
               CAT=SQMCAT
// EXEC IDCAMS,SIZE=AUTO
   PRINT INFILE(FILEIN) -
        COUNT(1)
/*
// IF $RC > 0000 THEN
// GOTO NODDL
// RESET SYSLST
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0003 REPRO VSAM DATA TO SAM
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL VSAMIN,'D.SQLDS610.SQLDBA.ACTIVITY',,VSAM,                    X
               CAT=SQMCAT,DISP=(OLD,KEEP)
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,'VSEPK3',1,0,0000201438,06300
// ASSGN SYS007,DISK,VOL='VSEPK3',SHR
// EXEC IDCAMS,SIZE=AUTO
   REPRO INFILE(VSAMIN               -
                ENV(RECFM(VB)        -
                    BLKSZ(8248)      -
                    RECSZ(8240)))    -
         OUTFILE(SQMDAT1             -
                 ENV(RECFM(SB)       -
                     BLKSZ(2048)     -
                     RECSZ(8240)))
   IF LASTCC > 0000 -
        THEN CANCEL JOB
/*
```

Figure 71 (Part 1 of 3). TABLE Reorg Option 4 (Reload TABLE) - Sample Jobstream

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0004 REPRO VSAM DDL TO SAM
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL VSAMIN,'L.SQLDS610.SQLDBA.ACTIVITY',,VSAM,                    X
              CAT=SQMCAT,DISP=(OLD,KEEP)
// DLBL SQMDDL1,'SQLMSTR.DDL.FILE1',0,SD
// EXTENT SYS008,'VSEPK3',1,0,0000239238,00900
// ASSGN SYS008,DISK,VOL='VSEPK3',SHR
// EXEC IDCAMS,SIZE=AUTO
   REPRO INFILE(VSAMIN )           -
        OUTFILE(SQMDDL1            -
                ENV(RECFM(F )      -
                    BLKSZ(0080)    -
                    RECSZ(0080)))
   IF LASTCC > 0000 -
        THEN CANCEL JOB
/*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0005 RELOAD TABLE "SQLDBA"."ACTIVITY"
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,'VSEPK3',1,0,0000201438,06300
// ASSGN SYS007,DISK,VOL='VSEPK3',SHR
// ASSGN SYS006,SYS007
// ASSGN SYS020,SYSIPT
// DLBL IJSYSIN,'SQLMSTR.DDL.FILE1',0,SD
// EXTENT SYSIPT,'VSEPK3',1,0,0000239238,00900
ASSGN SYSIPT,DISK,VOL='VSEPK3',SHR
// ON $ABEND GOTO CLOSEIPT
// ON $CANCEL GOTO CLOSEIPT
// EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(SQLDS510)'
/*
// IF $RC > 0006 THEN
// GOTO CLOSEIPT
CLOSE SYSIPT,SYS020
// GOTO THEEND
/. CLOSEIPT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0006 CLOSE SYSIPT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CLOSE SYSIPT,SYS020
// GOTO $EOJ
/.  NODAT // RESET SYSLST
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0007 NO DATA FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* RELOAD ABORTED: MISSING FILE =
* D.SQLDS510.SQLDBA.ACTIVITY
// GOTO $EOJ
/.  NODDL
// RESET SYSLST
```

Figure 71 (Part 2 of 3). TABLE Reorg Option 4 (Reload TABLE) - Sample Jobstream

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0008 NO DDL FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* RELOAD ABORTED: MISSING FILE =
* L.SQLDS510.SQLDBA.ACTIVITY
/. THEEND
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0009 THE END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/*
/&
* $$ EOJ
```

Figure 71 (Part 3 of 3). TABLE Reorg Option 4 (Reload TABLE) - Sample Jobstream

```
* $$ JOB JNM=SUMREORG,CLASS=8,DISP=D,NTFY=YES
* $$ LST PRI=3
// JOB SUMREORG SUM REORG DBSPACE VIA DISK
// OPTION LOG
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0001 RECORD TIME BEFORE DDL GENERATION
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// ON $RC = 4095 CONTINUE
// EXEC ARISQLDS,SIZE=AUTO,PARM='SYSMODE=S,LOGMODE=Y,DBNAME=DB2VSE61,PRX
               OGNAME=SQB02,NDIRBUF=000100,NPAGBUF=000100'
%%DB2VSE61             PUBLIC  SQMHELP           3NY1
/*
// IF $RC > 0000 AND $RC < 4095 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0002 GENERATE DDL FOR "PUBLIC"."SQMHELP"
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMDDL,'L.DB2VSE61.PUBLIC.SQMHELP',0,VSAM,                     X
               RECORDS=001000,RECSIZE=80,DISP=(NEW,KEEP),CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC PROC=ARIS34DB
// EXEC ARISQLDS,SIZE=AUTO,PARM='SYSMODE=S,LOGMODE=Y,DBNAME=DB2VSE61,PRX
               OGNAME=SQB01,NDIRBUF=000100,NPAGBUF=000100'
%%DB2VSE61             PUBLIC  SQMHELP           3 Y
/*
// IF $RC > 0000 AND $RC < 4095 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0003 UNLOAD DBSPACE "PUBLIC"."SQMHELP"
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,SYS302,1,0,02985,150
// ASSGN SYS007,DISK,VOL=SYS302,SHR
// EXEC PROC=ARIS34DB
// EXEC ARISQLDS,SIZE=AUTO,PARM='SYSMODE=S,LOGMODE=Y,DBNAME=DB2VSE61,PRX
               OGNAME=ARIDBS,NDIRBUF=000030,NPAGBUF=000030'
READ MEMBER SQMCONN.C NOCONT
COMMENT '***** UNLOAD DBSPACE "PUBLIC"."SQMHELP" *****'
SET UPDATE STATISTICS OFF;
UNLOAD DBSPACE ("PUBLIC"."SQMHELP")
OUTFILE(SQMDAT1 BLKSZ(02048) PDEV(DASD))
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
```

Figure 72 (Part 1 of 3). Sample Single User Mode DBSPACE Reorganization Jobstream

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0004 RECORD TIME AFTER UNLOAD
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%DB2VSE61             PUBLIC  SQMHELP           3N 2
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0005 REPRO VSAM DDL TO SAM
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL VSAMIN,'L.DB2VSE61.PUBLIC.SQMHELP',,VSAM,                     X
               CAT=SQMCAT,DISP=(OLD,KEEP)
// DLBL SQMDDL1,'SQLMSTR.DDL.FILE1',0,SD
// EXTENT SYS008,SYS302,1,0,03835,100
// ASSGN SYS008,DISK,VOL=SYS302,SHR
// EXEC IDCAMS,SIZE=AUTO
   REPRO INFILE(VSAMIN ) -
        OUTFILE(SQMDDL1 -
                ENV(RECFM(F ) -
                    BLKSZ(0080) -
                    RECSZ(0080)))
   IF LASTCC > 0000 -
        THEN CANCEL JOB
/*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0006 RECORD TIME BEFORE RELOAD
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%DB2VSE61             PUBLIC  SQMHELP           3N 3
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0007 RELOAD DBSPACE "PUBLIC"."SQMHELP"
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,SYS302,1,0,02985,150
// ASSGN SYS007,DISK,VOL=SYS302,SHR
// ASSGN SYS004,SYS005
// ASSGN SYS020,SYSIPT
// DLBL IJSYSIN,'SQLMSTR.DDL.FILE1',0,SD
// EXTENT SYSIPT,SYS302,1,0,03835,100
```

Figure 72 (Part 2 of 3). Sample Single User Mode DBSPACE Reorganization Jobstream

```
ASSGN SYSIPT,DISK,VOL=SYS302,SHR
// EXEC PROC=ARIS34DB
// EXEC ARISQLDS,SIZE=AUTO,PARM='SYSMODE=S,LOGMODE=Y,DBNAME=DB2VSE61,PRX
               OGNAME=ARIDBS,NDIRBUF=000030,NPAGBUF=000030'
/*
// IF $RC > 0006 THEN
// GOTO CLOSEIPT
CLOSE SYSIPT,SYS020
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0008 RECORD TIME AFTER RELOAD
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC PROC=ARIS34DB
// EXEC ARISQLDS,SIZE=AUTO,PARM='SYSMODE=S,LOGMODE=Y,DBNAME=DB2VSE61,PRX
               OGNAME=SQB02,NDIRBUF=000030,NPAGBUF=000030'
%%DB2VSE61             PUBLIC  SQMHELP           3NY4
/*
// IF $RC > 0000 AND $RC < 4095 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0009 DELETE VSAM DDL FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// ASSGN SYSLST,IGN
// DLBL FILEIN,'L.DB2VSE61.PUBLIC.SQMHELP',,VSAM,                     X
               CAT=SQMCAT,DISP=(OLD,DELETE)
// EXEC IDCAMS,SIZE=AUTO
   PRINT INFILE(FILEIN) -
        COUNT(1)
/*
// RESET SYSLST
// GOTO THEEND
/. CLOSEIPT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0010 CLOSE SYSIPT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CLOSE SYSIPT,SYS020
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0011 REPRO SAM DATA TO VSAM
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,SYS302,1,0,02985,150
// ASSGN SYS007,DISK,VOL=SYS302,SHR
// DLBL SQMDAT,'D.DB2VSE61.PUBLIC.SQMHELP',0,VSAM,                     X
               RECORDS=000100,RECSIZE=8240,DISP=(NEW,KEEP),CAT=SQMCAT
// EXEC IDCAMS,SIZE=AUTO
   REPRO INFILE(SQMDAT1  -
                ENV(RECFM(SB) -
                    BLKSZ(2048) -
                    RECSZ(8240))) -
         OUTFILE(SQMDAT  -
                 ENV(RECFM(VB) -
                     BLKSZ(8248) -
                     RECSZ(8240)))
   IF LASTCC > 0000 -
        THEN CANCEL JOB
/*
// GOTO $EOJ
/. THEEND
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0012 THE END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/*
/&
* $$ EOJ
```

