

DB2 Server for VSE & VM



# Control Center Operations Guide for VSE

*Version 7 Release 3*



DB2 Server for VSE & VM



# Control Center Operations Guide for VSE

*Version 7 Release 3*

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 135.

This document contains proprietary information of IBM. It is provided under a license agreement and is protected by copyright law. The information contained in this publication does not include any product warranties, and any statements provided in this manual should not be interpreted as such.

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

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.

© Copyright International Business Machines Corporation 1997, 2003. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>About This Manual</b> . . . . .	<b>v</b>
Who Should Use This Manual . . . . .	v
Conventions Used in This Manual . . . . .	v
Organization of This Manual . . . . .	v
Prerequisite IBM Publications . . . . .	vi

<b>Summary of Changes</b> . . . . .	<b>vii</b>
Summary of Changes for DB2 Version 7 Release 3 . . . . .	vii
Enhancements, New Functions, and New Capabilities . . . . .	vii
Reliability, Availability, and Serviceability Improvements . . . . .	viii

<b>Chapter 1. Introduction</b> . . . . .	<b>1</b>
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 . . . . .	2
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 . . . . .	3
About Control Center . . . . .	4
Control Center's DBA ID . . . . .	4
How to Invoke . . . . .	4
Before You Use Control Center . . . . .	5
Changing to a New Database . . . . .	5

<b>Chapter 2. Getting Started</b> . . . . .	<b>7</b>
Getting Started With Control Center . . . . .	7
Control Center Sign On . . . . .	8

<b>Chapter 3. Using the Operator Command Interface Tool</b> . . . . .	<b>9</b>
---	----------

<b>Chapter 4. DBSPACE Reorganization Tool</b> . . . . .	<b>13</b>
When To Reorganize . . . . .	13
Features . . . . .	14
How the DBSPACE Reorganization Tool Works . . . . .	14
Using the DBSPACE Reorganization Utility Screen . . . . .	15
Optional Parameters . . . . .	17
Using the DBSPACE REORGANIZATION Tool . . . . .	19
Option 1 - GENERATE DDL . . . . .	19
Option 2 - UNLOAD DBSPACE . . . . .	19

Option 3 - REORGANIZE DBSPACE . . . . .	20
Option 4 - RELOAD DBSPACE . . . . .	20
DBSPACE Reorganization Submit Screen . . . . .	20
VSE/POWER Parameters . . . . .	21
Single User Mode (SUM) DBSPACE Reorganization . . . . .	22
Before You Choose Single User Mode Execution . . . . .	22
Single User Mode Parameters . . . . .	22
DBSPACE Reorganization Tape Support . . . . .	23
Unloading to Tape . . . . .	23
Special Considerations . . . . .	24
Repetitive Scheduling . . . . .	24
Failure Restart . . . . .	24
Problem Analysis . . . . .	24

<b>Chapter 5. DBSPACE Analysis Tools</b> . . . . .	<b>27</b>
About the DBSPACE Analysis Tools . . . . .	27
Before You Begin . . . . .	27
How the DBSPACE Analysis Tools Work . . . . .	27
SQLMAINT Table . . . . .	28
DBSPACE Analysis Utility Screen . . . . .	29
Functions . . . . .	30
Selection Options . . . . .	31
DBSPACE Reorganization Criteria (CRITERIA) . . . . .	31
Update Statistics Analysis Tool . . . . .	32
DBSPACE Reorganization Analysis Tool . . . . .	33
DBSPACE Analysis Submit Screen . . . . .	35
VSE/POWER Job Parameters . . . . .	36
DBSPACE Analysis Job Options . . . . .	36
Additional Topics . . . . .	37
Initial Execution . . . . .	37
Reorganization Work Space . . . . .	37

<b>Chapter 6. Work File Label Definition Tool</b> . . . . .	<b>39</b>
About the Work File Label Definition Tool . . . . .	39
Work File Label Definition Screen . . . . .	39
How the Work File Label Definition Tool Works . . . . .	39
Disk Work File Label Definition Screen . . . . .	40
Disk Work File Label Definition Fields . . . . .	41
Tape Work File Definition Screen . . . . .	42
Special Considerations . . . . .	45

<b>Chapter 7. CICS Report Controller Interface Tool</b> . . . . .	<b>47</b>
About the CICS Report Controller Interface Tool . . . . .	47
A Sample CICS Report Controller Session . . . . .	47

<b>Chapter 8. Control Center Help Facility</b> . . . . .	<b>51</b>
About the Help Facility . . . . .	51

<b>Chapter 9. Package Utility</b> . . . . .	<b>53</b>
Introduction . . . . .	53
Package Utility Functions . . . . .	53
Package Function Descriptions . . . . .	54

Package Migration . . . . .	54
Invocation . . . . .	54
Package Utility Parameters . . . . .	55
Using the Package Utility. . . . .	56

**Chapter 10. Group Authorization Tool 59**

About the Group Authorization Tool . . . . .	59
Functions . . . . .	60
ADD a Group . . . . .	60
DROP a Group . . . . .	60
Manage Group Objects and Users . . . . .	60
Manage Privileges . . . . .	61
LIST Functions . . . . .	61
Using the Group Authorization Tool . . . . .	61
Step 1: Define Application Groups. . . . .	62
Step 2: Add (or Drop) Objects to the Application Group . . . . .	63
Step 3: Define User Groups . . . . .	64
Step 4: Add Users to the User Groups . . . . .	65
Step 5: Grant Authorities to the User Groups . . . . .	66
Special Considerations. . . . .	67

**Chapter 11. The Monitor Maintenance Menu . . . . . 69**

How the Monitors Work . . . . .	69
Options and Monitors Available . . . . .	70
Monitor Thresholds and VSE Console Messages . . . . .	70
Description of Monitor Options. . . . .	71
Start the Monitor Kernel . . . . .	71
Stop the Monitor Kernel . . . . .	71
List Monitors . . . . .	71
Add a Monitor . . . . .	72
Modify a Monitor . . . . .	72
Delete a Monitor . . . . .	72
Display a Monitor . . . . .	72
View Data . . . . .	72
Reset Data. . . . .	72
Print Report . . . . .	73
Types of Monitors . . . . .	73
SHOW ACTIVE . . . . .	73
SHOW LOCK . . . . .	73
SHOW DBEXTENT. . . . .	73
SHOW LOG . . . . .	73
SHOW CONNECT . . . . .	74
SHOW DBSPACE . . . . .	74
COUNTER * . . . . .	74
Invocation. . . . .	74
How To Use the Monitors . . . . .	74
Adding A Monitor . . . . .	74
Using the Monitor Maintenance Menu . . . . .	75
Specifying Monitor Thresholds . . . . .	77

Starting a Monitor . . . . .	78
Changing a Monitor . . . . .	78
Viewing Monitor Data. . . . .	79
Stopping a Monitor. . . . .	80

**Chapter 12. Table Utility . . . . . 81**

Invocation. . . . .	81
LIST TABLES. . . . .	83
List Tables Processing Flow . . . . .	84
DROP TABLE . . . . .	84
Drop Table Processing Flow . . . . .	84
REORGANIZE TABLE. . . . .	85
Reorganize Table Processing Flow . . . . .	87
Table Reorganization Menu Required Parameters . . . . .	88
Table Reorganization Menu Optional Parameters . . . . .	88
Using the TABLE REORGANIZATION Option . . . . .	90
Table Reorganization Submit Screen . . . . .	91
CREATE TABLE. . . . .	93
Create Table Processing Flow . . . . .	93
Using the Create Table Function . . . . .	94
Create Table Parameters . . . . .	96
UPDATE STATISTICS . . . . .	98
Update Statistics Submit Screen Required Parameters . . . . .	98
Update Statistics Submit Screen Optional Parameters . . . . .	98

**Chapter 13. Installing Stored Procedures Support . . . . . 101**

**Appendix A. Reorganization Job Streams. . . . . 105**

**Appendix B. DBSPACE and Table Reorganization Tool Related Files . . . . . 127**

**Notices . . . . . 135**  
Trademarks . . . . . 137

**Glossary . . . . . 139**

**Bibliography. . . . . 141**

**Index . . . . . 145**

**Contacting IBM . . . . . 149**  
Product information . . . . . 149

---

## 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 7.3 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*, and *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 7 Release 3 Modification 0.

---

### Organization of This Manual

**“Summary of Changes” on page vii** summarizes the changes made to DB2 Server for VSE & VM for this release.

**Chapter 1, “Introduction” on page 1** introduces the Control Center product and tool set.

**Chapter 2, “Getting Started” on page 7** introduces you to the Control Center Main Menu.

**Chapter 3, “Using the Operator Command Interface Tool” on page 9** describes how to use the Operator Command interface to issue SHOW and COUNTER commands to any connected DB2 Server in your VSE environment.

**Chapter 4, “DBSPACE Reorganization Tool” on page 13** describes the automated Control Center tools for reorganizing DBSPACES within a database.

**Chapter 5, “DBSPACE Analysis Tools” on page 27** 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 6, “Work File Label Definition Tool” on page 39** describes the Work File Label Definition tools that help you define the reorganization tool’s work files.

**Chapter 7, “CICS Report Controller Interface Tool” on page 47** describes the Control Center interface to the CICS® Report Controller function.

**Chapter 8, “Control Center Help Facility” on page 51** describes the Control Center Help Facility.

**Chapter 9, “Package Utility” on page 53** describes how you can automate tasks associated with packages within a database.

**Chapter 10, “Group Authorization Tool” on page 59** 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 to individual users or groups of individuals.

**Chapter 11, “The Monitor Maintenance Menu” on page 69** describes how the Database Monitoring tools are used to monitor database status and activities.

**Chapter 12, “Table Utility” on page 81** 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.

**Chapter 13, “Installing Stored Procedures Support” on page 101** describes the stored procedures that are used to do local processing.

**Appendix A, “Reorganization Job Streams” on page 105** gives examples of the job streams generated when a database or table reorganization is requested by the user.

**Appendix B, “DBSPACE and Table Reorganization Tool Related Files” on page 127** lists the DBSPACE Reorganization Tool related files and explains their use.

---

## **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

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

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

---

## Summary of Changes for DB2 Version 7 Release 3

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

### Enhancements, New Functions, and New Capabilities

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

#### Cancel TCP/IP Agent in VM

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

#### Force Inactive Users

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

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

#### CLI/JDBC V8 Redesign

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

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

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

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

### **Control Center for VSE Enhancements**

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

### **Control Center for VM Enhancements**

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

## **Reliability, Availability, and Serviceability Improvements**

### **Alternate Logging**

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

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

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

### **Data Restore Changes**

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

### **Release Empty Pages**

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

### **More Information on TCP/IP Messages**

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

## **Migration Considerations**

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



---

## 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. 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 helps 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. You must 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.

## Introduction

### 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 to 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. The DBSPACE Analysis tools then 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 or date 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 the 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 3 or later
- DB2 Server for VSE Version 7 Release 3 (5697-F42)

## Introduction

- VSE/REXX (part of VSE/Central Functions, Program Number 5686-066, and all subsequent releases)

Control Center Version 7.3 can work with previous version databases of DB2 and SQLDS. However, since the Control Center Version 7.3 packages must reside in those databases, installing Version 7.3 will overlay any earlier packages and will thereby preclude using earlier versions of Control Center with those databases. What this means is that a database can be used with one and only one version of Control Center.

- 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. Control Center 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 requires an ID with DBA authority to connect to every database it manages. Be sure you have completed the installation step that grants DBA authority to this ID.

## How to Invoke

### Using the Screens

You can enter the transaction ID "SQM" from a blank CICS 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 Use Control Center

### Distribution Library:

The default delivery library for Control Center is PRD2.CCF730. The installation procedures in the *DB2 Server for VSE Control Center Program Directory* describe how you can specify a different library. If you are using DB2 Server for VSE Version 7.1, the default installation library is PRD2.DB2710; for DB2 Server for VSE Version 7.2, it is PRD2.DB2720; and for DB2 Server for VSE Version 7.3, it is PRD2.DB2730.

### 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 fixes (PTFs). An updated version or release replaces the entire program code. A PTF replaces the changed program code only.

## Changing to a New Database

The Control Center installation process is defined in the *DB2 Server for VSE Control Center Program Directory*. Its instructions include initializing CICS and DB2 databases for use with Control Center. If you want to add a new database to those already initialized for Control Center, you must do the following steps.

### STEP TITLE

- Ensure that the Control Center DBA ID has proper DBA Authority
- Define and Load the Help Table
- Define the Maintenance Tracking Table
- Define the Monitor Tables
- Define the Group Authorization Tables
- Load Packages into Servers
- Installing Stored Procedure Support (Optional)

## Introduction

---

## Chapter 2. 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.

```
mm/dd/yyyy          CONTROL CENTER V7.3          hh:mm:ss
*-----*
OPTION =>  _  A          USER ID: B
DATABASE =>  _____ C          CICS ID: D
E ***** DBA FUNCTIONS *****
1 OPERATOR COMMANDS
2 DBSPACE REORGANIZATION
3 DBSPACE ANALYSIS
4 WORK FILE LABEL DEFINITION
5 CICS REPORT CONTROLLER
6 HELP FACILITY
7 PACKAGE UTILITY
8 GROUP AUTHORIZATION
9 MONITOR UTILITY
10 TABLE UTILITY
11 QMF
*-----*
F
G PRESS ENTER TO SELECT FUNCTION
H ENTER F1=HELP F3=EXIT
SQC01 -----*
```

Figure 1. Control Center Main Menu

Some things you should know about typical Control Center menus 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 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 from the current menu.