Figure 72 (Part 3 of 3). Sample Single User Mode DBSPACE Reorganization Jobstream

```
* $$ JOB JNM=TAPEREOM,CLASS=0,DISP=D,NTFY=YES
* $$ LST PRI=3
// JOB TAPEREOM MUM REORG DBSPACE VIA TAPE
// OPTION LOG
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0001 RECORD TIME BEFORE DDL GENERATION
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%DB2VSE61             PUBLIC  SQMHELP           3N 1
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0002 GENERATE DDL FOR "PUBLIC"."SQMHELP"
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMDDL,'L.DB2VSE61.PUBLIC.SQMHELP',0,VSAM,                    X
               RECORDS=001000,RECSIZE=80,DISP=(NEW,KEEP),CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB01,SIZE=AUTO
%%DB2VSE61             PUBLIC  SQMHELP           3 N
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0003 UNLOAD DBSPACE "PUBLIC"."SQMHELP"
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// TLBL SQMTAPE,'SQMHELP'
// ASSGN SYS005,CARTRIDGE
// MTC REW,SYS005
// EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(DB2VSE61)'
READ MEMBER SQMCONN.C NOCONT
COMMENT '***** UNLOAD DBSPACE "PUBLIC"."SQMHELP" *****'
SET UPDATE STATISTICS OFF;
UNLOAD DBSPACE ("PUBLIC"."SQMHELP")
OUTFILE(SQMTAPE BLKSZ(32760) PDEV(TAPE))
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0004 RECORD TIME AFTER UNLOAD
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%DB2VSE61             PUBLIC  SQMHELP           3N 2
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
```

Figure 73 (Part 1 of 3). Sample Single User Mode DBSPACE Reorganization using Tape

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0005 REPRO VSAM DDL TO SAM
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL VSAMIN,'L.DB2VSE61.PUBLIC.SQMHELP',,VSAM,                       X
              CAT=SQMCAT,DISP=(OLD,KEEP)
// DLBL SQMDDL1,'SQLMSTR.DDL.FILE1',0,SD
// EXTENT SYS008,SYS302,1,0,03835,100
// ASSGN SYS008,DISK,VOL=SYS302,SHR
// EXEC IDCAMS,SIZE=AUTO
   REPRO INFILE(VSAMIN ) -
        OUTFILE(SQMDDL1 -
                ENV(RECFM(F ) -
                    BLKSZ(0080) -
                    RECSZ(0080)))
   IF LASTCC > 0000 -
        THEN CANCEL JOB
/*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0006 RECORD TIME BEFORE RELOAD
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%DB2VSE61             PUBLIC  SQMHELP           3N 3
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0007 RELOAD DBSPACE "PUBLIC"."SQMHELP"
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// TLBL SQMTAPE,'SQMHELP'
// ASSGN SYS005,CARTRIDGE
// ASSGN SYS004,SYS005
// ASSGN SYS020,SYSIPT
// DLBL IJSYSIN,'SQLMSTR.DDL.FILE1',0,SD
// EXTENT SYSIPT,SYS302,1,0,03835,100
ASSGN SYSIPT,DISK,VOL=SYS302,SHR
// EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(DB2VSE61)'
/*
// IF $RC > 0006 THEN
// GOTO CLOSEIPT
CLOSE SYSIPT,SYS020
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0008 RECORD TIME AFTER RELOAD
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%DB2VSE61             PUBLIC  SQMHELP           3N 4
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
```

Figure 73 (Part 2 of 3). Sample Single User Mode DBSPACE Reorganization using Tape

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0009 DELETE VSAM DDL FILE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// ASSGN SYSLST,IGN
// DLBL FILEIN,'L.DB2VSE61.PUBLIC.SQMHELP',,VSAM,                        X
               CAT=SQMCAT,DISP=(OLD,DELETE)
// EXEC IDCAMS,SIZE=AUTO
   PRINT INFILE(FILEIN) -
        COUNT(1)
/*
// RESET SYSLST
// GOTO THEEND
/. CLOSEIPT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0010 CLOSE SYSIPT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
CLOSE SYSIPT,SYS020
/. THEEND
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0011 THE END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
/*
/&
* $$ EOJ
```

Figure 73 (Part 3 of 3). Sample Single User Mode DBSPACE Reorganization using Tape

Appendix C. Control Center Packages

The Control Center distribution library contains database PACKAGES, control structures produced during preparation of the Control Center programs that are used to execute SQL statements. Each PACKAGE relates to a Control Center PHASE. These members need not be imported into your editor. The PACKAGE reload job, SQMRLDPK, reads them directly from the production library. These PACKAGEs must be loaded into each database with which you wish to use Control Center. Listed below are some of the Control Center PACKAGEs supplied on the distribution library. Note that member type of the PACKAGE supplied by Control Center is "Q".

1. SQB01.Q Main batch DDL generator
2. SQB02.Q Batch reorg timekeeping and recording
3. SQB05.Q Batch update statistics timekeeping and recording
4. SQC01.Q Main menu
5. SQC02.Q Help detail display
6. SQC05.Q DBSPACE reorganization
7. SQC06.Q DBSPACE reorganization submission
8. SQC07.Q Help menu
9. SQC08.Q DBSPACE analysis menu
10. SQC10.Q DBSPACE analysis submission
11. SQC11.Q Operator command menu
12. SQC12.Q Operator command display

Figure 74 on page 158 provides an example of the library member, SQMRLDPK.Z, described in Chapter 2, “Installing Control Center” on page 7. To load these packages, follow the instructions given in “Step 16: Load Packages into Server(s)” on page 18.

```
$ $$ JOB JNM=SQMRLDPK,CLASS=0,DISP=D,PRI=9
$ $$ LST CLASS=Q
// JOB SQMRLDPK RELOAD CONTROL CENTER PACKAGES
// OPTION LOG
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* STEP0001 RELOAD CONTROL CENTER PACKAGES
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
// LIBDEF *,SEARCH=PRD2.DB2610
// EXEC ARIDBS,SIZE=AUTO,PARM='D(DB2VSE51)'
READ MEMBER SQMCONN.C NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQB01 *****'
RELOAD PACKAGE (SQLMSTR.SQB01)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQB01.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQB02 *****'
RELOAD PACKAGE (SQLMSTR.SQB02)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQB02.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQB05 *****'
RELOAD PACKAGE (SQLMSTR.SQB05)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQB05.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQB60 *****'
RELOAD PACKAGE (SQLMSTR.SQB60)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQB60.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC01 *****'
RELOAD PACKAGE (SQLMSTR.SQC01)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC01.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC02 *****'
RELOAD PACKAGE (SQLMSTR.SQC02)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC02.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC05 *****'
RELOAD PACKAGE (SQLMSTR.SQC05)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC05.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC06 *****'
RELOAD PACKAGE (SQLMSTR.SQC06)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC06.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC07 *****'
RELOAD PACKAGE (SQLMSTR.SQC07)
     REPLACE
     KEEP
```

Figure 74 (Part 1 of 5). Load the Control Center Packages, SQMRLDPK.Z

```
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC07.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC08 *****'
RELOAD PACKAGE (SQLMSTR.SQC08)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC08.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC10 *****'
RELOAD PACKAGE (SQLMSTR.SQC10)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC10.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC11 *****'
RELOAD PACKAGE (SQLMSTR.SQC11)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC11.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC12 *****'
RELOAD PACKAGE (SQLMSTR.SQC12)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC12.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC16 *****'
RELOAD PACKAGE (SQLMSTR.SQC16)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC16.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC17 *****'
RELOAD PACKAGE (SQLMSTR.SQC17)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC17.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC18 *****'
RELOAD PACKAGE (SQLMSTR.SQC18)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC18.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC19 *****'
RELOAD PACKAGE (SQLMSTR.SQC19)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC19.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC20 *****'
RELOAD PACKAGE (SQLMSTR.SQC20)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC20.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC21 *****'
RELOAD PACKAGE (SQLMSTR.SQC21)
     REPLACE
     KEEP
```

Figure 74 (Part 2 of 5). Load the Control Center Packages, SQMRLDPK.Z

```
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC21.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC22 *****'
RELOAD PACKAGE (SQLMSTR.SQC22)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC22.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC23 *****'
RELOAD PACKAGE (SQLMSTR.SQC23)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC23.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC24 *****'
RELOAD PACKAGE (SQLMSTR.SQC24)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC24.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC25 *****'
RELOAD PACKAGE (SQLMSTR.SQC25)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC25.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC26 *****'
RELOAD PACKAGE (SQLMSTR.SQC26)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC26.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC27 *****'
RELOAD PACKAGE (SQLMSTR.SQC27)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC27.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC28 *****'
RELOAD PACKAGE (SQLMSTR.SQC28)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC28.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC29 *****'
RELOAD PACKAGE (SQLMSTR.SQC29)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC29.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC40 *****'
RELOAD PACKAGE (SQLMSTR.SQC40)
     REPLACE
     KEEP
```

Figure 74 (Part 3 of 5). Load the Control Center Packages, SQMRLDPK.Z

```
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC40.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC41 *****'
RELOAD PACKAGE (SQLMSTR.SQC41)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC41.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC42 *****'
RELOAD PACKAGE (SQLMSTR.SQC42)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC42.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC43 *****'
RELOAD PACKAGE (SQLMSTR.SQC43)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC43.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC44 *****'
RELOAD PACKAGE (SQLMSTR.SQC44)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC44.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC45 *****'
RELOAD PACKAGE (SQLMSTR.SQC45)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC45.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC46 *****'
RELOAD PACKAGE (SQLMSTR.SQC46)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC46.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC47 *****'
RELOAD PACKAGE (SQLMSTR.SQC47)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC47.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC48 *****'
RELOAD PACKAGE (SQLMSTR.SQC48)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC48.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC49 *****'
RELOAD PACKAGE (SQLMSTR.SQC49)
     REPLACE
     KEEP
```

Figure 74 (Part 4 of 5). Load the Control Center Packages, SQMRLDPK.Z

```
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC49.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC50 *****'
RELOAD PACKAGE (SQLMSTR.SQC50)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC50.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC51 *****'
RELOAD PACKAGE (SQLMSTR.SQC51)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC51.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC52 *****'
RELOAD PACKAGE (SQLMSTR.SQC52)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC52.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC53 *****'
RELOAD PACKAGE (SQLMSTR.SQC53)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC53.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC54 *****'
RELOAD PACKAGE (SQLMSTR.SQC54)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC54.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC60 *****'
RELOAD PACKAGE (SQLMSTR.SQC60)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC60.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC61 *****'
RELOAD PACKAGE (SQLMSTR.SQC61)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC61.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC62 *****'
RELOAD PACKAGE (SQLMSTR.SQC62)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC62.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC64 *****'
RELOAD PACKAGE (SQLMSTR.SQC64)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC64.Q (NOCONT
COMMENT '***** RELOAD PACKAGE SQLMSTR.SQC65 *****'
RELOAD PACKAGE (SQLMSTR.SQC65)
     REPLACE
     KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER SQC65.Q (NOCONT
#*
#&
$ $$ EOJ
```

Figure 74 (Part 5 of 5). Load the Control Center Packages, SQMRLDPK.Z

Appendix D. Control Center Tool Tables

This appendix provides examples of the Control Center tables for the Maintenance Tracking, Database Monitors, and Group Authorization tools. Change the LIBDEF card, if necessary, to point to your production database manager library.

Maintenance Tracking Table

The figure that follows is job SQMCRMNT.Z. This DBSU job creates the SQLMAINT tables to help you keep track of database maintenance activities.

```
$ $$ JOB JNM=SQMCRMNT,CLASS=0,DISP=D,PRI=9
$ $$ LST CLASS=Q
// JOB SQMCRMNT
// LIBDEF *,SEARCH=PRD2.DB2610
// EXEC ARIDBS,SIZE=AUTO,PARM='D(DB2VSE61)'
READ MEMBER SQMCONN.C NOCONT
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*                                                       *'
COMMENT '*  CREATE THE CONTROL CENTER SQLMAINT TABLE             *'
COMMENT '*                                                       *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
SET AUTOCOMMIT (ON)
SET ERRORMODE (CONTINUE)
COMMENT '* * * * * * *          Drop DBSPACE           * * * * *'
DROP DBSPACE PUBLIC.SQLMAINT;
COMMENT '* * * * * * *          Acquire DBSPACE        * * * * *'
ACQUIRE PUBLIC DBSPACE NAMED PUBLIC.SQLMAINT
(PAGES    = 128,
 PCTINDEX = 33,
 PCTFREE  = 10,
 NHEADER  = 1,
 STORPOOL = 1,
 LOCK     = PAGE);
COMMENT '* * * * * * *          Create TABLE           * * * * *'
CREATE TABLE SQLMSTR.SQLMAINT
(OWNER              CHAR(8),
 DBSPACENAME        CHAR(18),
 DBSPACENO          SMALLINT,
 FREEPCT            SMALLINT,
 UPSTAT_DATE        DATE,
 UPSTAT_TIME        TIME,
 UPSTAT_ELAPSED     TIME,
 REORG_DATE         DATE,
 REORG_TIME         TIME,
 REORG_ELAPSED      TIME,
 REORG_FREEPCT      SMALLINT,
 REORG_PCTINDX      SMALLINT,
 REORG_STATUS       CHAR(2),
 REORG_WEIGHT       SMALLINT,
 NPAGES             INTEGER)
IN PUBLIC.SQLMAINT;
COMMENT '* * * * * * *          Primary Key            * * * * *'
CREATE UNIQUE INDEX SQLMSTR.SQLMAINT_INDX1
ON SQLMSTR.SQLMAINT
(OWNER              ASC,
 DBSPACENAME        ASC)
PCTFREE = 10;
COMMENT '* * * * * * *          Update All Statistics  * * * * *'
UPDATE ALL STATISTICS FOR DBSPACE
PUBLIC.SQLMAINT;
```

Figure 75 (Part 1 of 2). Define the Control Center Maintenance Table (SQMCRMNT.Z)

```
COMMENT '* * * * * * *          Table Grants          * * * * *'
GRANT SELECT ON SQLMSTR.SQLMAINT
TO PUBLIC;
#*
#&
$ $$ EOJ
```

Figure 75 (Part 2 of 2). Define the Control Center Maintenance Table (SQMCRMNT.Z)

Database Monitor Tables

The figure that follows is the SQMCRMON.Z batch DBSU job which defines the Database Monitor tables described in Chapter 6, “Using the Database Monitors” on page 37.

```
$ $$ JOB JNM=SQMCRMON,CLASS=0,DISP=D,PRI=9
$ $$ LST CLASS=A
// JOB SQMCRMON
// LIBDEF *,SEARCH=PRD2.DB2610
// EXEC ARIDBS,SIZE=AUTO,PARM='D(DB2VSE61)'
READ MEMBER SQMCONN.C NOCONT
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*                                                       *'
COMMENT '*  CREATE THE CONTROL CENTER MONITOR TABLES             *'
COMMENT '*                                                       *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
SET AUTOCOMMIT (ON);
SET ERRORMODE CONTINUE;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*          SET AUTOMATIC UPDATE STATISTICS OFF          *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
SET UPDATE STATISTICS (OFF);
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*                     DROP DBSPACE                      *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
DROP DBSPACE PUBLIC.CC_MONITOR;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*                     ACQUIRE DBSPACE                   *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
ACQUIRE PUBLIC DBSPACE NAMED PUBLIC.CC_MONITOR
(PAGES    = 128,
 PCTINDEX = 33,
 PCTFREE  = 0,
 NHEADER  = 8,
 STORPOOL = 1,
 LOCK     = PAGE);
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*               CREATE CONTROL TABLE                    *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
CREATE TABLE SQLMSTR.MONITOR_CONTROL
(MONITOR_NO         CHAR(2)     NOT NULL,
 DBSPACE_NO         CHAR(5)     NOT NULL,
 ACTIVE_IND         CHAR(1)     NOT NULL,
 RUN_SUN_IND        CHAR(1)     NOT NULL,
 RUN_MON_IND        CHAR(1)     NOT NULL,
 RUN_TUE_IND        CHAR(1)     NOT NULL,
 RUN_WED_IND        CHAR(1)     NOT NULL,
 RUN_THU_IND        CHAR(1)     NOT NULL,
 RUN_FRI_IND        CHAR(1)     NOT NULL,
 RUN_SAT_IND        CHAR(1)     NOT NULL,
 LAST_RUN_DATE      DATE        NOT NULL,
```

Figure 76 (Part 1 of 5). Define the Control Center Monitor Tables (SQMCRMON.Z)

```
 LAST_RUN_TIME      TIME        NOT NULL,
 START_TIME         TIME        NOT NULL,
 STOP_TIME          TIME        NOT NULL,
 INTERVAL           DEC(6,0)    NOT NULL,
 RESET_DATA_IND     CHAR(1)     NOT NULL,
 RESET_DAY_NO       CHAR(1)     NOT NULL,
 PRINT_REPORT_IND   CHAR(1)     NOT NULL,
 REPORT_NAME        CHAR(8)     NOT NULL,
 CLASS              CHAR(1)     NOT NULL,
 PRI                CHAR(1)     NOT NULL,
 DISP               CHAR(1)     NOT NULL,
 SCAN_CKPT_WAIT     CHAR(1)     NOT NULL,
 SCAN_USER_WAIT     CHAR(1)     NOT NULL,
 SCAN_AGENT_NOT     CHAR(1)     NOT NULL,
 SCAN_AGNT_WAIT     CHAR(1)     NOT NULL,
 SCAN_INACT         CHAR(1)     NOT NULL,
 SCAN_LOCK_NOT      CHAR(1)     NOT NULL,
 SCAN_CKPT          CHAR(1)     NOT NULL,
 SCAN_ANY           CHAR(1)     NOT NULL,
 SCAN_PCT_USED      CHAR(2)     NOT NULL,
 DATABASE_NAME      CHAR(18)    NOT NULL,
 MONITOR_NAME       CHAR(12)    NOT NULL,
 DESCRIPTION        CHAR(50)    NOT NULL)
IN PUBLIC.CC_MONITOR;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*              CREATE CONTROL TABLE PRIMARY KEY      *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
CREATE UNIQUE INDEX SQLMSTR.MONITOR_CONTROL_PK
ON SQLMSTR.MONITOR_CONTROL
(MONITOR_NO         ASC,
 DBSPACE_NO         ASC)
PCTFREE = 10;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*              CREATE SHOW ACTIVE TABLE              *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
CREATE TABLE SQLMSTR.SHOW_ACTIVE
(DATE               DATE        NOT NULL,
 TIME               TIME        NOT NULL,
 NACTIVE            SMALLINT    NOT NULL,
 NIW                SMALLINT    NOT NULL,
 R_O                SMALLINT    NOT NULL,
 R_W                SMALLINT    NOT NULL,
 NEW                SMALLINT    NOT NULL,
 COMMUNICATION_WAIT SMALLINT    NOT NULL,
 LOCK_WAIT          SMALLINT    NOT NULL,
 CHECKPOINT_WAIT    SMALLINT    NOT NULL,
 PAGE_BUFFER_WAIT   SMALLINT    NOT NULL,
 BLOCK_BUFFER_WAIT  SMALLINT    NOT NULL,
 I_O_WAIT           SMALLINT    NOT NULL)
IN PUBLIC.CC_MONITOR;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*              CREATE SHOW ACTIVE PRIMARY KEY        *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
CREATE UNIQUE INDEX SQLMSTR.SHOW_ACTIVE_PK
ON SQLMSTR.SHOW_ACTIVE
(DATE               ASC,
 TIME               ASC)
PCTFREE = 10;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '* * *          CREATE SHOW LOCK TABLE           * * *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
```

Figure 76 (Part 2 of 5). Define the Control Center Monitor Tables (SQMCRMON.Z)

```
CREATE TABLE SQLMSTR.SHOW_LOCK
(DATE               DATE        NOT NULL,
 TIME               TIME        NOT NULL,
 NLRBS              INTEGER     NOT NULL,
 IN_USE             INTEGER     NOT NULL,
 FREE               INTEGER     NOT NULL,
 NLRBU              INTEGER     NOT NULL,
 MAX_USED_BY_LUW    INTEGER     NOT NULL,
 LOCKWAIT_DBSPACENO INTEGER     NOT NULL,
 LOCK_HOLDER        CHAR(8)     NOT NULL,
 LOCK_REQUESTER     CHAR(8)     NOT NULL,
 LOCK_HOLDER_STATUS CHAR(22)    NOT NULL)
IN PUBLIC.CC_MONITOR;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '* * *           CREATE SHOW LOCK PRIMARY KEY      * * *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
CREATE UNIQUE INDEX SQLMSTR.SHOW_LOCK_PK
ON SQLMSTR.SHOW_LOCK
(DATE               ASC,
 TIME               ASC)
PCTFREE = 10;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '* * *          CREATE SHOW DBEXTENT TABLE          * * *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
CREATE TABLE SQLMSTR.SHOW_DBEXTENT
(DATE               DATE        NOT NULL,
 TIME               TIME        NOT NULL,
 POOL               SMALLINT    NOT NULL,
 TOTAL_PAGES        INTEGER     NOT NULL,
 USED_PAGES         INTEGER     NOT NULL,
 FREE_PAGES         INTEGER     NOT NULL,
 RESERVE_PAGES      INTEGER     NOT NULL,
 PCT_USED           SMALLINT    NOT NULL,
 TOTAL_EXTENTS      INTEGER     NOT NULL,
 SOS                CHAR(1)     NOT NULL)
IN PUBLIC.CC_MONITOR;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '* * *          CREATE SHOW DBEXTENT PRIMARY KEY    * * *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
CREATE UNIQUE INDEX SQLMSTR.SHOW_DBEXTENT_PK
ON SQLMSTR.SHOW_DBEXTENT
(DATE               ASC,
 TIME               ASC,
 POOL               ASC)
PCTFREE = 10;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '* * *          CREATE SHOW LOG TABLE               * * *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
CREATE TABLE SQLMSTR.SHOW_LOG
(DATE               DATE        NOT NULL,
 TIME               TIME        NOT NULL,
 PCT_USED           SMALLINT    NOT NULL,
 PCT_BEFORE_ARCHIVE SMALLINT    NOT NULL,
 PCT_BEFORE_OVERFLO SMALLINT    NOT NULL,
 PAGES_BEFORE_CKPT  INTEGER     NOT NULL,
 AGENTS_BEFORE_CKPT SMALLINT    NOT NULL,
 ARCHIVE_STATUS     CHAR(8)     NOT NULL)
IN PUBLIC.CC_MONITOR;
```