**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 1 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.

### Control Center Sign On

When you enter the main Control Center transaction ID (SQM), or one of the transactions that directly invoke a Control Center function, you will be presented with the Sign On Panel. Enter the DBA ID and the password, and press the Enter key to continue. You will remain signed on until you exit the transaction.

```
mm/dd/yyyy          CONTROL CENTER V7.3          hh:mm:ss
*-----*
SIGN ON PANEL
=====
DATABASE => VSEMCHMB          USER ID: SQLDBA
                               CICS ID: DBDCCICS

PLEASE ENTER CONTROL CENTER DBA ID AND PASSWORD

CC DBA ID =>          PASSWORD =>

*-----*
ENTER F1=HELP F3=EXIT          SQC00 -----*
```

Figure 2. Control Center DBA Sign On

---

## Chapter 3. 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. Multiple screens are used to display the set of operator commands that you can select. You can scroll between them using the Forward and Backward function keys.

This tool can be invoked by selecting option 1 from the main menu or by using the CICS transaction ID "SQOM".

The Operator Commands menus use this syntactical notation:

- Words in uppercase must be typed exactly as shown
- Words in lowercase should be replaced by your particular values in uppercase or lowercase
- Sets of phrases or words from which you must choose one are enclosed in parentheses and separated by vertical bars
- Default values are shown in a non-green (usually turquoise) color and are underlined.

## Using the Operator Command Interface Tool

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*----- OPERATOR COMMANDS -----*
OPTION ==>
PARMS ==>
DATABASE => SQLDBA

1 SHOW ACTIVE
2 SHOW ADDRESS      module name
3 SHOW BUFFERS
4 SHOW CONNECT      ( ALL | user-id | AGENT n | LUWID id
                    | ACTIVE | WAITING | INACTIVE )
5 SHOW DBCONFIG
6 SHOW DBEXTENT
7 SHOW DBSPACE      dbspace number
8 SHOW INDOUBT
9 SHOW INITPARM
10 SHOW INVALID
11 SHOW LOCK ACTIVE
12 SHOW LOCK DBSPACE ( ALL | dbspace number)

*----- SQC11A -----*
Default values are shown in turquoise.
ENTER OPTION, PARMS, AND DATABASE NAME AND PRESS ENTER
ENTER F1=HELP F3=EXIT F8=FWD

```

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*----- OPERATOR COMMANDS -----*
OPTION =>
PARMS =>
DATABASE => SQLDBA

13 SHOW LOCK GRAPH  ( auth-id | AGENT n )
14 SHOW LOCK MATRIX
15 SHOW LOCK USER   ( ALL | auth-id | AGENT n )
16 SHOW LOCK WANTLOCK ( ALL | auth-id | AGENT n )
17 SHOW LOG
18 SHOW LOGHIST     ( ALL | n ) ( SERVICE )
19 SHOW POOL        ( ALL | SUMMARY | DELETED | n )
20 SHOW PROC        ( proc-name | proc-name AUTHID auth-id
                    | * | * AUTHID auth-id )
21 SHOW PSERVER     ( * | procsvr-name | GROUP * | GROUP svr-group-name )
22 SHOW STORAGE
23 SHOW SYSTEM
30 COUNTER          ( * | counter-name )

*----- SQC11B -----*
Default values are shown in turquoise.
ENTER OPTION, PARMS, AND DATABASE NAME AND PRESS ENTER
ENTER F1=HELP F3=EXIT F7=BWD

```

Figure 3. DBA Operator Commands Screens

The DBA Operator Commands menus are shown in Figure 3.

The database server you are currently working with is identified in 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. The option number can be entered even if the option and syntax information are not on the screen currently being shown.

## Using the Operator Command Interface Tool

- Enter any required parameters. (The parameters shown in uppercase are to be entered as shown. Those shown in lowercase are for you to supply.) Default values are shown in a different color (usually turquoise) and are underlined.
- 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 SQLDBA:

```
OPTION ==> 1
PARMS ==>
DATABASE => SQLDBA
```

you would see:

```
mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*----- OPERATOR COMMAND DISPLAY SCREEN -----*
COMMAND => SHOW ACTIVE
DATABASE => SQLDBA

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 4. Show Active Command Display Screen

**Note:** When the Operator Command Selection Menu (Figure 3 on page 10) is being displayed, F1=HELP will display an explanation of the individual commands. When the Operator Command Display Screen (Figure 4) is being displayed, F1=HELP displays help information for the screen, not for the operator command results.

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





---

## Chapter 4. DBSPACE Reorganization 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, and on a one-time or repetitive basis.

---

### 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.

## DBSPACE Reorganization Tool

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 Works

When you choose full DBSPACE reorganization, Control Center generates and submits a job to:

1. Link and establish communication with the target server.
2. Connect as user SQLREORG.
3. Verify the availability of the new DBSPACE (if specified).
4. Gather system catalog information about the specified DBSPACE and create 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. Unload the DBSPACE data to the specified disk or tape.
  6. Execute the SQLDBSU command file from the Database Services Utility to reorganize the DBSPACE and rebind any dependent packages.
  7. Update the SQLMAINT table with the date, time, and duration of the reorganization (see Chapter 5, "DBSPACE Analysis Tools" on page 27).

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 REOnnnnn (where *nnnnn* is a random number).

---

### Using the DBSPACE Reorganization Utility Screen

To display the DBSPACE Reorganization Utility menu shown in Figure 5 on page 16, choose Option 2 on the Control Center Main Menu or enter the transaction ID **SQDR** on a CICS screen.

## DBSPACE Reorganization Tool

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          DBSPACE REORGANIZATION UTILITY          -----*
DATABASE => SQLDBA_____
OWNER    => _____
DBSPACE  => _____
FILE     => 1 (1-3)
OPTION   => 3 (1=GENERATE DDL          3=REORGANIZE DBSPACE   )
          (2=UNLOAD 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 5. DBSPACE Reorganization Utility Screen

You must enter the first three fields (DATABASE, OWNER, DBSPACE name) 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 DBSPACE name 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. Type the number of the file (1=small, 2=medium, 3=large) you want to use in the FILE field.) The FILE number determines what SAM data file to use if you do not enter 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:

<u>Option</u>	<u>Description</u>
<b>1 GENERATE DDL</b>	This option extracts from the database all of the DDL required to re-create a DBSPACE and the objects it contains. The DDL is saved in the punch queue for inspection, alteration, or backup.
<b>2 UNLOAD DBSPACE</b>	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 shown in Figure 56 on page 105.
<b>3 REORGANIZE DBSPACE</b>	This option results in a full DBSPACE reorganization. A jobstream is created that captures the DDL, unloads the DBSPACE, drops, acquires,

## DBSPACE Reorganization Tool

recreates, and reloads the DBSPACE. Error recovery logic is also included. An example of a REORGANIZE DBSPACE job is shown in Figure 57 on page 107.

### 4 RELOAD DBSPACE

This option submits a job to recreate and reload a DBSPACE that has been unloaded by Option 2. This is basically a DBSPACE recovery facility. An example of the job created to do this is shown in Figure 58 on page 110.

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 on the screen 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 DBSPACE name 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

Specifies 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.

## DBSPACE Reorganization Tool

<b>PCTFREE</b>	Indicates 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.
<b>ALTER PCTFREE</b>	Indicates 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.
<b>PCTINDEX</b>	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.
<b>LOCK</b>	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.
<b>REBIND PACKAGE</b>	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).
<b>UPDATE ALL STATISTICS</b>	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.
<b>COMMITCOUNT</b>	Used to specify the frequency of COMMITS during reload processing. Enter a number in the range 1 through 2,147,483,647 (without the commas) to cause a COMMIT WORK to be executed after that number of input rows has been reloaded.
<b>TLBL FILE-ID</b>	Used to specify that data should be unloaded to tape instead of disk. The tape file must have been defined using the WORK FILE LABEL DEFINITION tool. This does not apply to DDL; DDL is ALWAYS unloaded to disk.
<b>DDL STATEMENTS</b>	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, 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 6 is an example of the jobstream produced by Control Center to generate DDL for the PUBLIC.SAMPLE DBSPACE.

```
* $$ JOB JNM=GENDDL,CLASS=0,DISP=D,PRI0=3
* $$ LST PRI=3
* $$ PUN PRI=3
// JOB GENDDL MUM GENERATE DDL
// OPTION LOG
*****
* STEP0001 UNLOAD DDL
*****
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMDDL,'L.SQLDS730.PUBLIC.SAMPLE',0,VSAM,
RECORDS=001000,RECSIZE=80,DISP=(NEW,KEEP),CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS006,SYSPCH
// ASSGN SYS011,SYSLST
// EXEC SQB01,SIZE=AUTO
%%SQLDBA PUBLIC SAMPLE 1 N SQLDBA SQLDBAPW
/*
/&
```

Figure 6. 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 56 on page 105 shows a jobstream that was generated by Control Center to unload the PUBLIC.SQMHELP DBSPACE. See Appendix B, "DBSPACE and Table Reorganization Tool Related Files" on page 127 for information about the file.

## DBSPACE Reorganization Tool

### 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 57 on page 107 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 58 on page 110 shows a sample jobstream that was generated by Control Center to reload the PUBLIC.SQMHELP DBSPACE.

---

## DBSPACE Reorganization Submit Screen

Figure 7 shows the DBSPACE Reorganization Submit screen.

```
mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          DBSPACE REORG SUBMIT          *-----*
*****          VSE/POWER JOB PARAMETERS          *****
JOBNAME => _____ CLASS ==> A  PRI => 3  DISP ==> D  (D,H,L,K)
FROM    => _____ DUETIME => ____ (HHMM)  DUEDATE => _____ (AABBY)
DUEDAY  => _____ (DAY NAMES/NUMBERS)
OTHER   => _____

-----          SINGLE USER MODE PARAMETERS          -----
SUM?    => 2 (1=YES/2=NO)          DATABASE DEFINITION PROC => _____
LOGMODE => _ (L,A,Y,N)          NDIRBUF => _____          NPAGBUF => _____
*-----*          SQC06          *-----*

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

Figure 7. 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.



## VSE/POWER Parameters

<u>Parameter</u>	<u>Description</u>
<b>JOBNAME</b>	Specifies the name by which the DBSPACE REORGANIZATION job and its associated VSE/POWER queue entries is to be known.
<b>CLASS</b>	Specifies the class or partition in which you want this job to run. Class defaults to A.
<b>PRI</b>	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.
<b>DISP</b>	Specifies how the job is to be handled in the reader queue. Disposition may be specified as: <ul style="list-style-type: none"> <li>• D - Delete after processing</li> <li>• H - Hold until released</li> <li>• K - Keep after processing</li> <li>• L - Leave in the queue</li> </ul> <p>Disposition defaults to D.</p>
<b>FROM</b>	Specifies the ID of the user being allowed to manipulate or retrieve the job. Defaults to the CICS user ID.
<b>DUETIME</b>	Specifies the processing start time using hh for hour and mm for minute in 24-hour clock time (OPTIONAL). This can only be 0001 through 2359.
<b>DUEDATE</b>	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 (OPTIONAL). No check is made for invalid leap days nor passed dates.
<b>DUEDAY</b>	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 single quotes (apostrophes). 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 (OPTIONAL).
<b>OTHER</b>	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 (OPTIONAL).

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 8 on page 22 is a sample of the DBREORG Report.

## DBSPACE Reorganization Tool

```
SQB02                CONTROL CENTER FOR VSE
hh:mm:ss            DBSPACE REORGANIZATION REPORT          mm/dd/yyyy
                   DBSPACE 1                            DBSPACE 2
                   -----                            -----
DATABASE:           SQLDBA                            SQLDBA
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 8. 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 & VM 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

<u>Parameter</u>	<u>Description</u>
SUM?	Specify "1" (YES) to cause a Single User Mode job to be submitted. This parameter defaults to "2" (Multiple User Mode).
<b>DATABASE DEFINITION PROC</b>	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.

<b>LOGMODE</b>	Specifies the logmode you want Control Center to use during Single User Mode processing. You must enter a value. Valid values are: <ul style="list-style-type: none"><li>• A - All database changes are logged and regular database archives are maintained.</li><li>• L - All database changes are logged and regular log archives are maintained.</li><li>• N - No database changes are logged.</li><li>• Y - All database changes are logged but no archives are maintained.</li></ul>
<b>NDIRBUF</b>	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. If specified, the value must be in the range 10—400000. This parameter is not required.
<b>NPAGBUF</b>	The number of 4096-byte data pages to be kept in storage. Again, bigger is better, within reason. If specified, the value must be in the range 10—400000. Entry of this parameter is not required.

SUM processing requires that the database be ended prior to execution. If you submit a SUM job for immediate execution, it will fail. This is because the database server must be stopped when a SUM job is run but has to be up when you run the Control Center function that creates the job. So, you must delay the execution of the job from the reader manually or by using the scheduling controls of DUE TIME, DUE DATE, or DUE DAY.

When a job step requires access to a database, the database is started and the application program is executed. When the application program ends, control returns to the database server and the database is ended. The database remains down until it is restarted. Remember that changing the logmode will probably force some combination of coldlogs, and log or database archives.

Figure 59 on page 112 is an example of a Single User Mode REORGANIZE DBSPACE (Option 3). These parameters were used when choosing the DBspace Reorganization utility's option 3 (REORGANIZE DBSPACE): owner = PUBLIC, DBspace = SAMPLE, REBIND PACKAGE = 1 (YES), UPDATE ALL STATISTICS = 1 (YES), DDL STATEMENTS = 500. On the submit screen, these options were used: SUM = 1 (YES), LOGMODE = N, NDIRBUF = 10, NPAGBUF = 10.

---

## 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 data unload media. As a result, the jobstream that Control Center builds and submits is quite different. Figure 60 on page 116 is an example of a REORGANIZE DBSPACE (Option 3) using 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 for the same DBSPACE 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 SQMPARM 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.

## DBSPACE Reorganization Tool

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).

## DBSPACE Reorganization Tool

---

## Chapter 5. 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 into 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 can choose what criteria are to be used to identify 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 Utilities screen (Figure 10 on page 30) to list UPDATE STATISTICS candidates or Option 3 to list REORG candidates, a detailed analysis of each DBSPACE is conducted to determine the

## DBSPACE Analysis Tools

degree of need for maintenance. The results of this analysis are displayed. The DBA can then choose what DBSPACES to reorganize or to run UPDATE STATISTICS against.

The DBSPACE Analysis tool uses a database table (SQLMSTR.SQLMAINT) to maintain information about DBSPACES in the database. During execution, each DBSPACE is considered for maintenance based on parameters you enter. 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 you select.
6. Updates the SQLMAINT table for each DBSPACE that receives maintenance.

The selection process consists of one step for UPDATE STATISTICS analysis and two 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 31).
2. For reorganization, applies REORG CRITERIA to the list selected in step 1 above to produce a list of candidates that need reorganization. (See "DBSPACE Reorganization Criteria (CRITERIA)" on page 31).

## SQLMAINT Table

The Control Center installation process creates SQLMAINT in a public DBSPACE in the target database as shown in Figure 9.