Figure 76 (Part 3 of 5). Define the Control Center Monitor Tables (SQMCRMON.Z)

```
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '* * *          CREATE SHOW LOG PRIMARY KEY        * * *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
CREATE UNIQUE INDEX SQLMSTR.SHOW_LOG_PK
ON SQLMSTR.SHOW_LOG
(DATE               ASC,
 TIME               ASC)
PCTFREE = 10;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '* * *          CREATE SHOW CONNECT TABLE          * * *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
CREATE TABLE SQLMSTR.SHOW_CONNECT
(DATE               DATE        NOT NULL,
 TIME               TIME        NOT NULL,
 USERS_CONNECTED    SMALLINT    NOT NULL,
 USERS_ACTIVE       SMALLINT    NOT NULL,
 USERS_WAITING      SMALLINT    NOT NULL,
 USERS_INACTIVE     SMALLINT    NOT NULL,
 AGENTS_AVAILABLE   SMALLINT    NOT NULL,
 CONNECTIONS_AVAIL  SMALLINT    NOT NULL)
IN PUBLIC.CC_MONITOR;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '* * *          CREATE SHOW CONNECT PRIMARY KEY    * * *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
CREATE UNIQUE INDEX SQLMSTR.SHOW_CONNECT_PK
ON SQLMSTR.SHOW_CONNECT
(DATE               ASC,
 TIME               ASC)
PCTFREE = 10;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '* * *          CREATE SHOW DBSPACE TABLE          * * *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
CREATE TABLE SQLMSTR.SHOW_DBSPACE
(DATE               DATE        NOT NULL,
 TIME               TIME        NOT NULL,
 DBSPACE_NO         CHAR(5)     NOT NULL,
 TOTAL_HPAGES       INTEGER     NOT NULL,
 USED_HPAGES        INTEGER     NOT NULL,
 PCT_USED_HPAGES    SMALLINT    NOT NULL,
 PCT_FREE_HPAGES    SMALLINT    NOT NULL,
 EMPTY_HPAGES       INTEGER     NOT NULL,
 TOTAL_DPAGES       INTEGER     NOT NULL,
 USED_DPAGES        INTEGER     NOT NULL,
 PCT_USED_DPAGES    SMALLINT    NOT NULL,
 PCT_FREE_DPAGES    SMALLINT    NOT NULL,
 EMPTY_DPAGES       INTEGER     NOT NULL,
 TOTAL_IPAGES       INTEGER     NOT NULL,
 USED_IPAGES        INTEGER     NOT NULL,
 PCT_USED_IPAGES    SMALLINT    NOT NULL,
 PCT_FREE_IPAGES    SMALLINT    NOT NULL,
 EMPTY_IPAGES       INTEGER     NOT NULL)
IN PUBLIC.CC_MONITOR;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '* * *          CREATE SHOW DBSPACE PRIMARY KEY   * * *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
```

Figure 76 (Part 4 of 5). Define the Control Center Monitor Tables (SQMCRMON.Z)

```
CREATE UNIQUE INDEX SQLMSTR.SHOW_DBSPACE_PK
ON SQLMSTR.SHOW_DBSPACE
(DATE               ASC,
 TIME               ASC,
 DBSPACE_NO         ASC)
PCTFREE = 10;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '* * *             CREATE COUNTER TABLE            * * *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
CREATE TABLE SQLMSTR.COUNTER
(DATE               DATE        NOT NULL,
 TIME               TIME        NOT NULL,
 RDSCALL            INTEGER     NOT NULL,
 DBSSCALL           INTEGER     NOT NULL,
 BEGINLUW           INTEGER     NOT NULL,
 ROLL_BACK          INTEGER     NOT NULL,
 CHKPOINT           INTEGER     NOT NULL,
 LOCKLMT            INTEGER     NOT NULL,
 ESCALATE           INTEGER     NOT NULL,
 WAITLOCK           INTEGER     NOT NULL,
 DEADLCK            INTEGER     NOT NULL,
 LPAGBUFF           INTEGER     NOT NULL,
 PAGEREAD           INTEGER     NOT NULL,
 PAGWRITE           INTEGER     NOT NULL,
 LDIRBUFF           INTEGER     NOT NULL,
 DIRREAD            INTEGER     NOT NULL,
 DIRWRITE           INTEGER     NOT NULL,
 LOGREAD            INTEGER     NOT NULL,
 LOGWRITE           INTEGER     NOT NULL,
 DASDREAD           INTEGER     NOT NULL,
 DASDWRIT           INTEGER     NOT NULL,
 DASDIO             INTEGER     NOT NULL)
IN PUBLIC.CC_MONITOR;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '* * *             CREATE COUNTER PRIMARY KEY      * * *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
CREATE UNIQUE INDEX SQLMSTR.COUNTER_PK
ON SQLMSTR.COUNTER
(DATE               ASC,
 TIME               ASC)
PCTFREE = 10;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*                   UPDATE ALL STATISTICS           *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
UPDATE ALL STATISTICS FOR DBSPACE
PUBLIC.CC_MONITOR;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*                     TABLE GRANTS                  *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
GRANT SELECT ON SQLMSTR.MONITOR_CONTROL
TO PUBLIC;
GRANT SELECT ON SQLMSTR.SHOW_ACTIVE
TO PUBLIC;
GRANT SELECT ON SQLMSTR.SHOW_LOCK
TO PUBLIC;
GRANT SELECT ON SQLMSTR.SHOW_DBEXTENT
TO PUBLIC;
GRANT SELECT ON SQLMSTR.SHOW_LOG
TO PUBLIC;
GRANT SELECT ON SQLMSTR.SHOW_CONNECT
TO PUBLIC;
GRANT SELECT ON SQLMSTR.SHOW_DBSPACE
TO PUBLIC;
GRANT SELECT ON SQLMSTR.COUNTER
TO PUBLIC;
#*
#&
$ $$ EOJ
```

Figure 76 (Part 5 of 5). Define the Control Center Monitor Tables (SQMCRMON.Z)

Group Authorization Tables

The figure below shows the SQMCRGRP.Z batch DBSU job that defines the Group Authorization tables described in Chapter 7, “Group Authorization Tool” on page 49.

```
$ $$ JOB JNM=SQMCRGRP,CLASS=0,DISP=D,PRI=9
$ $$ LST CLASS=A
// JOB SQMCRGRP
// LIBDEF *,SEARCH=PRD2.DB2610
// EXEC ARIDBS,SIZE=AUTO,PARM='D(DB2VSE61)'
READ MEMBER SQMCONN.C NOCONT
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*                                                         *'
COMMENT '* CREATE THE CONTROL CENTER GROUP AUTHORIZATION TABLES*'
COMMENT '*                                                         *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
SET AUTOCOMMIT (ON);
SET ERRORMODE CONTINUE;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*          SET AUTOMATIC UPDATE STATISTICS OFF            *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
SET UPDATE STATISTICS (OFF);
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*                    DROP DBSPACE                         *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
DROP DBSPACE PUBLIC.ADMGROUP;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*                    ACQUIRE DBSPACE                      *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
```

Figure 77 (Part 1 of 3). Define the Control Center Group Authorization Tables (SQMCRGRP.Z)

```
ACQUIRE PUBLIC DBSPACE NAMED PUBLIC.ADMGROUP
(PAGES    = 256,
 PCTINDEX = 10,
 PCTFREE  = 10,
 NHEADER  = 1,
 STORPOOL = 1,
 LOCK     = PAGE);
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*              CREATE TABLES                            *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
CREATE TABLE "SQLMSTR"."APPL_GROUP_TAB" (
  "APPL_GROUP_NAME" CHAR(8)    NOT NULL
 ,"APPL_GROUP_ID" SMALLINT     NOT NULL
 ,"APPL_TYPE" CHAR(1)          NOT NULL
 ,"APPL_DESC" VARCHAR(50)      NOT NULL
 ) IN "PUBLIC"."ADMGROUP"           ;
CREATE TABLE "SQLMSTR"."GROUP_AUTH_TAB" (
  "USERID_GROUP_ID" SMALLINT   NOT NULL
 ,"APPL_GROUP_ID" SMALLINT     NOT NULL
 ,"S_AUTH" CHAR(1)             NOT NULL
 ,"I_AUTH" CHAR(1)             NOT NULL
 ,"D_AUTH" CHAR(1)             NOT NULL
 ,"U_AUTH" CHAR(1)             NOT NULL
 ,"A_AUTH" CHAR(1)             NOT NULL
 ,"E_AUTH" CHAR(1)             NOT NULL
 ) IN "PUBLIC"."ADMGROUP"           ;
CREATE TABLE "SQLMSTR"."OBJECT_TAB" (
  "OBJECT_OWNER" CHAR(8)       NOT NULL
 ,"OBJECT_NAME" CHAR(18)       NOT NULL
 ,"APPL_GROUP_ID" SMALLINT     NOT NULL
 ) IN "PUBLIC"."ADMGROUP"           ;
CREATE TABLE "SQLMSTR"."USERID_GROUP_TAB" (
  "GROUP_NAME" CHAR(8)         NOT NULL
 ,"GROUP_ID" SMALLINT          NOT NULL
 ,"GROUP_STATUS" CHAR(1)       NOT NULL
 ,"GROUP_DESC" CHAR(50)        NOT NULL
 ) IN "PUBLIC"."ADMGROUP"           ;
CREATE TABLE "SQLMSTR"."USERID_TAB" (
  "USERID" CHAR(8)             NOT NULL
 ,"GROUP_ID" SMALLINT          NOT NULL
 ) IN "PUBLIC"."ADMGROUP"           ;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*              CREATE INDEXES                           *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
CREATE UNIQUE INDEX "SQLMSTR"."APIX1"
 ON "SQLMSTR"."APPL_GROUP_TAB"
 ("APPL_GROUP_NAME" )
 PCTFREE =  10;
CREATE UNIQUE INDEX "SQLMSTR"."APIX2"
 ON "SQLMSTR"."APPL_GROUP_TAB"
 ("APPL_GROUP_ID" )
 PCTFREE =  10;
CREATE INDEX "SQLMSTR"."GAX1"
 ON "SQLMSTR"."GROUP_AUTH_TAB"
 ("USERID_GROUP_ID" )
 PCTFREE =  10;
CREATE INDEX "SQLMSTR"."GAX2"
 ON "SQLMSTR"."GROUP_AUTH_TAB"
 ("APPL_GROUP_ID" )
 PCTFREE =  10;
CREATE UNIQUE INDEX "SQLMSTR"."OIX1"
 ON "SQLMSTR"."OBJECT_TAB"
 ("APPL_GROUP_ID" ,
  "OBJECT_OWNER" ,
  "OBJECT_NAME" )
 PCTFREE =  10;
```