```
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 9. 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 as a candidate for reorganization).

The columns that make up SQLMSTR.SQLMAINT are:

<u>Column Name</u>	<u>Description</u>
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 most recent UPDATE STATISTICS
UPSTAT_TIME	Time of the most recent UPDATE STATISTICS
UPSTAT_ELAPSED	Elapsed time of the most recent UPDATE STATISTICS job for this DBSPACE
REORG_DATE	Date of the most recent reorganization
REORG_TIME	Time of the most recent reorganization
REORG_ELAPSED	Elapsed time of the most recent reorganization job
REORG_FREEPCT	FREEPCT value in the most recent reorganization
REORG_PCTINDX	PCTINDX value in the most recent reorganization
REORG_STATUS	A two-character reorganization status indicator
REORG_WEIGHT	An integer indicating the relative need for reorganization. This is derived using a formula based on other information.
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 10 on page 30 shows the DBSPACE Analysis Utilities screen.

## DBSPACE Analysis Tools

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          DBSPACE ANALYSIS UTILITIES -----*
DATABASE => SQLDBA
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 10. 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:

<u>Function</u>	<u>Description</u>
<b>1 INITIALIZE CONTROL TABLE</b>	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.
<b>2 UPDATE STATISTICS CANDIDATES</b>	Lists the DBSPACES that need their statistics updated. The candidate list is displayed based upon the other selection options chosen.
<b>3 LIST REORG CANDIDATES</b>	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.

<u>Parameter</u>	<u>Description</u>
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. <b>1 indicates yesterday.</b>

## 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 12 on page 34). 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:

<u>Criteria</u>	<u>Description</u>
<b>1 CLUSTERRATIO &lt; nnnn</b>	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.

## DBSPACE Analysis Tools

- 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 11. The list was created by specifying:

```
OPTION ==> 2
OWNER ==> PUBLIC
PAGES: MINIMUM ==> 0                MAXIMUM ==> 9999999
DAYS SINCE ==> 4
```

```
mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          UPDATE STATISTICS LIST -----*
DATABASE => SQLDBA
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
*-----*          SQC09 -----*

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

Figure 11. 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 most recent Update Statistics job for this DBSPACE.
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 35).

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 12 on page 34. The list was created by specifying:

```

OPTION ==> 3
OWNER ==> PRIVATE
PAGES: MINIMUM ==> _____ MAXIMUM =====> 8192
DAYS SINCE =====> 14
REORG CRITERIA ==> 4
    
```

## DBSPACE Analysis Tools

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*----- REORG CANDIDATES LIST -----*
DATABASE => SQLDBA

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

*-----*----- SQC09 -----*

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

```

Figure 12. DBSPACE Reorganization Analysis List Screen

Figure 12 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 most recent reorganization job for this DBSPACE.
- 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 7 on page 20 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 13 shows the DBSPACE Analysis SUBMIT screen.

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*----- DBSPACE ANALYSIS SUBMIT -----*
*****          VSE/POWER JOB PARAMETERS          *****
JOBNAME => _____ CLASS ==> A  PRI => 3  DISP ==>> D  (D,H,L,K)
FROM    => VS02    DUETIME => ____ (HHMM)  DUE DATE => _____ (AABBY)
DUE DAY => _____
OTHER   => _____
*****          JOB OPTIONS          *****
UPDATE ALL STATISTICS => 2 (1=YES/2=NO)          REBIND PACKAGES => 1 (1/2)
COMMITCOUNT          => _____
TLBL FILE-ID          => _____          FILE #          => 2 (1-3)
*----- SQC10 -----*

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

```

Figure 13. 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.

## DBSPACE Analysis Tools

<u>Parameter</u>	<u>Description</u>
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: <ul style="list-style-type: none"><li>• D - Delete after processing</li><li>• H - Hold until released</li><li>• K - Keep after processing</li><li>• L - Leave in the queue</li></ul> 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 single quotes (apostrophes). 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.

## VSE/POWER Job Parameters

### DBSPACE Analysis Job Options

<u>Parameter</u>	<u>Description</u>
UPDATE ALL STATISTICS	Specifies 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 2 (NO).



<b>REBIND PACKAGES</b>	Specifies by 2 (NO) that the DBSPACE Reorganization tool should <i>not</i> rebind packages as part of a DBSPACE reorganization. The default is 1 (YES), which means rebinding will occur.
<b>TAPE FILE NAME</b>	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 using the Work File Label Definition function. See Chapter 6, "Work File Label Definition Tool" on page 39.
<b>FILE #</b>	Defines which of the previously defined data and DDL files (See "How the Work File Label Definition Tool Works" on page 39 ) is to be used for the unloaded data and DDL. Valid values are 1, 2, or 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 you want 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 a database command such as:

```
UPDATE SQLMSTR.SQLMAINT SET UPSTAT_DATE = '2000-04-15', REORG_DATE  
= '2000-04-15' WHERE OWNER < 'N'
```

This would change about half of the DBSPACES to a different maintenance date. You could then use the List Reorg Candidates function (option 3) with an appropriate DAYS SINCE parameter so that only those DBSPACES whose names start with N through Z would be selected.

### Reorganization Work Space

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



---

## Chapter 6. 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 14 is displayed. This is also the screen that is presented when the CICS transaction SQFM is entered directly.

```
mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*----- WORK FILE LABEL DEFINITION -----*
OPTION              => 1 (DEFINE WORK FILE          )
*****
FILE TYPE           => (1=DATA                2=DDL          )
                   (3=PACKAGES              4=MESSAGES      )
FILE/PARTITION     =>          FILE TYPE 1 - 2 : FILE SIZE NUMBER
NUMBER              :          FILE TYPE 3   : PARTITION NUMBER
                   :          FILE TYPE 4   : NOT USED
TLBL FILE-ID       =>
*----- SQ003 -----*
PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT
```

Figure 14. 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

## Work File Label Definition Tool

Data work files are used by the DBSPACE Reorganization, DBSPACE Analysis, and Table Reorganization 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; see Figure 14 on page 39. Remember that tape may only be used for data work files.

Defining disk files is more involved. Because disks do not have the same capacity as tapes, file size becomes important. The FILE/PARTITION NUMBER field is used to identify small (1), medium (2), and large (3) DDL and DATA work files. If you define only one set of disk work files, we suggest you choose file number 2 because that is the default value on the screens.

When defining a package work file, the field is used to enter a partition number (job class) in which the View Package job stream will run; valid values are 0-9 and A-Z. These two types of values share the same input field on the screen.

As an example, if the user wants to define a medium-size disk DATA work file, the FILE TYPE would be 1 (DATA) and the FILE NUMBER would be 2 (medium). The file name (and JCL) will be generated as file name SQMDAT2 with file identifier as SQLMSTR.DATA.FILE2. The actual size of the files defined is entered on the next screen and depends on the DBA's estimates of the needed space. (See Chapter 4, "DBSPACE Reorganization Tool" on page 13 for an explanation of how you can combine a disk dataset for the DDL and a tape dataset for the DATA.)

---

## 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 15 on page 41.

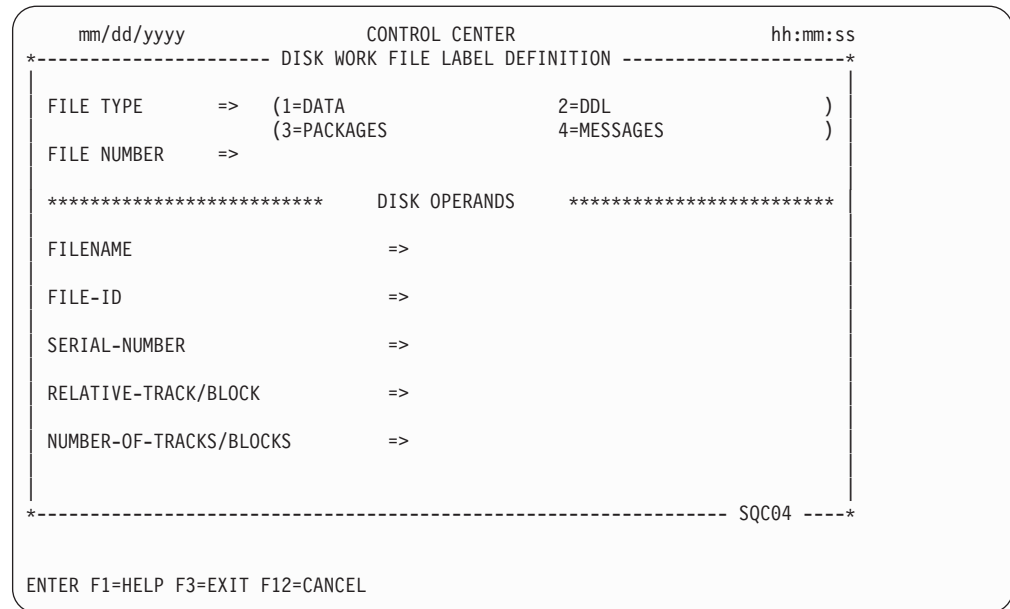


Figure 15. Disk Work File Label Definition Screen

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.

<u>Parameter</u>	<u>Description</u>
<b>VOLUME SERIAL NUMBER</b>	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.
<b>RELATIVE TRACK/BLOCK</b>	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 2,147,483,645 (without the commas), where the file is to begin.
<b>NUMBER OF TRACKS/BLOCKS</b>	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 2,147,483,645 (without the commas).

## Work File Label Definition Tool

The actual size of each file depends on the size of the DBSPACES in your database. Refer to the *DB2 Server for VSE Control Center Program Directory* 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 stored in 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 16 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                hh:mm:ss
*----- TAPE WORK FILE LABEL DEFINITION -----*
|
| FILENAME    => SQMTAPE                FILE-ID    =>
| *****   TAPE OPERANDS   *****
|
| FILE-SERIAL-NUMBER                =>
| VOLUME-SEQUENCE-NUMBER            =>
| FILE-SEQUENCE-NUMBER              =>
| GENERATION-NUMBER                 =>
| VERSION-NUMBER                    =>
| DATE                             =>          (YYYY/DDD OR 0-9999)
| DEVICE-CLASS                      => 1      (1-CARTRIDGE/2-TAPE)
| MODE                              =>
|
|----- SQC04 -----*
|
| ENTER F1=HELP F3=EXIT F12=CANCEL

```

Figure 16. 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

## Work File Label Definition Tool

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.

### Package Work File

The View Package Utility uses a SAM work file (SQLPKGx) to hold the unloaded package used to produce the Package Report. You must allocate a package work file for **each** partition that might use the View Package tool. (The partition number is used as the "x" in the name SQLPKGx.) Therefore, you may have to run this step more than once. Control Center unloads the package into a package work file and processes it to produce a package report. Note, only static partition numbers are allowed.

## Work File Label Definition Tool

For example, if you want to run View Package jobs in the background partition (CLASS 0), you would specify FILE TYPE 3 (PACKAGES) and FILE/PARTITION NUMBER 0 (CLASS 0 = BG) as:

FILE TYPE	=> <u>3</u>	(1=DATA	2=DDL	)
		(3=PACKAGES	4=MESSAGES	)
FILE/PARTITION NUMBER	=> <u>0</u>	FILE TYPE 1 - 2 :	FILE SIZE NUMBER	
		FILE TYPE 3 :	PARTITION NUMBER	
		FILE TYPE 4 :	NOT USED	

Press ENTER to display the Disk Work File Label Definition screen. Then enter SERIAL-NUMBER, RELATIVE-TRACK/BLOCK, and NUMBER-OF-TRACKS/BLOCKS. Note that as of Control Center Version 7.1, the suggested size of the package work file is 10K bytes; this is adequate for a report on a very big package. If necessary, you can redefine the file's size at any later time. If your existing file is larger, you can redefine it with this step, making it smaller.

SERIAL-NUMBER	=>	SYSWK1
RELATIVE-TRACK/BLACK	=>	03695_____
NUMBER-OF-TRACKS/BLOCKS	=>	20_____

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

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

## Package Messages Work File

You must also define a single package messages file that is used by all partitions. The (batch) Package Report job obtains report headings from this file. To define the file, specify FILE TYPE 4 (MESSAGES). When the FILE TYPE is 4, the FILE/PARTITION NUMBER is not used.

FILE TYPE	=> <u>4</u>	(1=DATA	2=DDL	)
		(3=PACKAGES	4=MESSAGES	)
FILE/PARTITION NUMBER	=>	=	FILE TYPE 1 - 2 :	FILE SIZE NUMBER
			FILE TYPE 3 :	PARTITION NUMBER
			FILE TYPE 4 :	NOT USED

Press ENTER to display the Disk Work File Label Definition screen. Then enter SERIAL-NUMBER, RELATIVE-TRACK/BLOCK, and NUMBER-OF-TRACKS/BLOCKS. Note that as of Control Center Version 7.1, the suggested size of the package work file is 10K bytes; this is adequate for a report on a very big package. If necessary, you can redefine the file's size at any later time. If your existing file is larger, you can redefine it with this step, making it smaller. The new size value includes an allowance for expansion.