Figure 77 (Part 2 of 3). Define the Control Center Group Authorization Tables (SQMCRGRP.Z)

```
CREATE UNIQUE INDEX "SQLMSTR"."IX1"
 ON "SQLMSTR"."USERID_GROUP_TAB"
 ("GROUP_NAME" )
 PCTFREE =  10;
CREATE UNIQUE INDEX "SQLMSTR"."IX2"
 ON "SQLMSTR"."USERID_GROUP_TAB"
 ("GROUP_ID" )
 PCTFREE =  10;
CREATE UNIQUE INDEX "SQLMSTR"."UIX1"
 ON "SQLMSTR"."USERID_TAB"
 ("GROUP_ID" ,
  "USERID" )
 PCTFREE =  10;
CREATE INDEX "SQLMSTR"."UIX2"
 ON "SQLMSTR"."USERID_TAB"
 ("USERID" )
 PCTFREE =  10;
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
COMMENT '*                    UPDATE ALL STATISTICS            *'
COMMENT '* * * * * * * * * * * * * * * * * * * * * * * * * * * * *'
UPDATE ALL STATISTICS FOR DBSPACE
PUBLIC.ADMGROUP;
#*
#&
$ $$ EOJ
```

Figure 77 (Part 3 of 3). Define the Control Center Group Authorization Tables (SQMCRGRP.Z)

Appendix E. DBSPACE Reorganization Tool Related Files

A DBSPACE reorganization job involves several files. Depending on the options chosen, some or all of the following files are used:

Filename	File ID & Description
SQMPARM	"SQLMSTR.REORG.PARMS" This is the DBSPACE Reorganization parameter file. The job submission programs write a record to this file whenever a job is submitted. The SQMPARM record contains REORG parameters and is read by the batch DDL generation and timekeeping programs. SQMPARM is a VSAM KSDS file whose key is composed of database name, owner, DBSPACE name, and option. SQMPARM is defined during installation and resides on the Control Center user catalog.
SQMRDAT	"SQLMSTR.REORG.DATA" This file holds statistical data relating to a DBSPACE unload/reload. At job end, the SQLMAINT table is updated with this data. SQMRDAT is used for Options **2**, **3**, and **4** of the reorganization tool. It is a VSAM KSDS file whose key is composed of database name, owner, and DBSPACENAME. The file is defined during installation and resides on the Control Center user catalog.
SQMDDL	File ID built dynamically during job submission. SQMDDL is a VSAM-managed SAM file that is used to contain the DDL created by SQB01, the batch DDL generation program. The file id is composed of a concatenation of: 1. "L" - (indicates DDL) 2. Database name 3. DBSPACE owner name 4. DBSPACE name File size is defined by the DDL STMTS parameter on the DBSPACE REORGANIZATION screen (defaults to 1000 80-byte records). This file resides on the Control Center user catalog and remains there until it is deleted by a successful REORGANIZE DBSPACE or deleted specifically by the user.
SQMDATn	"SQLMSTR.DATA.FILEn" This SAM file is used to hold the output of the UNLOAD DBSPACE step when DISK is selected as the unload media. "n" corresponds to the FILE # parameter that appears on the DBSPACE REORGANIZATION UTILITY screen. Valid values are 1 to 3. This file is defined from the WORK FILE LABEL DEFINITION facility (Option 4 of the Main Menu). It is used in Options 2 and 3 of the reorganization tool.
SQMDDLn	"SQLMSTR.DDL.FILEn" This SAM file is required to contain the DDL extracted by SQB01, the batch DDL generator. It is used as the DBSU command input

file in the RELOAD step. Because DBSU expects commands to come from SYSIPT and a VSAM file cannot be assigned to SYSIPT, the VSAM SQMDDL file is REPRO'd to SQMDDLn. "n" corresponds to the FILE # parameter on the DBSPACE REORGANIZATION UTILITY screen. Valid values are 1 to 3. This file is defined from the WORK FILE LABEL DEFINITION facility and is used in Options 3 and 4 of the reorganization tool.

SQMDAT — Built dynamically by the submit program.

SQMDAT is a VSAM-managed SAM file that is used to contain the DBSPACE data unloaded from DBSU in the UNLOAD DBSPACE step. The file id is composed of a concatenation of:

1. "D" - (indicates data)
2. Database name
3. DBSPACE owner name
4. DBSPACE name

File size is computed from DBSPACE catalog information. This file is only created when a user selects UNLOAD DBSPACE (Option 2) and does not specify tape. This file is also created if an error occurs during the RELOAD step of a REORGANIZE DBSPACE from disk (Option 3). In effect, this file is a backup copy of the unloaded DBSPACE. It can be used as input to a RELOAD DBSPACE (Option 4). SQMDAT resides on the Control Center user catalog and remains there until the same DBSPACE is unloaded again or the file is specifically deleted.

SQMTAPE — Specified by the user.

SQMTAPE is used when the user selects tape media by entering a TAPE FILE NAME on the DBSPACE REORGANIZATION UTILITY screen. This file must be defined to Control Center from the WORK FILE LABEL DEFINITION facility (Option 4 of the Main Menu).

SQMMESG — "SQLMSTR.MESSAGES"

SQMMESG is a VSAM KSDS file that is used to contain Control Center error message text. The key is a 4-digit number. SQMMESG is defined during installation and resides on the Control Center user catalog.

SQMWORK — "SQLMSTR.WORK.FILES"

SQMWORK is a VSAM KSDS file that holds the ASSGN, DLBL, EXTENT, and TLBL statements that define your Control Center work files. The key is composed of a 17-character file ID and a sequence number. SQMWORK is defined during installation and resides on the Control Center user catalog.

Figure 78 on page 175 is an example of the SQMDDL file created when a REORGANIZE DBSPACE job executes for the PUBLIC.SAMPLE DBSPACE. This SQMDDL command file contains all the 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 SQMDDL file may contain considerably more commands.

```
COMMENT '**************** CONTROL CENTER DBSPACE REORG **************
COMMENT '*    Database: DB2VSE61                                    *'
COMMENT '*     DBSPACE: "PUBLIC"."SAMPLE"                           *'
COMMENT '*        Date: 12/13/96 09:21:58                           *'
COMMENT '*********************************************************'
CONNECT "SQLMSTR" IDENTIFIED BY ********;
COMMIT WORK;
GRANT DBA TO "SQLREORG" IDENTIFIED BY "PU24L5AR";
CONNECT "SQLREORG" IDENTIFIED BY "PU24L5AR";
COMMIT WORK;
COMMENT '*************            Drop DBSPACE            *************'
SET ERRORMODE (CONTINUE)
DROP DBSPACE "PUBLIC"."SAMPLE"              ;
COMMIT WORK;
SET ERRORMODE (OFF)
SET AUTOCOMMIT (ON)
COMMENT '*************          Acquire DBSPACE           *************'
ACQUIRE PUBLIC DBSPACE NAMED "PUBLIC"."SAMPLE"
 (PAGES =     512, PCTINDEX = 33,
  PCTFREE = 10, NHEADER = 8,
  STORPOOL =   1, LOCK = PAGE   );
COMMENT '*************           Create Tables            *************'
CREATE TABLE "SQLDBA"."ACTIVITY" (
  "ACTNO" SMALLINT         NOT NULL
 ,"ACTKWD" CHAR(6)          NOT NULL
 ,"ACTDESC" VARCHAR(20)      NOT NULL
 ) IN "PUBLIC"."SAMPLE"             ;
CREATE TABLE "SQLDBA"."DEPARTMENT" (
  "DEPTNO" CHAR(3)          NOT NULL
 ,"DEPTNAME" VARCHAR(36)      NOT NULL
 ,"MGRNO" CHAR(6)
 ,"ADMRDEPT" CHAR(3)          NOT NULL
 ) IN "PUBLIC"."SAMPLE"              ;
CREATE TABLE "SQLDBA"."EMP_ACT" (
  "EMPNO" CHAR(6)          NOT NULL
 ,"PROJNO" CHAR(6)          NOT NULL
 ,"ACTNO" SMALLINT         NOT NULL
 ,"EMPTIME" DECIMAL(5,2)
 ,"EMSTDATE" DATE
 ,"EMENDATE" DATE
 ) IN "PUBLIC"."SAMPLE"              ;
CREATE TABLE "SQLDBA"."EMPLOYEE" (
  "EMPNO" CHAR(6)          NOT NULL
 ,"FIRSTNME" VARCHAR(12)      NOT NULL
 ,"MIDINIT" CHAR(1)          NOT NULL
 ,"LASTNAME" VARCHAR(15)      NOT NULL
 ,"WORKDEPT" CHAR(3)
 ,"PHONENO" CHAR(4)
 ,"HIREDATE" DATE
 ,"JOB" CHAR(8)
 ,"EDLEVEL" SMALLINT         NOT NULL
 ,"SEX" CHAR(1)
 ,"BIRTHDATE" DATE
 ,"SALARY" DECIMAL(9,2)
 ,"BONUS" DECIMAL(9,2)
 ,"COMM" DECIMAL(9,2)
 ) IN "PUBLIC"."SAMPLE"              ;
CREATE TABLE "SQLDBA"."PROJ_ACT" (
  "PROJNO" CHAR(6)          NOT NULL
 ,"ACTNO" SMALLINT         NOT NULL
 ,"ACSTAFF" DECIMAL(5,2)
 ,"ACSTDATE" DATE             NOT NULL
```