SERIAL-NUMBER	=>	SYSWK1
RELATIVE-TRACK/BLACK	=>	03683_____
NUMBER-OF-TRACKS/BLOCKS	=>	12_____



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

### 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 7. 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 Control Center 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 Control Center 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 shown in Figure 17.

```
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 17. CICS Report Controller Main Menu Screen*

You enter selection 1 (Report selection) and are presented with the Report Selection screen as shown in Figure 18 on page 48.

## CICS Report Controller Interface Tool

```
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 18. CICS Report Controller Report Selection Screen

You tab down to the Report name field, enter "M\*" (show all reports whose name starts with "M"), and are presented with the Report List screen as shown in Figure 19.

```
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

PF1=Help PF3=End PF6=Bulk Change ENTER=Continue.
```

Figure 19. 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 20 on page 49.

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

09:14:50 Gathering table create DDL...
09:14:51 Gathering deactivate 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 20. CICS Report Controller Browse Screen

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



---

## Chapter 8. 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 21 by selecting Option 6 from Control Center's Main Menu.

```
mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          HELP FACILITY          *-----*
DATABASE => SQLDBA
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 21. Sample 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.

An alphabetically sorted 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. The list of topics and their order may change from one release of Control Center to another due to the addition of new topics.

In most cases, 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/>. Click on "More Info on DB Server ..." and then on "Control Center".





---

## Chapter 9. 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 a 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 useful 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 function, Control Center generates a batch job, 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.

For each of these functions, using information you supply on the Package Utility menu (Figure 22 on page 55), Control Center builds and submits a DBSU job.

## Package Utility

### Package Function Descriptions

Control Center stores packages for this tool in a Librarian library, SQLMSTR.PACKAGE. Periodically, you may want to delete packages no longer needed in this library. Ask your System Programmer to do the delete for you.

#### Unload Package

The job executes DBSU to unload the package to the VSE punch queue. It then catalogs the punch file containing the package into SQLMSTR.PACKAGE, where it can be accessed for reloading.

#### Reload Package

The job punches the package to the VSE reader queue from SQLMSTR.PACKAGE. It then executes DBSU to reload the package from the VSE reader queue. Options such as whether an existing package is to be replaced and whether to keep or revoke existing run privileges are supported. Optionally, you may specify a server name other than the one that is displayed.

**Note:** This function can only be used to reload packages that Control Center has unloaded.

#### Rebind Package

Optionally, you may specify that the package is only to be rebound if it is already marked as invalid. The job executes DBSU to rebind the package.

#### View Package

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.

In the job, the package messages are copied to a SAM file where they are accessible to the REXX/VSE package report program. Next, the package is unloaded 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 22 on page 55) when invoking the Reload function. During the job submission step, the parameter **DEST** must be specified. See “Package Utility Parameters” on page 55 and “Package Utility Job Submit Parameters” on page 56.

---

## 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**.

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          PACKAGE UTILITY          *-----*
DATABASE   => SQLDBA
OWNER      => _____          PACKAGE-NAME => _____
OPTION     => _ (1=UNLOAD PACKAGE          2=RELOAD PACKAGE)
           (3=REBIND PACKAGE           4=VIEW PACKAGE)
*****          OPTIONAL PARAMETERS          *****
INVALID ONLY?      => 2 (1=YES/2=NO)
TO SERVER-NAME     => _____
REPLACE/NEW        => 1 (1=REPLACE/2=NEW)
KEEP/REVOKE        => 1 (1=KEEP/2=REVOKE)
*-----*          SQC16          *-----*

PRESS ENTER TO PROCESS
ENTER F1=HELP F3=EXIT

```

Figure 22. Package Utility Screen

## Package Utility Parameters

The Package Utility parameters are:

Entry Field	Description
<b>DATABASE</b>	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.
<b>OWNER</b>	the ID of the person who created this package or preprocessed the program associated with it.
<b>PACKAGE-NAME</b>	the name of the package.
<b>OPTION</b>	specifies the function to be executed
<b>INVALID ONLY?</b>	You may want to not 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).
<b>TO SERVER-NAME</b>	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.
<b>REPLACE/NEW</b>	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.
<b>KEEP/REVOKE</b>	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 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 22 on page 55). 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 23, is displayed. This is where you specify the parameters for job submission to the VSE/POWER queue.

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*----- PACKAGE JOB SUBMIT SCREEN -----*
JOBNAME => _____          CLASS => A
PRI      => 3                  DISP  => D (D,H,L,K)
FROM     => _____
DUETIME  => ____ (HHMM)       DUEDATE => ____ (AABBY)
DUEDAY   => _____
OTHER    => _____
LST CLASS => A
DEST     => _____
*----- SQC17 -----*

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

Figure 23. Package Job Submit Screen

### Package Utility Job Submit Parameters

The job submission parameters are:

<u>Parameter</u>	<u>Description</u>
<b>JOBNAME</b>	Specifies the name by which the DBSPACE REORGANIZATION job and its associated queue entries is to be known.
<b>CLASS</b>	Specifies the class or partition in which you want this job to run. Class defaults to A.
<b>PRI</b>	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.
<b>DISP</b>	Specifies how the job is to be handled in the reader queue. Disposition may be specified as: <ul style="list-style-type: none"> <li>• D - Delete after processing</li> </ul>

- H - Hold until released
- K - Keep after processing
- L - Leave in the queue

Disposition defaults to D.

<b>FROM</b>	Specifies the ID of the user being allowed to manipulate or retrieve the job. Defaults to the CICS user ID.
<b>DUETIME</b>	Specifies the processing start time using hh for hour and mm for minute in 24-hour clock time (OPTIONAL).
<b>DUEDATE</b>	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 (OPTIONAL).
<b>DUEDAY</b>	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 single quotes (apostrophes). 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 (OPTIONAL).
<b>OTHER</b>	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 (OPTIONAL).
<b>LST CLASS</b>	defines the class to be assigned to printed output. Specify any letter of the alphabet, or any number 0 through 9.
<b>DEST</b>	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 22 on page 55), 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.

## Package Utility

Figure 24 shows a report generated from the View Package option.

```
Date: dd Mmm yyyy                                     Page: 1
                                                    Control Center Package Report
-----
Package: SQLMSTR.SQC05
Database: SQLDBA

First Create Release: 7.1
Last Create Release: 7.1
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 24. Example Output from View Package Option

---

## Chapter 10. 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 Control Center DBA ID grants all authorizations. The tool records and maintains all authorizations.

You can use the “LIST Functions” on page 61 to view the various reports that are available to help manage database access. These 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 67 for an example.

The Control Center DBA ID owns five authorization tables that contain its information about User, Application Groups and authorizations. These tables are:

<b>USERID_GROUP_TAB</b>	User Group Table. This table is used to hold the name, internal ID, and description of a User group.
<b>USERID_TAB</b>	User ID Table. This table has one row for each unique combination of Userid and User Group ID.
<b>APPL_GROUP_TAB</b>	Application Group Table. This table is used to hold the name, internal ID, and description of an Application Group.
<b>OBJECT_TAB</b>	Object ID Table. This table has an entry for each unique combination of object (object owner, object name) and Application Group ID.
<b>GROUP_AUTH_TAB</b>	Group Authorization Table. This table records each group authorization made by the Control Center DBA. 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.

## Group Authorization Tool

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
- DROP a group
- Manage group objects and users
- Manage privileges
- Manage privileges
- LIST privileges

---

## Functions

### 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 users 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 users 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 in “DROP a Group” on page 60.)

### 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 Control Center DBA connect ID 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 (or Drop) objects to the Application Groups
3. Define User Groups
4. Add Users to the User Groups
5. Grant Authorities to the User Groups

In addition, you need to be able to:

6. Drop a User
7. Drop a Group

To reach the Group Authorization Menu shown in Figure 25:

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

## Group Authorization Tool

```
mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          GROUP AUTHORIZATION MENU  -----*
DATABASE => SQLDBA
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 25. 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 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          hh:mm:ss
*-----*          APPLICATION GROUP MENU  -----*
DATABASE   => SQLDBA
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=TABLES AND VIEWS, P=PACKAGES)

*-----*          SQC26 -----*

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

Figure 26. Control Center Application Group Menu

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 26 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 (or Drop) 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 Figure 26, 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 27 on page 64 shows how to add the three personnel tables and a view to the AGROUP1 Application Group.

If you later want to delete an object from an application group, use this function (accessed by Group Authorization, main menu option 3) to delete the object.

## Group Authorization Tool

```
mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          OBJECT FUNCTIONS          *-----*
DATABASE => SQLDBA
GROUP   => AGROUP1
OPTION  ==> 1

1= ADD OBJECTS          2= DROP OBJECTS

OWNER  OBJECT NAME
PERS   PERSTAB1
PERS   PERSTAB2
PERS   PERSTAB3
PERS   PERSVIEW1

*-----*          SQC28          *-----*

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

Figure 27. 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 28 on page 65 show how to add the UGROUP1 User Group.

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          USER GROUP MENU          -----*
DATABASE => SQLDBA
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 28. Adding a User Group

Figure 29 shows the result of using option 4 to see what groups already exist.

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          USER GROUP LIST          -----*
DATABASE => SQLDBA

GROUP  DESCRIPTION
TESTGRP TEST GROUP 1

-----*          SQC21 -----*

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

Figure 29. Checking for Existing 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 30 on page 66, or Option 2 from the Group Authorization Menu.

## Group Authorization Tool

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          ADD USERS          *-----*
DATABASE => SQLDBA
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 30. Adding Users to UGROUP1

## Step 5: Grant Authorities to the User Groups

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 31 shows how to grant Select and Update on AGROUP1 to UGROUP1.

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          AUTHORIZATIONS MENU          *-----*
DATABASE => SQLDBA

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 31. Authorization Menu

The choices made in the example show in Figure 31, 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 USER1, and 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

Using Control Center 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:

<p><b>Individual Tables</b> GRANT Select on SQLDBA.ACTIVITY to UGROUP1</p>
--

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. All grants and revokes are executed with the Control Center DBA connect ID.
5. When granting EXECUTE on packages, the Control Center DBA 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 Control Center DBA ID grants all privileges. As you build your authorization scheme and issue grants using the tool, remove old SQLMSTR grants (if you are migrating your Control Center to Version 7.3 or later), or non-Control Center DBA grants as appropriate.

## Group Authorization Tool

8. The following is an example of a query you can use to obtain 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, use:

```

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 32. 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 33. 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 11. The Monitor Maintenance Menu

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, on the operator's console, 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, on the operator's console, 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; such as, 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. For each database for which you have started a kernel, there will be a separate kernel task. 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 optionally 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 to check if it needs to dispatch any monitors.

### Options and Monitors Available

As shown in Figure 38 on page 75, 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 reports

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 COUNTER \*, 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 34 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:      SQLDBA
F2 0002 SQM0632 CMORGAN IS IN COMMUNICATION WAIT
F2 0002 SQM0318 DATABASE:      SQLDBA

==>

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 34. VSE Console - Monitor Threshold Notification

In this figure, Database SQLDBA 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 Main Menu, and then Option 1. It is shown in Figure 35. You must have defined (“Add a Monitor” on page 72) at least one monitor in order to use this function. Once started, the monitor kernel runs until you execute the Stop Monitor command.

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          MONITOR UTILITY          *-----*
DATABASE    =>  SQLDBA
*****
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 PROCESS
ENTER F1=HELP F3=EXIT
    
```

Figure 35. Monitor Menu

### Stop the Monitor Kernel

Lets you end the execution of all active monitors, once monitors currently executing have 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 36 on page 72 shows the monitors belonging to the SQLDBA database.

## The Monitor Maintenance Menu

```
mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          MONITOR LIST          *-----*
DATABASE => SQLDBA

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 36. 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. You can choose to update the frequency, start and stop times, and the monitor thresholds, and to activate or deactivate a monitor.

### Delete a Monitor

Lets you delete a monitor. To delete the SHOW LOG monitor from the SQLDBA database, for example, select Option **06**. Next select Monitor **04** and press ENTER.

Note that you can press PF3 to cancel 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 the monitor is reset.

## 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 when the job executes. Specify "1" to refresh the table, or "2" to show cumulative data.

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          MONITOR REPORT          *-----*
DATABASE    => SQLDBA_____          MONITOR    => DBSPACE00001
*****          REPORT PARAMETERS          *****
REPORT NAME    => _____
CLASS          => A
PRI            => 3
DISP          => D (D,H,L,K)
*****
RESET DATA?   => 1 (1=YES/2=NO)
*-----*          SQC26 -----*

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

Figure 37. Monitor Report Option

## Types of Monitors

### SHOW ACTIVE

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

## The Monitor Maintenance Menu

### SHOW CONNECT

Monitors users connected to the database. You can 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 and the database name.
2. Start a kernel for the database to which the monitor has been 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 (number) that you want to add.

Figure 38 on page 75 shows an example of adding the SHOW ACTIVE monitor for the SQLDBA database.

## The Monitor Maintenance Menu

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          MONITOR UTILITY          -----*
DATABASE    => SQLDBA

*****

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 38. 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 in the current database. You use this menu to make a specific monitor active or not. You also can 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          hh:mm:ss
*-----*          MONITOR MAINTENANCE      -----*
DATABASE          => SQLDBA          MONITOR          => ACTIVE
DESCRIPTION       => monitor our active users
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 39. 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 controlling the resetting of the data collected in the monitor table.

## Using the Monitor Maintenance Menu

You can specify:

## The Monitor Maintenance Menu

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. (It will be converted to all uppercase letters.)
ACTIVE?	Deactivates a monitor without having to delete the monitor definition. Use <b>1</b> to activate the monitor and <b>2</b> 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 <b>1</b> for YES and <b>2</b> 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. This is specified as HHMM, for example, 0001 for one minute after midnight. This value must be earlier than the STOP TIME value.
STOP TIME	Indicates when the monitor is to stop for the day(s) specified. This is specified as HHMM, for example, 2359 for one minute before midnight. This value must be later than the START TIME value.
INTERVAL	Indicates <b>approximately</b> 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. The actual interval may vary, depending upon the system loading and the time needed for the task to begin executing.
RESET DATA?	Deletes all rows of data collected by the monitor from the associated monitor table. Use <b>1</b> to delete all the rows and <b>2</b> to keep them.
RESET DAY	Used to indicate on what day to RESET DATA. The reset takes place at about 23:59. This value is ignored unless RESET DATA is 1.
PRINT REPORT?	Generates a monitor report and places it directly on the VSE/POWER List queue.
REPORT NAME	Specifies the name for the monitor report job and its associated queue entries.
CLASS	Specifies the class (partitions) 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 VSE 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 40 shows how to set the CHECKPOINT WAIT and USER WAIT thresholds for the SHOW ACTIVE monitor that belongs to database SQLDBA.

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          MONITOR THRESHOLDS      *-----*
DATABASE    => SQLDBA          MONITOR    => ACTIVITY
*****
CHECKPOINT WAIT          SHOW ACTIVE          *****
=> 1 (1=YES/2=NO)
USER WAIT              => 1 (1=YES/2=NO)
*****
AGENT NOT PROCESSING    SHOW CONNECT          *****
=> 2 (1=YES/2=NO)
INACTIVE              => 2 (1=YES/2=NO)
*****
CHECKPOINT              SHOW LOCK             *****
=> 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 40. 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 41 is a partial screen image showing how to set the **SHOW LOG** monitor to issue a VSE console message if the monitor detects a value of 75% or greater.

## The Monitor Maintenance Menu

```

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

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

```

Figure 41. SHOW LOG Threshold Specification

The fields displayed on the Monitor Threshold screen set thresholds for:

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 enable it. Choose Option **1, Start Kernel** from the Monitor Utility menu. A kernel must be "started" in each database for which any monitor is defined before that monitoring activity can take place.

### 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. Figure 42 on page 79 shows how to enter **2** in the ACTIVE? field to deactivate the SHOW ACTIVE monitor.

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          MONITOR MAINTENANCE      -----*
DATABASE            => SQLDBA                MONITOR            => ACTIVE
DESCRIPTION         => MONITOR OUR ACTIVE USERS
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 42. Updating (Deactivating) a Monitor

Figure 43 shows how to enter 2 to execute the SHOW ACTIVE monitor every two minutes instead of the five minutes originally defined in Figure 42.

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*          MONITOR MAINTENANCE      -----*
DATABASE            => SQLDBA                MONITOR            => ACTIVE
DESCRIPTION         => MONITOR OUR ACTIVE USERS
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            => 2                    (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 43. 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.

## The Monitor Maintenance Menu

mm/dd/yyyy		CONTROL CENTER MONITOR DATA										hh:mm:ss	
*-----*												*-----*	
DATABASE => SQLDBA		MONITOR => ACTIVE											
DATE	TIME	NACT	NIW	R/O	R/W	NEW	WAIT	WAIT	WAIT	PAGE	BLOCK	I/O	
-----		-----										-----	
2000-01-18	16.25.10	3	2	1	0	0	0	0	0	0	0	0	
2000-01-18	17.06.56	3	2	1	0	0	0	0	0	0	0	0	
2000-01-19	16.16.37	3	2	1	0	0	2	0	0	0	0	0	
2000-01-19	16.17.05	3	2	1	0	0	2	0	0	0	0	0	
2000-01-19	16.20.01	3	2	1	0	0	2	0	0	0	0	0	
2000-01-19	16.22.01	3	0	3	0	0	2	0	0	0	0	0	
2000-01-19	16.24.01	3	0	3	0	0	2	0	0	0	0	0	
2000-01-19	16.26.01	3	1	2	0	0	2	0	0	0	0	0	
2000-01-19	16.28.01	3	1	2	0	0	2	0	0	0	0	0	
2000-01-19	16.49.18	3	2	1	0	0	2	0	0	0	0	0	
2000-01-19	16.55.18	3	1	2	0	0	2	0	0	0	0	0	
2000-01-19	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 44. 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 in a specific database 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 all monitors, stopped due to the kernel being stopped, select the START KERNEL from the Monitor Utility menu, or to restart a single monitor, reactivate it from the Monitor Maintenance menu. For a monitor to be active, a kernel must be running in its database.

---

## Chapter 12. Table Utility

The Table Utility provides an easy way for you to create a table or 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.
- Update statistics.

The Table Utility consists of a set of programs, a batch DDL generator program, and batch jobstreams that execute Database Services Utility (DBSU) commands.

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 a table to a different DBSPACE or Server
  - 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.

## Table Utility

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

```
mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
-----*-----
          TABLE UTILITY -----*-----
DATABASE      => SQLDBA

*****

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 45. Table Utility Main Menu

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

- “LIST TABLES” on page 83
- “DROP TABLE” on page 84
- “REORGANIZE TABLE” on page 85
- “CREATE TABLE” on page 93
- “UPDATE STATISTICS” on page 98

If Option 1 is selected, several choices exist:

- If CREATOR and TABLE are left blank, then **all** tables in the database will be listed.
- If one of the CREATOR or TABLE 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 you can 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 have been 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, the Control Center main menu or CICS.

Functions 2 through 5 produce a job that is submitted to VSE/POWER for execution. If you code DISP=H on the job submission screens, then you can view, and if necessary, alter the job for your particular environment.

## 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 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 Search 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                hh:mm:ss
*-----*
*-----*
DATABASE => SQLDBA
*-----*
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
*-----*
*-----*
SELECT A TABLE AND PRESS ENTER TO RETURN
F1=HELP F3=EXIT F4=TOP F5=BOT F7=BWD F8=FWD F12=CANCEL
*-----*
SQC63

```

Figure 46. 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, 3, and 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. If you press F12, control returns to the Table Utility main menu, without changing anything that was originally on that menu.

The Table List menu (Figure 46) displays these table attributes from SYSTEM.SYSCATALOG:

Column Label	Content
--------------	---------

## Table Utility

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

```
mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*-----*
DATABASE    => SQLDBA
*****
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 47. Table Utility Main Menu with Prompt for Verifying a Drop Request

## 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 drop is successful, message 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 48 on page 86.

You can choose one of the following options:

### 1 - GENERATE DDL

Generates from the database all of the DDL required to recreate the table and the objects it contains, including indexes, views, and grants. The DDL is placed in the punch queue for inspection, alteration, or later use. See Figure 50 on page 91.

### 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 61 on page 120.

### 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 included. An example of a REORGANIZE TABLE job is in Figure 62 on page 121.

### 4 - RELOAD TABLE

Submits a job to recreate and reload a table that has been unloaded by Option 2. This is basically a table recovery facility. An example of the job created to do this is in Figure 63 on page 124.

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

## Table Utility

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*----- TABLE REORGANIZATION UTILITY -----*
DATABASE => SQLDBA _____
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 => : 2 (1/2)
COMMITCOUNT  => _____          DATA CAPTURE           => : none
TLBL FILE-ID  => _____

*----- SQC61 -----*

PRESS ENTER TO PROCESS
F1=HELP F3=EXIT

```

Figure 48. Table Reorganization Screen

The following is a summary of the various processing options:

1. 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.
2. 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.
3. 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:

  - a. generate the DDL for recreating the table and its objects,
  - b. unload the data to a VSAM file,
  - c. copy the VSAM file to a SAM file,
  - d. drop the table,
  - e. recreate the table,
  - f. reload the table,
  - g. and generate the table objects, such as indexes, etc.

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 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 to 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.

4. 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 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.

## Table Utility

### 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. Referential Integrity Considerations:
  - If the table being reorganized has a deactivated primary key, any dependent tables (that is, those with foreign keys) will not be able to have their foreign keys created.
  - If the table being reorganized has an **implicitly** deactivated foreign key, it cannot be created.
  - If the table being reorganized has an **explicitly** deactivated foreign key, it can be created, then deactivated, **IF** the parent table's primary key is active.
  - If the table being reorganized has an active primary key, any dependent table with an explicitly deactivated foreign key **may** be able to have its foreign key created **AFTER** the primary key table is loaded. The create will fail if the dependent table's data is not correct. If the create is successful, the key will then be deactivated.

## Table Reorganization Menu Required Parameters

To reach the Table Reorganization menu shown in Figure 48 on page 86, 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 one to 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.

### 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 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 2,147,483,647 (without the commas) 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 Chapter 6, "Work File Label Definition Tool" on page 39. This field does not apply to DDL because DDL is **always** unloaded to disk.

## Table Utility

### DATA CAPTURE

Allows changing or adding the attribute to a table for DB2 Release 5.1 (or later) 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 49 is an example of the jobstream produced by Control Center to generate DDL for the SQLDBA.ACTIVITY table.

```
* $$ JOB JNM=GENDDL,CLASS=0,DISP=D,PRI=9
* $$ LST CLASS=A,PRI=3
* $$ PUN PRI=3
// JOB TRAY2 GENERATE DDL FOR TABLE "SQLDBA"."ACTIVITY"
// OPTION LOG
* * * * *
* STEP0001 UNLOAD DDL
* * * * *
// DLBL SQMTPRM,'SQLMSTR.TABLE.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMDDL,'L.DB2730.EXAMPLE.ROUTINE',0,VSAM, X
RECORDS=001000,RECSIZE=80,DISP=(NEW,KEEP),CAT=SQMCAT
// ASSGN SYS005,YSRDR
// ASSGN SYS006,SYSPCH
// ASSGN SYS011,SYSLST
// EXEC SQB60,SIZE=AUTO
%SQLDBA          SQLDBA  ACTIVITY          1 N SQLDBA  SQLDBAPW
/*
/&
```

Figure 49. 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 61 on page 120 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 62 on page 121 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 63 on page 124 shows a jobstream generated by Control Center to reload the SQLDBA.ACTIVITY table.

**Table Reorganization Submit Screen**

Figure 50 shows the Table Reorganization Submit Screen.

```

mm/dd/yyyy          CONTROL CENTER          hh:mm:ss
*----- TABLE REORGANIZATION SUBMIT SCREEN -----*
JOBNAME  => _____          CLASS    => A
PRI      => 3                    DISP    => D (D,H,K,L)
FROM     => _____          LST CLASS => A
DUETIME  => ____ (HHMM)        DUEDATE  => ____ (AABBY)
DUEDAY   => _____
OTHER    => _____

*----- SQC62 -----*

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

Figure 50. Table Reorganization Submit Screen

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

**Table Reorganization Job Submission Screen 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.

## Table Utility

### 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 VSE/POWER \$\$ LST card. Specify any letter of the alphabet, or any number 0 through 9. This parameter is required; its default value is **A**.

### DUETIME

Specifies the job processing start time using **HH** for hour and **MM** for minute in 24-hour clock time (OPTIONAL).

### 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 (OPTIONAL).

### 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 number 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 (OPTIONAL).

If you schedule repetitive processing, you can cancel the job whenever necessary by displaying the POWER READER queue and deleting the job.

**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 (OPTIONAL).

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 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 DB2 tables and having them created by the Table Utility. You are presented with a Create Table screen (see Figure 51).

```

mm/dd/yyyy                CONTROL CENTER                hh:mm:ss
*----- CREATE TABLE -----*
DATABASE => SQLDBA
CREATOR  => SQLDBA                OWNER      => _____
TABLE    => ACTIVITY2             DBSPACE   => _____

COL CNAME          COLTYPE          LENGTH NULLS (1/2)
001 _____          _____          _____ -
002 _____          _____          _____ -
003 _____          _____          _____ -
004 _____          _____          _____ -
005 _____          _____          _____ -
006 _____          _____          _____ -
007 _____          _____          _____ -
008 _____          _____          _____ -
009 _____          _____          _____ -
010 _____          _____          _____ -
*****
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 51. 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 (CNAME), column type (COLTYPE), and if required by the column type, a length (default lengths are described just below).

## Table Utility

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 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 92, 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 53 on page 97) 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 overwrite 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 overwrite existing fields that you want to change, and, finish any partially complete entries; over-type changes are automatically recognized by Control Center.

Pressing ENTER after INS or DEL displays the changed screen. You must press ENTER again if you are finished with the table's definition.

**Note:** 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 or pressing ENTER 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, When 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.

## Table Utility

### 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.

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

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

#### 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 remove the job from the READER queue, edit it, add the additional parameters, and resubmit 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":

#### 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 (OPTIONAL).

#### 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 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. It is converted to all uppercase letters.

Figure 52 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                hh:mm:ss
*----- CREATE TABLE -----*
DATABASE => SQLDBA
CREATOR  => SQLDBA                OWNER      => owner
TABLE    => ACTIVITY2            DBSPACE   => dbspacename

COL CNAME      COLTYPE      LENGTH NULLS (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 52. Create Table Data Entry Screen

When you press ENTER, the Create Table Submit screen is displayed; see Figure 53.

```

mm/dd/yyyy                CONTROL CENTER                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 53. Create Table Submit screen

The first parameter, JOBNAME, is the only one that is required because the other parameters have default values.

## Table Utility

For a detailed description of the job submission parameters, see the description of the job submission parameters on page 91.

**Note:** In the discussion of the “job submission DISP parameter” on page 92, be sure to note that using DISP=K as a means of recovering from some of the errors you might make when creating a table.

A job return code of 6 is normal.

---

## UPDATE STATISTICS

This function provides you with the capability of issuing the UPDATE STATISTICS or UPDATE ALL STATISTICS commands in a background DBSU job.

When you select Option 5, Update Statistics, on the Table Utility main menu, the Update Statistics Submit screen is displayed (Figure 54).

```
mm/dd/yyyy          CONTROL CENTER          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 54. 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 91.

### Update Statistics Submit Screen Optional Parameters

**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

## Table Utility

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.