Figure 78 (Part 1 of 4). Example SQLDBSU Command File

```
 ,"ACENDATE" DATE
 ) IN "PUBLIC"."SAMPLE"              ;
CREATE TABLE "SQLDBA"."PROJECT" (
  "PROJNO" CHAR(6)          NOT NULL
 ,"PROJNAME" VARCHAR(24)      NOT NULL
 ,"DEPTNO" CHAR(3)
 ,"RESPEMP" CHAR(6)
 ,"PRSTAFF" DECIMAL(5,2)
 ,"PRSTDATE" DATE
 ,"PRENDATE" DATE
 ,"MAJPROJ" CHAR(6)
 ) IN "PUBLIC"."SAMPLE"              ;
COMMENT '*************      Deactivated Primary Keys      *************'
COMMENT '*************    Inactive Unique Constraints     *************'
COMMENT '*************          Reload Tables             *************'
RELOAD TABLE("SQLDBA"."ACTIVITY")
 PURGE
 INTABLE("SQLDBA"."ACTIVITY")
 INFILE(DBSFILE);
RELOAD TABLE("SQLDBA"."DEPARTMENT")
 PURGE
 INTABLE("SQLDBA"."DEPARTMENT")
 INFILE(DBSFILE);
RELOAD TABLE("SQLDBA"."EMP_ACT")
 PURGE
 INTABLE("SQLDBA"."EMP_ACT")
 INFILE(DBSFILE);
RELOAD TABLE("SQLDBA"."EMPLOYEE")
 PURGE
 INTABLE("SQLDBA"."EMPLOYEE")
 INFILE(DBSFILE);
RELOAD TABLE("SQLDBA"."PROJ_ACT")
 PURGE
 INTABLE("SQLDBA"."PROJ_ACT")
 INFILE(DBSFILE);
RELOAD TABLE("SQLDBA"."PROJECT")
 PURGE
 INTABLE("SQLDBA"."PROJECT")
 INFILE(DBSFILE);
ALTER DBSPACE "PUBLIC"."SAMPLE"               (PCTFREE =    0);
COMMENT '*************           Comment Tables           *************'
COMMENT '*************          Comment Columns           *************'
COMMENT '*************        Primary Cluster Keys        *************'
ALTER TABLE "SQLDBA"."ACTIVITY"
 ADD PRIMARY KEY
 ("ACTNO" )
 PCTFREE =  10;
ALTER TABLE "SQLDBA"."DEPARTMENT"
 ADD PRIMARY KEY
 ("DEPTNO" )
 PCTFREE =  10;
ALTER TABLE "SQLDBA"."EMPLOYEE"
 ADD PRIMARY KEY
 ("EMPNO" )
 PCTFREE =  10;
ALTER TABLE "SQLDBA"."PROJ_ACT"
```

Figure 78 (Part 2 of 4). Example SQLDBSU Command File

```
 ADD PRIMARY KEY
 ("PROJNO" ,
  "ACTNO" ,
  "ACSTDATE" )
 PCTFREE =  10;
ALTER TABLE "SQLDBA"."PROJECT"
 ADD PRIMARY KEY
 ("PROJNO" )
 PCTFREE =  10;
COMMENT '*************  Clustering Unique Constraints     *************'
COMMENT '*************           Create Indexes           *************'
CREATE INDEX "SQLDBA"."PROJNOIN"
 ON "SQLDBA"."EMP_ACT"
 ("PROJNO" )
 PCTFREE =  10;
CREATE INDEX "SQLDBA"."DEPTNOI"
 ON "SQLDBA"."PROJECT"
 ("DEPTNO" )
 PCTFREE =  10;
CREATE INDEX "SQLDBA"."EMPNOIN"
 ON "SQLDBA"."EMP_ACT"
 ("EMPNO" )
 PCTFREE =  10;
CREATE INDEX "SQLDBA"."MGRNOI"
 ON "SQLDBA"."DEPARTMENT"
 ("MGRNO" )
 PCTFREE =  10;
CREATE INDEX "SQLDBA"."RESPEMPI"
 ON "SQLDBA"."PROJECT"
 ("RESPEMP" )
 PCTFREE =  10;
CREATE INDEX "SQLDBA"."WORKDEPTI"
 ON "SQLDBA"."EMPLOYEE"
 ("WORKDEPT" )
 PCTFREE =  10;
COMMENT '*************           Primary Keys             *************'
COMMENT '*************       Remaining Foreign Keys       *************'
ALTER TABLE "SQLDBA"."PROJ_ACT"
  ADD FOREIGN KEY "R_ACTIV"
 ("ACTNO"
 ) REFERENCES "SQLDBA"."ACTIVITY"
  ON DELETE RESTRICT;
ALTER TABLE "SQLDBA"."EMPLOYEE"
  ADD FOREIGN KEY "R_DEPT1"
 ("WORKDEPT"
 ) REFERENCES "SQLDBA"."DEPARTMENT"
  ON DELETE SET NULL;
ALTER TABLE "SQLDBA"."PROJECT"
  ADD FOREIGN KEY "R_DEPT2"
 ("DEPTNO"
 ) REFERENCES "SQLDBA"."DEPARTMENT"
  ON DELETE RESTRICT;
ALTER TABLE "SQLDBA"."DEPARTMENT"
  ADD FOREIGN KEY "R_EMPLY1"
 ("MGRNO"
 ) REFERENCES "SQLDBA"."EMPLOYEE"
  ON DELETE SET NULL;
```

Figure 78 (Part 3 of 4). Example SQLDBSU Command File

```
ALTER TABLE "SQLDBA"."EMP_ACT"
  ADD FOREIGN KEY "R_EMPLY3"
 ("EMPNO"
 ) REFERENCES "SQLDBA"."EMPLOYEE"
  ON DELETE CASCADE;
ALTER TABLE "SQLDBA"."PROJECT"
  ADD FOREIGN KEY "R_EMPLY2"
 ("RESPEMP"
 ) REFERENCES "SQLDBA"."EMPLOYEE"
  ON DELETE SET NULL;
ALTER TABLE "SQLDBA"."EMP_ACT"
  ADD FOREIGN KEY "R_PROACT"
 ("PROJNO"
 , "ACTNO"
 , "EMSTDATE"
 ) REFERENCES "SQLDBA"."PROJ_ACT"
  ON DELETE RESTRICT;
ALTER TABLE "SQLDBA"."PROJ_ACT"
  ADD FOREIGN KEY "R_PROJ2"
 ("PROJNO"
 ) REFERENCES "SQLDBA"."PROJECT"
  ON DELETE RESTRICT;
COMMENT '*************      Active Unique Constraints     *************'
COMMENT '*************           Table Grants             *************'
CONNECT "SQLDBA" IDENTIFIED BY "BOOMER";
GRANT SELECT
 ON "SQLDBA"."ACTIVITY"
 TO "PUBLIC";
GRANT SELECT
 ON "SQLDBA"."DEPARTMENT"
 TO "PUBLIC";
GRANT SELECT
 ON "SQLDBA"."EMP_ACT"
 TO "PUBLIC";
GRANT SELECT
 ON "SQLDBA"."EMPLOYEE"
 TO "PUBLIC";
GRANT SELECT
 ON "SQLDBA"."PROJ_ACT"
 TO "PUBLIC";
GRANT SELECT
 ON "SQLDBA"."PROJECT"
 TO "PUBLIC";
COMMENT '*************          Column Grants             *************'
COMMENT '*************       Views with Grants            *************'
   COMMENT '******** SQLDBA.VPHONE                 ********'
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;
GRANT SELECT
 ON "SQLDBA"."VPHONE"
 TO "PUBLIC";
   COMMENT '******** SQLDBA.VPROJ                 ********'
CREATE VIEW VPROJ (EMP_NO, AC_STAFF, DEPT_NO) AS SELECT EMPNO, ACSTAFF,
 DEPTNO FROM EMP_ACT, PROJ_ACT,
 PROJECT WHERE EMP_ACT.PROJNO = PROJ_ACT.PROJNO AND PROJ_ACT.PROJNO =
 PROJECT.PROJNO;
COMMENT '********************  PACKAGE REBIND ********************'
```

Figure 78 (Part 4 of 4). Example SQLDBSU Command File

Appendix F. Miscellaneous Members

The distribution tape also contains 5 miscellaneous members that must be installed in the production library. They are listed below along with a brief description.

1. SQMCDBA.C SQL statement to connect as SQLDBA
2. SQMGDBA.C SQL statement to grant DBA to SQLMSTR
3. SQMCONN.C SQL statement to connect as SQLMSTR
4. SQMESSGS.Z Control Center error message text
5. SQMHLPTX.Z Control Center help text

Glossary

access-path. The path used to get data specified in SQL statements. An access path can involve either an index, a sequential search, or a combination of both.

applid. The name of a CICS system as known to VTAM®.

batch. Processing that involves little or no terminal interaction.

catalog. 1. A set of tables maintained by the database manager. 2. A directory of files and libraries, with reference to their locations. 3. To store a library member such as a phase, module, or book in a sublibrary.

CEDA. The resource definition online transaction.

CICS. Customer Information Control System. IBM's teleprocessing monitor for VSE/ESA.

CICS Report Controller Feature. A set of programs and transactions that interface with VSE/POWER to help users create reports and print them at distributed locations.

CICS transaction. Computing and data-access tasks grouped together as a unit of work.

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

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

| **CSD**. CICS System Definition file.

Control Center. An IBM licensed program consisting of menus and programs to assist a DBA in the on-going administration of DB2 Server for VSE & VM databases.

database. An organized collection of stored operational data, used by the application systems of an organization.

database administrator (DBA). An individual responsible for the availability, development, design, maintenance, operation, performance, recoverability, and security of the database.

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

dbextent. The physical medium upon which data is stored. One or more dbextents comprise a storage pool.

database manager. A program product that processes SQL statements.

database switching. The facility that allows users and applications to connect from one database server to another.

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

| **DBSU**. Database Services Utility program used to run
| the DDL needed to perform a function.

DB2 (Database 2). Pertaining to the IBM licensed program that is the version of DB2 Server for the VSE and VM environments.

DB2 Optimizer. A component of a relational DBMS that carries out the logic required to find data in a database. The optimizer determines the access path.

DDL. Data Definition Language. The SQL statements for deleting and defining objects such as tables and indexes in an RDBMS.

distribution tape. A magnetic tape that contains, for example, a preconfigured operating system such as VSE/ESA. This tape is shipped to the customer for program installation.

FCT. File Control Table. This CICS control table contains entries that define files to CICS.

JCL (Job Control Language). A language that serves to prepare a job or each job step of a job to be run.

job scheduling. The process of creating the JCL necessary to run a job, then to invoke VSE/POWER to actually submit the job to the system for execution.

locking. A mechanism that prevents concurrent users from accessing the same data, at the same time. This ensures data integrity.

LIBDEF. A VSE system control statement that defines what sublibraries are to be searched for members of a specified type or the sublibrary in which new phases are to be stored.

menu. A screen that offers the user a choice of execution options.

MUM (multiple user mode). A mode of operating the DB2 database manager in which one or more users or application programs can access the database at the same time.

online processing. Processing by which the input data enters the computer directly from a display station and the output data is transmitted directly to the display station.

Online Resource Adapter. The DB2 code that provides the connection between DB2 databases and online (CICS) applications.

operator command. A statement to a control program, issued using a console or terminal

package. A control structure produced during program preparation that is used to execute SQL statements.

PCT. Program Control Table. The CICS control table that contains entries that describe transactions.

PPT. Processing Program Table. The CICS control table that contains entries that describe programs.

production library. The VSE/ESA library that contains the DB2 and Control Center code (PRD2).

program preparation. The process of producing an executable DB2 application program. The process includes precompilation, compilation, and bind.

pseudo-conversational. A method of on-line programming whereby a program is removed from storage when it is waiting for data from the terminal. A much more efficient technique than conversational programming.

RDBMS. Relational Data Base Management System.

| **RDO**. Resource Definition Online.

rebind. To recreate a package.

SAM. Sequential Access Method. Files processed without an index.

SNT. Signon Table. The CICS control table that contains an entry for each userid.

spanned records. Records that are defined to span multiple blocks.

spool file. 1. A file that contains output data saved for later processing. 2. One of three VSE/POWER files on disk: queue file, data file, and account file.

SQL. Structured Query Language. A data sub-language for defining and accessing data in an RDBMS.

static SQL. SQL statements that are embedded within a program, and are prepared during the program preparation process before the program is executed. Static SQL statements have a corresponding access plan in the database.

standard labels. Disk file labels (DLBLS) that are loaded into either the partition or system standard label areas of a VSE system and are thus available to all subsequent jobs that run in that partition or system.

storage pool. A specific set of available storage areas. These areas are used by the database administrator to control storage of the database. A storage pool contains one or more DBSPACES.

sublibrary. In VSE, a subdivision of a library. Members can only be accessed in a sublibrary.

SUM (single user mode). A mode of operation in which the DB2 database manager and one application run in the same virtual machine. No other application programs or users can access the database at the same time.

time event scheduling. In VSE/POWER, the facility to schedule jobs for processing in a partition at a predefined time once or repetitively.

TLBL. The VSE Tape LaBeL system control statement that defines a tape file to an application.

transaction. Execution of one or more programs that function together as a unit in a CICS environment.

transaction identifier. The unique four-character code that identifies a CICS transaction.

VSE (Virtual Storage Extended). A system that consists of a basic operating system and any IBM supplied and user-written programs required to meet the data processing needs of a user. VSE and the hardware it controls form a complete computing system.

VSE/DITTO (VSE/Data Interfile Transfer, Testing, and Operations). An IBM licensed program that provides file-to-file services for disk, tape, and card devices.

VSE/ESA (VSE/Enterprise Systems Architecture). The most advanced VSE system currently available.

VSE/ICCF (VSE/Interactive Computing and Control Facility). An IBM licensed program that serves as interface, on a time-slice basis, to authorized users of terminals linked to the system's processor.

VSE/ICCF library. A file composed of smaller files (libraries) including system and user data which can be accessed under the control of VSE/ICCF.

VSE/POWER. An IBM licensed program primarily used to spool input and output. The program's networking functions enable a VSE system to exchange files with or run jobs on another remote processor.

| **XCTL**. CICS Transfer Program Control Command.

Bibliography

This bibliography lists publications that are referenced in this manual or that may be helpful.

DB2 Server for VSE Publications

- *DB2 Server for VSE Application Programming*, SC09-2662
- *DB2 Server for VSE Database Administration*, SC09-2655
- *DB2 Server for VSE & VM Database Services Utility*, SC09-2663
- *DB2 Server for VSE Diagnosis Guide and Reference*, LC09-2673
- *DB2 Server for VSE & VM Overview*, GC09-2806
- *DB2 Server for VSE Installation*, GC09-2656
- *DB2 Server for VSE & VM Interactive SQL Guide and Reference*, SC09-2674
- *DB2 Server for VSE Master Index and Glossary*, SC09-2667
- *DB2 Server for VSE Messages and Codes*, GC09-2665
- *DB2 Server for VSE & VM Operation*, SC09-2668
- *DB2 Server for VSE System Administration*, SC09-2658
- *DB2 Server for VSE & VM Performance Tuning Handbook*, GC09-2669
- *DB2 Server for VSE & VM SQL Reference*, SC09-2671

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

Virtual Storage Extended/Enterprise Systems Architecture (VSE/ESA) Publications

- *IBM VSE/ESA Administration*, SC33-6505
- *IBM VSE/ESA Diagnosis Tools*, SC33-6514
- *IBM VSE/ESA General Information*, GC33-6501
- *IBM VSE/ESA Guide for Solving Problems*, SC33-6510
- *IBM VSE/ESA Guide to System Functions*, SC33-6511
- *IBM VSE/ESA Installation*, SC33-6504
- *IBM VSE/ESA Messages & Codes*, SC33-6507
- *IBM VSE/ESA Networking Support*, SC33-6508
- *IBM VSE/ESA Operation*, SC33-6506
- *IBM VSE/ESA Planning*, SC33-6503
- *IBM VSE/ESA System Control Statements*, SC33-6513
- *IBM VSE/ESA System Macros User's Guide*, SC33-6515
- *IBM VSE/ESA System Macros Reference*, SC33-6516
- *IBM VSE/ESA System Utilities*, SC33-6517
- *IBM VSE/ESA Unattended Node Support*, SC33-6512
- *IBM VSE/ESA Using IBM Workstations*, SC33-6509

CICS/VSE Publications

- *CICS/VSE Application Programming Reference*, SC33-0713
- *CICS/VSE Application Programming Guide*, SC33-0712
- *CICS Application Programming Primer (VS COBOL II)*, SC33-0674
- *CICS/VSE CICS-Supplied Transactions*, SC33-0710
- *CICS/VSE Customization Guide*, SC33-0707
- *CICS/VSE Facilities and Planning Guide*, SC33-0718
- *CICS/VSE Intercommunication Guide*, SC33-0701
- *CICS/VSE Performance Guide*, SC33-0703
- *CICS/VSE Problem Determination Guide*, SC33-0716
- *CICS/VSE Recovery and Restart Guide*, SC33-0702
- *CICS/VSE Release Guide*, GC33-0700
- *CICS/VSE Report Controller User's Guide*, SC33-0705
- *CICS/VSE Resource Definition (Macro)*, SC33-0709
- *CICS/VSE Resource Definition (Online)*, SC33-0708
- *CICS/VSE System Definition and Operations Guide*, SC33-0706
- *CICS/VSE System Programming Reference*, SC33-0711
- *CICS/VSE User's Handbook*, SX33-6079
- *CICS/VSE XRF Guide*, SC33-0704

CICS/ESA Publications

- *CICS/ESA General Information*, GC33-0155

VSE/Virtual Storage Access Method (VSE/VSAM) Publications

- *VSE/VSAM Commands and Macros*, SC33-6532
- *VSE/VSAM Introduction*, GC33-6531
- *VSE/VSAM Messages and Codes*, SC24-5146
- *VSE/VSAM Programmer's Reference*, SC33-6535
- *Using the VSE/VSAM Backup/Restore Feature*, SC24-5216
- *Using the VSE/VSAM Space Management for SAM Feature*, SC24-5192
- *Using VSE/VSAM Commands and Macros*, SC24-5144

VSE/Interactive Computing and Control Facility (VSE/ICCF) Publications

- *VSE/ICCF Administration and Operation*, SC33-6562
- *VSE/ICCF Primer*, SC33-6561
- *VSE/ICCF User's Guide*, SC33-6563

VSE/POWER Publications

- *VSE/POWER Administration and Operation*, SC33-6571
- *VSE/POWER Application Programming*, SC33-6574
- *VSE/POWER Installation and Operations Guide*, SH12-5329
- *VSE/POWER Networking*, SC33-6573
- *VSE/POWER Remote Job Entry*, SC33-6572

Distributed Relational Database Architecture (DRDA) Library

- *Application Programming Guide*, SC26-4773
- *Architecture Reference*, SC26-4651
- *Connectivity Guide*, SC26-4783
- *DRDA: Every Manager' s Guide*, GC26-3195
- *Planning for Distributed Relational Database*, SC26-4650
- *Problem Determination Guide*, SC26-4782

C/370 for VSE Publications

- *IBM C/370 General Information*, GC09-1386
- *IBM C/370 Programming Guide for VSE*, SC09-1399
- *IBM C/370 Installation and Customization Guide for VSE* GC09-1417
- *IBM C/370 Reference Summary for VSE*, SX09-1246
- *IBM C/370 Diagnosis Guide and Reference for VSE* LY09-1805

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

- *Control Center Installation and Operations Guide for VSE*, GC09-2679
- *IBM Replication Guide and Reference*, S95H-0999

Index

A

adding objects to a group 53
adding users to a group 55
Analysis
  of dbspaces 2
Application Group
  adding objects to 53
  defining 52
Architecture
  database connection 4
  invocation 5
  overview 1
  product 1

B

backup 110
  processing flow 112, 119
Batch jobs
  deleting 3
  holding 3
  releasing 3
Benefits
  automation 1
  DDL generation 1
  jobscheduling 1
  QMF 1
  reorganization 1
  statistics updating 1

C

catalog
  customization 11
  definition 10
CICS
  preparing for Control Center 12
  preparing for migration 25, 31
  report controller interface 3
  required table entries 12
  TRANSIDs 5
CICS ID 33
CICS Report Controller Interface
  a sample session 99
  highlights 3
  overview 99
CLUSTERRATIO < nnnnn 77
commitcount on reload 64
Control Center Packages 157
  loading 18, 28, 32
  rebinding 2
Controlling access 1