---

## Chapter 13. Installing Stored Procedures Support

DB2 UDB's Web Control Center, running on your workstation, can be used to perform certain tasks using a DB2 Server for VSE & VM database running on a "host" computer. Control Center for VSE and VM provides stored procedures that are used to do that type of processing. The stored procedures are invoked by UDB's Web Control Center to process DB2 commands such as Reorganize Index and Rebind Package.

Web Control Center stored procedures support requires Version 7 or greater of DB2 UDB, DB2 for VSE, and Control Center for VSE. If you are planning to use DB2 Web Control Center, then perform the process described below. If you will not be installing DB2 Web Control Center, then the following process is not needed.

This optional procedure is used to install two IBM-supplied stored procedures. One can be used to Reorganize an Index and to Rebind a Package; the other is used as part of the Visual Explain services.

To use stored procedures, you must define a stored procedure server (part of your DB2 installation and tailoring process) and then inform DB2 about the stored procedures you want to execute. The stored procedures must be accessible to the stored procedure server at execution time. Stored procedures are explained in detail in the *DB2 Server for VSE & VM Database Administration* manual.

In this step, you will:

1. Define the stored procedures to DB2. (Step 1 in SQMSPRC1)
2. Linkedit the procedures into the Control Center library. A linkedit is necessary because the Language Environment (LE) used by the Control Center development team may be different than that you are using. (Step 2 in SQMSPRC1)
3. Load the stored procedure packages into the databases that will use them. (Step 3 in SQMSPRC1)
4. Make the Control Center library accessible to the stored procedure server.

To install the support, punch SQMSPRC1.Z from your Control Center installation library, tailor it for your installation and submit it for execution.

**Note:** The SQMSPRC1 job must be re-run for each database that will use these stored procedures. Each time you re-run it, the stored procedures' phases will be linkedited into the Control Center library again although the rest of the job will involve a different database each time. Setting up the job in this manner involves a little extra CPU time, but simplifies your installation process.

Executing this process does not START the stored procedure server nor any stored procedures; it only enables those actions.

There are a number of stored procedure server initialization procedures that are required, including previously defining a stored procedure server. See the *DB2 Server for VSE & VM Database Administration* manual for information about these.

To complete this step:

## Stored Procedures Support

1. Punch SQMSPRC1.Z and edit.
2. Modify EACH procedure used to start a stored procedure server so that it can refer to your Control Center library. The sample procedure is in the *DB2 Server for VSE & VM Database Administration* manual which discusses Stored Procedures.

```
$ $$ JOB JNM=SQMSPRC1,CLASS=0,DISP=D,PRI=9
$ $$ LST CLASS=Q
// JOB SQMSPRC1          INSTALL THE STORED PROCEDURES    04/22/2000
// LIBDEF *,SEARCH=PRD2.DB2710
*
* *****
*
* NOTE: YOU MUST HAVE AT LEAST ONE PSERVER DEFINED TO DB2 BEFORE
*       RUNNING THIS PROCESS!
*
* *****
*
* CHANGE ALL PARAMETERS ENCLOSED IN <> AND REMOVE THE <>.
* THE PARAMETERS ARE:
*   DATABASE NAME                      (4X)
*   SQLDBA PASSWORD                     (2X)
*   CONTROL CENTER LIBRARY:  WHERE CONTROL CENTER IS INSTALLED (2X)
*   STORED PROCEDURE LIBRARY:  WHERE THE EXECUTABLE MODULES FOR (1X)
*                               THE STORED PROCEDURES WILL RESIDE
*
```

Figure 55. Install Stored Procedures (SQMSPRC1.Z) (Part 1 of 3)

```

* *****
*
* STEP 1: DEFINE THE STORED PROCEDURES
*
* *****
*
// EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(<DATABASE NAME>)'
CONNECT SQLDBA IDENTIFIED BY <SQLDBA PASSWORD> TO <DATABASE NAME>
CREATE PROCEDURE ARISCCF
    (IN FUNCODE INT,
     IN OWNER CHAR(8),
     IN PARM1 CHAR(18),
     IN PARM1LN INT,
     IN PARM2 INT,
     OUT RETCODE INT )
LANGUAGE PLI,
EXTERNAL,
SERVER GROUP,
GENERAL,
RESULT SET 0;
CREATE PROCEDURE ARISVEF
    (IN CREATOR CHAR(8),
     IN PKNAME CHAR(8),
     OUT RETCODE INT )
LANGUAGE PLI,
EXTERNAL,
SERVER GROUP,
GENERAL,
RESULT SET 0;
#*
// IF $RC > 0 THEN
// GOTO $EOJ
* *****
*
* STEP 2. LINK EDIT THE STORED PROCEDURES
*
* *****
*
// OPTION CATAL
// LIBDEF *,SEARCH=(<CONTROL CENTER LIBRARY>,PRD2.DB2710)
// LIBDEF PHASE,CATALOG=<STORED PROCEDURE LIBRARY>
// PHASE ARISCCF,*
// INCLUDE ARISCCF
// EXEC LNKEDT,PARM='MSHP'
// IF $RC > 4 THEN
// GOTO $EOJ
// PHASE ARISVEF,*
// INCLUDE ARISLKSV
// EXEC LNKEDT,PARM='MSHP'
// IF $RC > 4 THEN
// GOTO $EOJ

```

Figure 55. Install Stored Procedures (SQMSPRC1.Z) (Part 2 of 3)

```

* *****
*
* STEP 3. LOAD THE STORED PROCEDURE PACKAGE INTO THE DATABASE
*
* *****
*
// LIBDEF *,SEARCH=(<CONTROL CENTER LIBRARY>,PRD2.DB2710)
// OPTION LOG
// EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(<DATABASE NAME>)'
CONNECT SQLDBA IDENTIFIED BY <SQLDBA PASSWORD> TO <DATABASE NAME>
COMMENT '***** RELOAD PACKAGE SQLDBA.ARIS7VE *****'
RELOAD PACKAGE (SQLDBA.ARIS7VE)
    REPLACE
    KEEP
INFILE(SYSIPT BLKSZ(80) PDEV(DASD));
READ MEMBER ARIS7VE.Q (NOCONT
#*
#&
$$$ E0J

```

Figure 55. Install Stored Procedures (SQMSPRC1.Z) (Part 3 of 3)

## Appendix A. Reorganization Job Streams

```

* $$ JOB JNM=DBUNLOAD,CLASS=0,DISP=D
* $$ LST PRI=3
// JOB DBUNLOAD 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,YSRDR
// ASSGN SYS011,SYLST
// EXEC SQB02,SIZE=AUTO
%%SQLDS730          PUBLIC  SQMHELP          2NN1 SQLDBA  SQLDBAPW
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * *
* STEP0002 UNLOAD DDL
* * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMDDL,'L.SQLDS730.PUBLIC.SQMHELP',0,VSAM,
RECORDS=001000,RECSIZE=80,DISP=(NEW,KEEP),CAT=SQMCAT
// ASSGN SYS005,YSRDR
// ASSGN SYS011,SYLST
// EXEC SQB01,SIZE=AUTO
%%SQLDS730          PUBLIC  SQMHELP          2 N SQLDBA  SQLDBAPW
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * *
* STEP0003 UNLOAD DBSPACE
* * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,'VSEPK3',1,0,0000423488,22000
// ASSGN SYS007,DISK,VOL='VSEPK3',SHR
// EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(SQLDS730)'
CONNECT SQLDBA IDENTIFIED BY SQLDBAPW
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 56. DBSPACE Reorg Option 2 (Unload DBSPACE) - Sample Jobstream (Part 1 of 2)

## Reorganization Job Streams

```
*****
* STEP0004 RECORD TIME AFTER UNLOAD
*****
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%SQLDS730      PUBLIC  SQMHELP      2NN2 SQLDBA  SQLDBAPW
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
*****
* STEP0005 REPRO SAM DATA TO VSAM
*****
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,'VSEPK3',1,0,0000423488,22000
// ASSGN SYS007,DISK,VOL='VSEPK3',SHR
// DLBL SQMDAT,'D.SQLDS730.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)))
/*
/&
```

Figure 56. DBSPACE Reorg Option 2 (Unload DBSPACE) - Sample Jobstream (Part 2 of 2)

```

* $$ JOB JNM=DBREORG,CLASS=0,DISP=D
* $$ LST PRI=3
// JOB DBREORG MUM REORG DBSPACE - 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,YSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%SQLDS730          PUBLIC  SQMHELP          3NN1 SQLDBA  SQLDBAPW
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
*****
* STEP0002 UNLOAD DDL
*****
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMDDL,'L.SQLDS730.PUBLIC.SQMHELP',0,VSAM,
RECORDS=001000,RECSIZE=80,DISP=(NEW,KEEP),CAT=SQMCAT
// ASSGN SYS005,YSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB01,SIZE=AUTO
%%SQLDS730          PUBLIC  SQMHELP          3NN  SQLDBA  SQLDBAPW
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
*****
* STEP0003 UNLOAD DBSPACE
*****
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,'VSEPK3',1,0,0000423488,22000
// ASSGN SYS007,DISK,VOL='VSEPK3',SHR
// EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(SQLDS730)'
CONNECT SQLDBA IDENTIFIED BY SQLDBAPW
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,YSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%SQLDS730          PUBLIC  SQMHELP          3NN2 SQLDBA  SQLDBAPW
/*
// IF $RC > 0000 THEN
// GOTO $EOJ

```

Figure 57. DBSPACE Reorg Option 3 (Reorganize DBSPACE) - Sample Jobstream (Part 1 of 3)

## Reorganization Job Streams

```

* * * * *
* STEP0005 REPRO VSAM DDL TO SAM
* * * * *
// DLBL VSAMIN,'L.SQLDS730.PUBLIC.SQMHELP',,VSAM, X
      CAT=SQMCAT,DISP=(OLD,KEEP)
// DLBL SQMDDL1,'SQLMSTR.DDL.FILE1',0,SD
// EXTENT SYS008,'VSEPK3',1,0,0000445488,00600
// 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
/*
* * * * *
* STEP0006 RECORD TIME BEFORE RELOAD
* * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,YSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%SQLDS730          PUBLIC  SQMHELP          3NN3 SQLDBA  SQLDBAPW
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * *
* STEP0007 RELOAD DBSPACE
* * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,'VSEPK3',1,0,0000423488,22000
// ASSGN SYS007,DISK,VOL='VSEPK3',SHR
// ASSGN SYS004,SYS005
// ASSGN SYS020,SYSIPT
// DLBL IJSYSIN,'SQLMSTR.DDL.FILE1',0,SD
// EXTENT SYSIPT,'VSEPK3',1,0,0000445488,00600
ASSGN SYSIPT,DISK,VOL='VSEPK3',SHR
// ON $ABEND GOTO CLOSEIPT
// ON $CANCEL GOTO CLOSEIPT
// EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(SQLDS730)'
/*
// 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,YSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%SQLDS730          PUBLIC  SQMHELP          3NN4 SQLDBA  SQLDBAPW
/*
// IF $RC > 0000 THEN
// GOTO $EOJ

```

Figure 57. DBSPACE Reorg Option 3 (Reorganize DBSPACE) - Sample Jobstream (Part 2 of 3)



## Reorganization Job Streams

```

* * * * *
* STEP0009 DELETE VSAM DDL FILE
* * * * *
// ASSGN SYSLST,IGN
// DLBL FILEIN,'L.SQLDS730.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,'VSEPK3',1,0,0000423488,22000
// ASSGN SYS007,DISK,VOL='VSEPK3',SHR
// DLBL SQMDAT,'D.SQLDS730.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
* * * * *
/*
/&

```

Figure 57. DBSPACE Reorg Option 3 (Reorganize DBSPACE) - Sample Jobstream (Part 3 of 3)

## Reorganization Job Streams

```

* $$ JOB JNM=RELOAD,CLASS=0,DISP=D,NTFY=YES
* $$ LST PRI=3
// JOB RELOAD MUM RELOAD DBSPACE FROM DISK
// OPTION LOG
* * * * *
* STEP0001 CHECK FOR DATA FILE
* * * * *
// ASSGN SYSLST,IGN
// DLBL FILEIN,'D.SQDLS730.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.SQDLS730.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.SQDLS730.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
%%SQDLS730          PUBLIC  SQMHELP          4NN4 SQLDBA  SQLDBAPW
/*
// IF $RC > 0000 THEN
// GOTO $EOJ

```

Figure 58. DBSPACE Reorg Option 4 (Reload DBSPACE) - Sample Jobstream (Part 1 of 3)

```

* * * * *
* STEP0005 REPRO VSAM DDL TO SAM
* * * * *
// DLBL VSAMIN,'L.SQLDS730.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
* * * * *
// 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
// ON $ABEND GOTO CLOSEIPT
// ON $CANCEL GOTO CLOSIPT
// EXEC ARIDBS,SIZE=AUTO,PARM='DBNAME(SQLDS730)'
/*
// 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,SYSDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%SQLDS730          PUBLIC  SQMHELP          4NN4  SQLDBA  SQLDBAPW
/*
// GOTO THEEND
/. CLOSEIPT
* * * * *
* STEP0008 CLOSE SYSIPT
* * * * *
CLOSE SYSIPT,SYS020
// GOTO $EOJ
/. NODAT
// RESET SYSLST
* * * * *
* STEP0009 NO DATA FILE
* * * * *
* RELOAD ABORTED: MISSING FILE = D.SQLDS730.PUBLIC.SQMHELP
// GOTO $EOJ
/. NODDL
// RESET SYSLST

```

Figure 58. DBSPACE Reorg Option 4 (Reload DBSPACE) - Sample Jobstream (Part 2 of 3)

## Reorganization Job Streams

```

* * * * *
* STEP0010 NO DDL FILE
* * * * *
* RELOAD ABORTED: MISSING FILE = L.SQDLS730.PUBLIC.SQMHELP
/. THEEND
* * * * *
* STEP0011 THE END
* * * * *
/*
/&

```

Figure 58. DBSPACE Reorg Option 4 (Reload DBSPACE) - Sample Jobstream (Part 3 of 3)

```

* $$ JOB JNM=SUMREORG,CLASS=A,DISP=H,PRI=3
* $$ LST PRI=3
// JOB SUMREORG SUM REORG DBSPACE - 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 ARISQDLS,SIZE=AUTO,PARM='SYSMODE=S,LOGMODE=N,DBNAME=SQLDBA,PROGX
NAME=SQB02,NDIRBUF=000010,NPAGBUF=000010'
%%SQLDBA PUBLIC SAMPLE 3NY1 SQLDBA SQLDBAPW
/*
// IF $RC > 0000 AND $RC < 4095 THEN
// GOTO $EOJ
* * * * *
* STEP0002 UNLOAD DDL
* * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMDDL,'L.SQLDBA.PUBLIC.SAMPLE',0,VSAM, X
RECORDS=000500,RECSIZE=80,DISP=(NEW,KEEP),CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// ON $RC = 4095 CONTINUE
// EXEC ARISQDLS,SIZE=AUTO,PARM='SYSMODE=S,LOGMODE=N,DBNAME=SQLDBA,PROGX
NAME=SQB01,NDIRBUF=000010,NPAGBUF=000010'
%%SQLDBA PUBLIC SAMPLE 3NY SQLDBA SQLDBAPW
/*
// IF $RC > 0000 AND $RC < 4095 THEN
// GOTO $EOJ
* * * * *
* STEP0003 UNLOAD DBSPACE
* * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,'VSEPK3',1,0,0000423488,22000
// ASSGN SYS007,DISK,VOL='VSEPK3',SHR
// EXEC ARISQDLS,SIZE=AUTO,PARM='SYSMODE=S,LOGMODE=N,DBNAME=SQLDBA,PROGX
NAME=ARIDBS,NDIRBUF=000010,NPAGBUF=000010'
CONNECT SQLDBA IDENTIFIED BY SQLDBAPW;

```

Figure 59. Sample Single User Mode DBSPACE Reorganization Jobstream (Part 1 of 4)

```

COMMENT '***** UNLOAD DBSPACE "PUBLIC"."SAMPLE" *****'
SET UPDATE STATISTICS OFF;
UNLOAD DBSPACE ("PUBLIC"."SAMPLE")
OUTFILE(SQMDAT1 BLKSZ(02048) PDEV(DASD))
/*
// IF $RC > 0000 AND $RC < 4095 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,YSRDR
// ASSGN SYS011,YSLST
// EXEC SQB02,SIZE=AUTO
%%SQLDBA          PUBLIC  SAMPLE          3NY2 SQLDBA  SQLDBAPW
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * *
* STEP0005 REPRO VSAM DDL TO SAM
* * * * *
// DLBL VSAMIN,'L.SQLDBA.PUBLIC.SAMPLE',,VSAM,          X
          CAT=SQMCAT,DISP=(OLD,KEEP)
// DLBL SQMDDL1,'SQLMSTR.DDL.FILE1',0,SD
// EXTENT SYS008,'VSEPK3',1,0,0000445488,00600
// 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
/*
* * * * *
* STEP0006 RECORD TIME BEFORE RELOAD
* * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,YSRDR
// ASSGN SYS011,YSLST
// EXEC SQB02,SIZE=AUTO
%%SQLDBA          PUBLIC  SAMPLE          3NY3 SQLDBA  SQLDBAPW
/*
// IF $RC > 0000 THEN
// GOTO $EOJ

```

Figure 59. Sample Single User Mode DBSPACE Reorganization Jobstream (Part 2 of 4)

## Reorganization Job Streams

```

* * * * *
* STEP0007 RELOAD DBSPACE
* * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,'VSEPK3',1,0,0000423488,22000
// ASSGN SYS007,DISK,VOL='VSEPK3',SHR
// ASSGN SYS004,SYS005
// ASSGN SYS020,SYSIPT
// DLBL IJSYSIN,'SQLMSTR.DDL.FILE1',0,SD
// EXTENT SYSIPT,'VSEPK3',1,0,0000445488,00600
ASSGN SYSIPT,DISK,VOL='VSEPK3',SHR
// ON $ABEND GOTO CLOSEIPT
// ON $CANCEL GOTO CLOSEIPT
// EXEC ARISQLDS,SIZE=AUTO,PARM='SYSMODE=S,LOGMODE=N,DBNAME=SQLDBA,PROGX
      NAME=ARIDBS,NDIRBUF=000010,NPAGBUF=000010'

/*
// IF $RC > 0006 AND $RC < 4095 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
// ON $RC = 4095 CONTINUE
// EXEC ARISQLDS,SIZE=AUTO,PARM='SYSMODE=S,LOGMODE=N,DBNAME=SQLDBA,PROGX
      NAME=SQB02,NDIRBUF=000010,NPAGBUF=000010'
%%SQLDBA      PUBLIC SAMPLE      3NY4 SQLDBA  SQLDBAPW
/*
// IF $RC > 0000 AND $RC < 4095 THEN
// GOTO $EOJ
* * * * *
* STEP0009 DELETE VSAM DDL FILE
* * * * *
// ASSGN SYSLST,IGN
// DLBL FILEIN,'L.SQLDBA.PUBLIC.SAMPLE',,VSAM,
      CAT=SQMCAT,DISP=(OLD,DELETE) X
// EXEC IDCAMS,SIZE=AUTO
      PRINT INFILE(FILEIN) -
      COUNT(1)
/*
// RESET SYSLST
// GOTO THEEND
/. CLOSEIPT

```

Figure 59. Sample Single User Mode DBSPACE Reorganization Jobstream (Part 3 of 4)

```

* * * * *
* STEP0010 CLOSE SYSIPT
* * * * *
CLOSE SYSIPT,SYS020
* * * * *
* STEP0011 REPRO SAM DATA TO VSAM
* * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,'VSEPK3',1,0,0000423488,22000
// ASSGN SYS007,DISK,VOL='VSEPK3',SHR
// DLBL SQMDAT,'D.SQDBA.PUBLIC.SAMPLE',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 59. Sample Single User Mode DBSPACE Reorganization Jobstream (Part 4 of 4)

## Reorganization Job Streams

```

$$$ JOB JNM=TAPREORG,CLASS=0,DISP=D,PRI=9
* $$$ LST PRI=3
// JOB TAPREORG SUM REORG DBSPACE - DISK
// OPTION LOG
* * * * *
* STEP0001 RECORD TIME BEFORE DDL GENERATION
* * * * *
// TLBL ARIARCH
// 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=A,DBNAME=DB2710,PROGX
        NAME=SQB02,NDIRBUF=000010,NPAGBUF=000015'
%%DB2710          PUBLIC SAMPLE          3NY1 SQLDBA  SQLDBAPW
/*
// IF $RC > 0000 AND $RC < 4095 THEN
// GOTO $EOJ
* * * * *
* STEP0002 UNLOAD DDL
* * * * *
// TLBL ARIARCH
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMDDL,'L.DB2710.PUBLIC.SAMPLE',0,VSAM,                                X
        RECORDS=001000,RECSIZE=80,DISP=(NEW,KEEP),CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// ON $RC = 4095 CONTINUE
// EXEC ARISQLDS,SIZE=AUTO,PARM='SYSMODE=S,LOGMODE=A,DBNAME=DB2710,PROGX
        NAME=SQB01,NDIRBUF=000010,NPAGBUF=000015'
%%DB2710          PUBLIC SAMPLE          3NY SQLDBA  SQLDBAPW
/*
// IF $RC > 0000 AND $RC < 4095 THEN
// GOTO $EOJ
* * * * *
* STEP0003 UNLOAD DBSPACE
* * * * *
// TLBL ARIARCH
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,'VSEPK3',1,0,0000201438,06300
// ASSGN SYS007,DISK,VOL='VSEPK3',SHR
// EXEC ARISQLDS,SIZE=AUTO,PARM='SYSMODE=S,LOGMODE=A,DBNAME=DB2710,PROGX
        NAME=ARIDBS,NDIRBUF=000010,NPAGBUF=000015'
CONNECT SQLDBA IDENTIFIED BY SQLDBAPW;
COMMENT '***** UNLOAD DBSPACE "PUBLIC"."SAMPLE" *****'
SET UPDATE STATISTICS OFF;
UNLOAD DBSPACE ("PUBLIC"."SAMPLE")
OUTFILE(SQMDAT1 BLKSZ(02048) PDEV(DASD))
/*
// IF $RC > 0000 AND $RC < 4095 THEN
// GOTO $EOJ

```

Figure 60. Sample Single User Mode DBSPACE Reorganization using Tape (Part 1 of 4)



```

* * * * *
* STEP0004 RECORD TIME AFTER UNLOAD
* * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,YSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%DB2710          PUBLIC  SAMPLE          3NY2 SQLDBA  SQLDBAPW
/*
// IF $RC > 0000 THEN
// GOTO $EOJ
* * * * *
* STEP0005 REPRO VSAM DDL TO SAM
* * * * *
// DLBL VSAMIN,'L.DB2710.PUBLIC.SAMPLE',,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
/*
* * * * *
* STEP0006 RECORD TIME BEFORE RELOAD
* * * * *
// DLBL SQMPARM,'SQLMSTR.REORG.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMRDAT,'SQLMSTR.REORG.DATA',,VSAM,CAT=SQMCAT
// ASSGN SYS005,YSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB02,SIZE=AUTO
%%DB2710          PUBLIC  SAMPLE          3NY3 SQLDBA  SQLDBAPW
/*
// IF $RC > 0000 THEN
// GOTO $EOJ

```

Figure 60. Sample Single User Mode DBSPACE Reorganization using Tape (Part 2 of 4)

## Reorganization Job Streams

```

* * * * *
* STEP0007 RELOAD DBSPACE
* * * * *
// TLBL ARIARCH
// 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 ARISQLDS,SIZE=AUTO,PARM='SYSMODE=S,LOGMODE=A,DBNAME=DB2710,PROGX
        NAME=ARIDBS,NDIRBUF=000010,NPAGBUF=000015'

/*
// IF $RC > 0006 AND $RC < 4095 THEN
// GOTO CLOSEIPT
CLOSE SYSIPT,SYS020
* * * * *
* STEP0008 RECORD TIME AFTER RELOAD
* * * * *
// TLBL ARIARCH
// 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=A,DBNAME=DB2710,PROGX
        NAME=SQB02,NDIRBUF=000010,NPAGBUF=000015'

%%DB2710          PUBLIC  SAMPLE          3NY4
/*
// IF $RC > 0000 AND $RC < 4095 THEN
// GOTO $EOJ
* * * * *
* STEP0009 DELETE VSAM DDL FILE
* * * * *
// ASSGN SYSLST,IGN
// DLBL FILEIN,'L.DB2710.PUBLIC.SAMPLE',,VSAM,
        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

```

Figure 60. Sample Single User Mode DBSPACE Reorganization using Tape (Part 3 of 4)

## Reorganization Job Streams

```
*****
* STEP0011 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.DB2710.PUBLIC.SAMPLE',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
*****
```

Figure 60. Sample Single User Mode DBSPACE Reorganization using Tape (Part 4 of 4)

## Reorganization Job Streams

```

* $$ JOB JNM=TBUNLOAD,DISP=D
* $$ LST CLASS=A,PRI=3
// JOB TBUNLOAD UNLOAD TABLE "EXAMPLE"."ROUTINE" - DISK
// OPTION LOG
* * * * *
* STEP0001 UNLOAD DDL
* * * * *
// DLBL SQMTPRM,'SQLMSTR.TABLE.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMDDL,'L.DB2710.EXAMPLE.ROUTINE',0,VSAM,
RECORDS=001000,RECSIZE=80,DISP=(NEW,KEEP),CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB60,SIZE=AUTO
%%DB2710          EXAMPLE ROUTINE          2 N SQLDBA  SQLDBAPW
/*
// 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(DB2710)'
CONNECT SQLDBA IDENTIFIED BY SQLDBAPW;
COMMENT '***** UNLOAD TABLE "EXAMPLE"."ROUTINE" *****'
SET UPDATE STATISTICS OFF;
UNLOAD TABLE ("EXAMPLE"."ROUTINE")
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.DB2710.EXAMPLE.ROUTINE',0,VSAM,
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)))
/*
/&

```

Figure 61. TABLE Reorg Option 2 (Unload TABLE) - Sample Jobstream

```

* $$ JOB JNM=TBREORG,CLASS=0,DISP=D
* $$ LST CLASS=A,PRI=3
// JOB TBREORG REORG TABLE "EXAMPLE"."ROUTINE" - DISK
// OPTION LOG
* * * * *
* STEP0001 UNLOAD DDL
* * * * *
// DLBL SQMTPRM,'SQLMSTR.TABLE.PARMS',,VSAM,CAT=SQMCAT
// DLBL SQMDDL,'L.DB2710.EXAMPLE.ROUTINE',0,VSAM, X
RECORDS=001000,RECSIZE=80,DISP=(NEW,KEEP),CAT=SQMCAT
// ASSGN SYS005,SYSRDR
// ASSGN SYS011,SYSLST
// EXEC SQB60,SIZE=AUTO
%%DB2710          EXAMPLE ROUTINE          3NN SQLDBA  SQLDBAPW
/*
// 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(DB2710)'
CONNECT SQLDBA IDENTIFIED BY SQLDBAPW;
COMMENT '***** UNLOAD TABLE "EXAMPLE"."ROUTINE" *****'
SET UPDATE STATISTICS OFF;
UNLOAD TABLE ("EXAMPLE"."ROUTINE")
OUTFILE(SQMDAT1 BLKSZ(02048) PDEV(DASD))
/*
// IF $RC > 0000 THEN
// GOTO $EOJ

```

Figure 62. TABLE Reorg Option 3 (Reorganize TABLE) - Sample Jobstream (Part 1 of 3)