D

DASD
  allocating VSAM space 10
  production library requirements 8
  SAM package space 24
DATA CAPTURE 115
database ix
DATABASE DEFINITION PROC parameter 69
DATE tape file parameter 97
DBA 1
DBSPACE Analysis Tools
  additional topics 83
    initial execution 84
    reorganization work space 84
  before you begin 73
  DBSPACE Analysis Submit screen 82
  DBSPACE reorganization analysis tool 79
  highlights 2
  how they work 73
  job options 82
  overview 73
  reorg criteria 77
  the DBSPACE Analysis Utility screen 75
    functions 76
    reorg criteria 77
    selection options 77
  the SQLMAINT table 74
  update statistics analysis tool 78
DBSPACE Reorganization
  before you begin 59
  scheduling 59
  single user mode 69
  when to reorg 59
DBSPACE REORGANIZATION Screen 62
  optional parameters 63, 113
  options 62, 113
DBSPACE Reorganization Tool
  about the tool 59
  benefits 59
  failure restart 70
  features 60
  highlights 2
  how it works 61
    processing flow 61
    related files 173
  problem analysis 71
  scheduling 70
  single user mode reorganization 69
    before you choose it 69
    parameters 69

DBSPACE Reorganization Tool *(continued)*
special considerations 70
failure restart 70
problem analysis 71
scheduling 70
submit screen 66
optional parameters 67
unloading to tape 70
use 65, 115
generate DDL 65
reload DBSPACE 66
reorganize DBSPACE 66
unload DBSPACE 66
DBSPACES
analyzing 2
DDL to recreate 2
listing reorganization candidates 76
listing update statistics candidates 76
reloading 2
reorganizing 2, 59
unloading 2
DDL 1
DDL STMTS 65
DEVICE CLASS 97

E

error message loading 14, 26, 31

F

FILE # 83
FILE SEQUENCE NUMBER 97
function keys 33

G

Generate DDL 62, 109, 115
GENERATION NUMBER 97
granting
authority to a user group 56
DBA authority to SQLMSTR 15
Group Authorization Tool
adding objects to a group 53
adding users to a group 55
defining a user group 54
defining an application group 52
granting authority to a user group 56
highlights 2

H

Help Facility
highlights 3
overview 103
TRANSID 5

I

Initializing the SQLMAINT Control Table 76
Install tape 7
scanning 7
Installation
allocating VSAM space 10
before you begin... 5
CICS table entries required 12
defining the SQLMAINT table 16
granting SQLMSTR DBA authority 15
help for 7
IBM supplied aids 7
initial work file label definition 19
jcl to scan the distribution tape 8
jobstream 9
library space required 8, 13
loading error messages 14
loading standard labels 11
loading the database packages 18
loading the SQMHELP table 16
machine-readable material 7
materials needed for 5
overview 5
placement of code 5
preparing CICS for 12
prerequisite programs 4
preventive service planning 6
production library 5
steps 7
tape 7
VSAM definition 10
Installation JCL 127
Interfaces
CICS report controller 3
operator command 2
Invocation
using panels 5
using TRANSID 5

J

Job Scheduling
highlights 3
parameters 67
repetitive DBSPACE reorganizations 70

L

LIBDEF
changing 14, 15, 16, 17, 18, 26, 27, 28, 31, 32
Library
specifying for installation 9
Listing DBSPACE Reorganization Candidates 76
Listing Update Statistics Candidates 76

LOGFULL condition 71
LOGMODE parameter 69

M

Main Menu 33
migration 23
allocating SAM space 24
before you begin... 23
jobstream 23, 30
library space required 24
loading error messages 26, 31
loading the database packages 28, 32
loading the SQMHELP table 26
loading the SQMHELPtable 31
package work file label definition 28
preparing CICS for 25, 31
steps 23
Miscellaneous Members 179
MODE 97
Monitor Utilities
highlights 3
Moving dbspaces 2
Multiple database support 2

N

NDIRBUF parameter 69
NOVERFLOW ROWS > n% 77
NPAGBUF parameter 69
NUMBER OF TRACKS/BLOCKS 95

O

Operation
function keys 33
getting started 33
operator command tool 35
Operator Commands
COUNTER 35
highlights 2
menu 35
SHOW 35
Overview
of architecture 1
of installation 5
product 1

P

Package Utilities
highlights 3
package migration 87
rebind package 86
reload package 86
unload package 86
view package 87
Parameters
database name 15, 16, 17, 18, 26, 27, 28, 31, 32
DAYS SINCE 77
DBSPACE analysis 77
DBSPACE analysis jobs 82
DBSPACE migration 63
DBSPACE reorganization 63
disk work file label definition 95
FILE # 62, 113
job scheduling 67
rebinding 64, 83, 114
REORG CRITERIA 77
single user mode 1, 69
STORPOOL 16, 17, 26, 27, 31
TABLE migration 113
TABLE reorganization 113
tape work file label definition 97
TLBL 3
WITH REFRESH 76
Product
benefits 1
description 4
software requirements 3
Production library 5

Q

QMF 3, 33, 34

R

Rebind Package 64, 114
REBIND PACKAGES 83
RELATIVE TRACK/BLOCK 95
Reload DBSPACE 63
reload TABLE 116
Reorg Criteria 77
REORG_DATE 75
REORG_ELAPSED 75
REORG_FREEPCT 75
REORG_PCTINDX 75
REORG_STATUS 75
REORG_TIME 75
REORG_WEIGHT 75
Reorganize DBSPACE 63
reorganize table 116
Requirements
package library space 13, 24
program 4
software 3

S

Scheduling
DBSPACE analysis jobs 82
DBSPACE reorganizations 59

Scheduling *(continued)*
UPDATE STATISTICS jobs 82
Scheduling jobs 3
Single User Mode
before you choose it 69
DBSPACE reorganization 69
reorganization parameters 69
startupparameters 1
Software requirements 3
SQL Master for VSE
and DB2 4
highlights 1
SQL/DS
and Control Center for VSE 4
SQLMAINT 127
defining 16
description 74
initializing 76
updated during reorgs 61
SQLMSTR ID 15
SQLMSTR.DATA.FILE2 94
SQLMSTR.MESSAGES 10
SQLMSTR.REORG.DATA 10
SQLMSTR.REORG.PARMS 10
SQLMSTR.TABLE.PARMS 10
SQLMSTR.USER.CATALOG 10
SQLMSTR.WORK.FILES 10, 11
SQMCAT 10, 11
customization 11
definition 10
SQMCDBA 15
SQMCDBA.C 15, 179
SQMCONN.C 179
SQMCRGRP 169
SQMCRGRP.Z 18, 27, 127, 169
SQMCRHLP 135
SQMCRHLP.Z 16, 26, 31, 127
SQMCRMNT 163
SQMCRMNT.Z 16, 17, 127, 163
SQMCRMON 17, 27, 164
SQMCRMON.Z 17, 27, 127, 164
SQMCSDUP.Z 12, 25, 127, 130
SQMDAT 173, 174
SQMDATx 19
SQMDDL 173, 174, 175
definition 173
SQMDDLn 173
SQMDDLx 19
SQMDELET.Z 14, 127
SQMESSGS 14, 15, 26, 31
SQMESSGS.Z 14, 179
SQMFCT.A 12, 25, 127, 133
SQMGDBA 15
SQMGDBA.C 15, 179
SQMGRANT 15
SQMGRANT.Z 15, 16, 127
SQMHELP
defining 16, 26, 31
loading 16, 26, 31
SQMHLPTX 16, 26, 31
SQMHLPTX.Z 26, 179
SQMLDMSG 14, 26, 31
SQMLDMSG.Z 14, 15, 26, 127
SQMLIBDF 13
SQMLIBDF.Z 13, 24, 127, 134
SQMMESG 31, 174
SQMMESG.Z 26
SQMMSHPI 9, 10, 23, 30
SQMMSHPI.Z 127
SQMPARM 70, 173
definition 173
SQMRDAT 173
definition 173
SQMRENAM.Z 14, 127
SQMRLDPK 18, 28, 158
SQMRLDPK.Z 18, 28, 32, 127, 157, 158
SQMSTD 11, 12
SQMSTD.Z 127
SQMTAPE 174
SQMTSCAN 8
SQMTSCAN.Z 127
SQMVRHLP.Z 32
SQMVSAM 10, 11
SQMVSAM.Z 127, 128
SQMWORK 93, 94, 95, 96, 97, 174
Standard labels
loading 11
Startup parameters 1
STORPOOL
for Monitor tables 17, 27
for SQLMAINT DBSPACE 16
for SQMHELP DBSPACE 16, 26, 31
System
software 3

T

table job submission
optional parameters 123, 125
required parameters 117
table parameter file
define 24
table reorganization
submit screen 116
table utility
CREATE TABLE function 119
creating table parameters 121
description 105
DROP TABLE function 108
inserting and deleting data 120
invocation 106

table utility *(continued)*
LIST TABLES function 107
REORGANIZE TABLE function 109
UPDATE STATISTICS function 124
table utility tool 3
TAPE FILE NAME 83
TRANSIDs 5
CEMS 5
for CICS report controller 5
for dbspace analysis menu 5
for dbspace reorganization 5
for group authorization tool 5
for help facility 5
for main menu 5
for operator commands menu 5
for package utility tool 5
for work file label definition 5
SQDR 5
SQFM 5
SQGA 5
SQHM 5
SQM 5
SQMM1 5
SQOM 5
SQPM 5

U

UNCLUSTERED INDEX 77
Unload DBSPACE 62
unload table 116
UPDATE ALL STATISTICS 82
update statistics 64
UPSTAT_DATE 75
UPSTAT_ELAPSED 75
UPSTAT_TIME 75
User Group
adding users to 55
defining 54
User ID 33

V

VERSION NUMBER 97
VOLUME SEQUENCE NUMBER 97
VOLUME SERIAL NUMBER 95, 97
VSAM
allocating VSAM space 10
definitions 10

W

Work File Label Definition
disk work file label definition fields 95
disk work file label definition screen 94
highlights 3
Work File Label Definition *(continued)*
how it works 94
initial definition 19
migration definition 28
overview 93
special considerations 98
size of defined files 98
tape work file label definition fields 96, 97
tape work file label definition screen 96
Work File Label Definition screen 93
Work files
initial definition 19
migration definition 28

Communicating Your Comments to IBM

DB2® Server for VSE & VM
Control Center Installation
and Operations Guide for VSE
Version 6 Release 1

Publication No. GC09-2678-00

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 VSE
Version 6 Release 1

Publication No. GC09-2678-00

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
GC09-2678-00

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

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.

GC09-2678-00

Spine information:

DB2® Server for VSE & VM

Control Center Installation and Operations Guide for VSE

Version 6 Release 1