## Reorganization Job Streams

```

* * * * *
* STEP0003 REPRO VSAM DDL TO SAM
* * * * *
// DLBL VSAMIN,'L.DB2710.EXAMPLE.ROUTINE',,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
/*
* * * * *
* 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()'
/*
// IF $RC > 0006 THEN
// GOTO CLOSEIPT
CLOSE SYSIPT,SYS020
* * * * *
* STEP0005 DELETE PARAMETER RECORD
* * * * *
// DLBL SQMTPRM,'SQLMSTR.TABLE.PARMS',,VSAM,CAT=SQMCAT
// ASSGN SYS005,SYSDR
// ASSGN SYS011,SYSLST
// EXEC SQB62,SIZE=AUTO
%%DB2710          EXAMPLE ROUTINE          NN  SQLDBA  SQLDBAPW
/*
// IF $RC > 0000 THEN
// GOTO $EOJ

```

Figure 62. TABLE Reorg Option 3 (Reorganize TABLE) - Sample Jobstream (Part 2 of 3)

```

* * * * *
* STEP0006 DELETE VSAM DDL FILE
* * * * *
// ASSGN SYSLST,IGN
// DLBL FILEIN,'L.DB2710.EXAMPLE.ROUTINE',,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
* * * * *
// DLBL SQMDAT1,'SQLMSTR.DATA.FILE1',0,SD
// EXTENT SYS007,'VSEPK3',1,0,0000201438,06300
// ASSGN SYS007,DISK,VOL='VSEPK3',SHR
// DLBL SQMDAT,'D.DB2710.EXAMPLE.ROUTINE',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
* * * * *
/*
/&

```

Figure 62. TABLE Reorg Option 3 (Reorganize TABLE) - Sample Jobstream (Part 3 of 3)

## Reorganization Job Streams

```

* $$ JOB JNM=TBREORG,CLASS=0,DISP=D
* $$ LST CLASS=A,PRI=3
// JOB TBREORG RELOAD TABLE "EXAMPLE"."ROUTINE" - DISK
// OPTION LOG
* * * * *
* STEP0001 CHECK FOR DATA FILE
* * * * *
// ASSGN SYSLST,IGN
// DLBL FILEIN,'D.DB2710.EXAMPLE.ROUTINE',,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.DB2710.EXAMPLE.ROUTINE',,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.DB2710.EXAMPLE.ROUTINE',,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 63. TABLE Reorg Option 4 (Reload TABLE) - Sample Jobstream (Part 1 of 3)



```

* * * * *
* STEP0004 REPRO VSAM DDL TO SAM
* * * * *
// DLBL VSAMIN,'L.DB2710.EXAMPLE.ROUTINE',,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
* * * * *
// 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(DB2710)'
/*
// IF $RC > 0006 THEN
// GOTO CLOSEIPT
CLOSE SYSIPT,SYS020
// GOTO THEEND
/. CLOSEIPT
* * * * *
* STEP0006 CLOSE SYSIPT
* * * * *
CLOSE SYSIPT,SYS020
// GOTO $EOJ
/. NODAT
// RESET SYSLST

```

Figure 63. TABLE Reorg Option 4 (Reload TABLE) - Sample Jobstream (Part 2 of 3)

```
* * * * *
* STEP0007 NO DATA FILE
* * * * *
* RELOAD ABORTED: MISSING FILE =
* D.DB2710.EXAMPLE.ROUTINE
// GOTO $EOJ
/. NODDL
// RESET SYSLST
* * * * *
* STEP0008 NO DDL FILE
* * * * *
* RELOAD ABORTED: MISSING FILE =
* L.DB2710.EXAMPLE.ROUTINE
/. THEEND
* * * * *
* STEP0009 THE END
* * * * *
/*
/ &
```

Figure 63. TABLE Reorg Option 4 (Reload TABLE) - Sample Jobstream (Part 3 of 3)

---

## Appendix B. DBSPACE and Table 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:

<u>Filename</u>	<u>File ID &amp; Description</u>
<b>SQMPARM</b>	"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. The record is deleted when the REORG completes.
<b>SQMRDAT</b>	"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.
<b>SQMTPARM</b>	"SQLMSTR.TABLE.PARMS"  This is the Table Reorganization parameter file. The job submission programs write a record to this file whenever a job is submitted. The SQMTPARM record contains REORG parameters and is read by the batch DDL generation and timekeeping programs. SQMTPARM is a VSAM KSDS file. It is defined during installation and resides in the Control Center user catalog.
<b>SQMDDL</b>	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: <ol style="list-style-type: none"><li>1. "L" - (indicates DDL)</li><li>2. Database name</li><li>3. DBSPACE owner name</li><li>4. DBSPACE name</li></ol> 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.
<b>SQMDDATn</b>	"SQLMSTR.DATA.FILEn"

## DBSPACE Reorganization Tool Related Files

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.

**SQM DAT** Built dynamically by the submit program.

SQM DAT 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). SQM DAT 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

## DBSPACE Reorganization Tool Related Files

sequence number. SQMWORK is defined during installation and resides on the Control Center user catalog.

Figure 64 on page 130 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.

## DBSPACE Reorganization Tool Related Files

```

COMMENT '***** CONTROL CENTER DBSPACE REORG *****'
COMMENT '* Database: SQLDBA *'
COMMENT '* DBSPACE: "PUBLIC"."SAMPLE" *'
COMMENT '* Date: 06/30/00 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 64. Sample Generated DDL File (Part 1 of 4)

## DBSPACE Reorganization Tool Related Files

```

,"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 64. Sample Generated DDL File (Part 2 of 4)

## DBSPACE Reorganization Tool Related Files

```
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_EMP1"
("MGRNO"
) REFERENCES "SQLDBA"."EMPLOYEE"
ON DELETE SET NULL;
```

Figure 64. Sample Generated DDL File (Part 3 of 4)



## DBSPACE Reorganization Tool Related Files

```

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, FIRSTNAME, 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 64. Sample Generated DDL File (Part 4 of 4)

## DBSPACE Reorganization Tool Related Files

---

## Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10594-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Mail Station P300  
522 South Road  
Poughkeepsie, NY 12601-5400  
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

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

---

## Trademarks

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

AS/400  
C/370  
CICS  
CICS/VSE  
DATABASE 2  
DataPropagator  
DB2  
DRDA  
IBM  
MVS  
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, other countries, or both.

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



---

## 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.

**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 & VM Application Programming*, SC09-2889
- *DB2 Server for VSE & VM Database Administration*, SC09-2888
- *DB2 Server for VSE & VM Database Services Utility*, SC09-2983
- *DB2 Server for VSE & VM Diagnosis Guide and Reference*, LC09-2907
- *DB2 Server for VSE & VM Overview*, GC09-2995
- *DB2 Server for VSE & VM Interactive SQL Guide and Reference*, SC09-2990
- *DB2 Server for VSE & VM Master Index and Glossary*, SC09-2890
- *DB2 Server for VSE Messages and Codes*, GC09-2985
- *DB2 Server for VSE & VM Operation*, SC09-2986
- *DB2 Server for VSE System Administration*, SC09-2981
- *DB2 Server for VSE & VM Performance Tuning Handbook*, GC09-2987
- *DB2 Server for VSE & VM SQL Reference*, SC09-2989

### **Related Publications**

- *DB2 Server for VSE & VM Data Restore*, SC09-2991
- *DRDA: Every Manager's Guide*, GC26-3195
- *IBM SQL Reference, Version 2, Volume 1*, SC26-8416
- *IBM SQL Reference*, SC26-8415

### **Virtual Storage Extended/Enterprise Systems Architecture (VSE/ESA) Publications**

- *IBM VSE/ESA Administration*, SC33-6505
- *IBM VSE/ESA Diagnosis Tools*, SC33-6514
- *IBM VSE/ESA General Information*, GC33-6501
- *IBM VSE/ESA Guide for Solving Problems*, SC33-6510
- *IBM VSE/ESA Guide to System Functions*, SC33-6511
- *IBM VSE/ESA Installation*, SC33-6504

- *IBM VSE/ESA Messages & Codes*, SC33-6507
- *IBM VSE/ESA Networking Support*, SC33-6508
- *IBM VSE/ESA Operation*, SC33-6506
- *IBM VSE/ESA Planning*, SC33-6503
- *IBM VSE/ESA System Control Statements*, SC33-6513
- *IBM VSE/ESA System Macros User's Guide*, SC33-6515
- *IBM VSE/ESA System Macros Reference*, SC33-6516
- *IBM VSE/ESA System Utilities*, SC33-6517
- *IBM VSE/ESA Unattended Node Support*, SC33-6512
- *IBM VSE/ESA Using IBM Workstations*, SC33-6509

### **CICS/VSE Publications**

- *CICS/VSE Application Programming Reference*, SC33-0713
- *CICS/VSE Application Programming Guide*, SC33-0712
- *CICS Application Programming Primer (VS COBOL II)*, SC33-0674
- *CICS/VSE CICS-Supplied Transactions*, SC33-0710
- *CICS/VSE Customization Guide*, SC33-0707
- *CICS/VSE Facilities and Planning Guide*, SC33-0718
- *CICS/VSE Intercommunication Guide*, SC33-0701
- *CICS/VSE Performance Guide*, SC33-0703
- *CICS/VSE Problem Determination Guide*, SC33-0716
- *CICS/VSE Recovery and Restart Guide*, SC33-0702
- *CICS/VSE Release Guide*, GC33-1645
- *CICS/VSE Report Controller User's Guide*, SC33-0705
- *CICS/VSE Resource Definition (Macro)*, SC33-0709
- *CICS/VSE Resource Definition (Online)*, SC33-0708
- *CICS/VSE System Definition and Operations Guide*, SC33-0706
- *CICS/VSE System Programming Reference*, SC33-0711
- *CICS/VSE User's Handbook*, SX33-6079
- *CICS/VSE XRF Guide*, SC33-0704

### **CICS/ESA Publications**

- CICS/ESA General Information, GC33-0803

### **VSE/Virtual Storage Access Method (VSE/VSAM) Publications**

- VSE/VSAM Commands and Macros, SC33-6532
- VSE/VSAM Introduction, GC33-6531
- VSE/VSAM Messages and Codes, SC24-5146
- VSE/VSAM Programmer's Reference, SC33-6535

### **VSE/Interactive Computing and Control Facility (VSE/ICCF) Publications**

- VSE/ICCF Administration and Operation, SC33-6562
- VSE/ICCF Primer, SC33-6561
- VSE/ICCF User's Guide, SC33-6563

### **VSE/POWER Publications**

- VSE/POWER Administration and Operation, SC33-6571
- VSE/POWER Application Programming, SC33-6574
- VSE/POWER Networking, SC33-6573
- VSE/POWER Remote Job Entry, SC33-6572

### **Distributed Relational Database Architecture (DRDA) Library**

- Application Programming Guide, SC26-4773
- Architecture Reference, SC26-4651
- Connectivity Guide, SC26-4783
- DRDA: Every Manager's Guide, GC26-3195
- Planning for Distributed Relational Database, SC26-4650
- Problem Determination Guide, SC26-4782

### **C/370 for VSE Publications**

- IBM C/370 General Information, GC09-1386
- IBM C/370 Programming Guide for VSE, SC09-1399
- IBM C/370 Installation and Customization Guide for VSE, GC09-1417
- IBM C/370 Reference Summary for VSE, SX09-1246
- IBM C/370 Diagnosis Guide and Reference for VSE, LY09-1805

### **VSE/REXX Publication**

- VSE/REXX Reference, SC33-6642

### **Other Distributed Data Publications**

- IBM Distributed Data Management (DDM) Architecture, Architecture Reference, Level 4, SC21-9526
- IBM Distributed Data Management (DDM) Architecture, Implementation Programmer's Guide, SC21-9529
- VM/Directory Maintenance Licensed Program Specification, GC20-1836
- IBM Distributed Relational Database Architecture Reference, SC26-4651
- IBM Systems Network Architecture, Format and Protocol Reference, SC30-3112
- SNA LU 6.2 Reference: Peer Protocols, SC31-6808
- Reference Manual: Architecture Logic for LU Type 6.2, SC30-3269
- IBM Systems Network Architecture, Logical Unit 6.2 Reference: Peer Protocols, SC31-6808
- Distributed Data Management (DDM) General Information, GC21-9527

### **CCSID Publications**

- Character Data Representation Architecture, Executive Overview, GC09-2207
- Character Data Representation Architecture Reference and Registry, SC09-2190

### **C/370 Publications**

- IBM C/370 Installation and Customization Guide, GC09-1387
- IBM C/370 Programming Guide, SC09-1384

### **Communication Server for OS/2 Publications**

- Up and Running!, GC31-8189
- Network Administration and Subsystem Management Guide, SC31-8181
- Command Reference, SC31-8183
- Message Reference, SC31-8185
- Problem Determination Guide, SC31-8186

### **Distributed Database Connection Services (DDCS) Publications**

- DDCS User's Guide for Common Servers, S20H-4793
- DDCS for OS/2 Installation and Configuration Guide, S20H-4795

### **VTAM Publications**

- VTAM Messages and Codes, SC31-6493
- VTAM Network Implementation Guide, SC31-6494
- VTAM Operation, SC31-6495

- *VTAM Programming*, SC31-6496
- *VTAM Programming for LU 6.2*, SC31-6497
- *VTAM Resource Definition Reference*, SC31-6498
- *VTAM Resource Definition Samples*, SC31-6499

#### **CSP/AD and CSP/AE Publications**

- *Developing Applications*, SH20-6435
- *CSP/AD and CSP/AE Installation Planning Guide*, GH20-6764
- *Administering CSP/AD and CSP/AE on VM*, SH20-6766
- *Administering CSP/AD and CSP/AE on VSE*, SH20-6767
- *CSP/AD and CSP/AE Planning*, SH20-6770
- *Cross System Product General Information*, GH23-0500

#### **Query Management Facility (QMF) Publications**

- *Introducing QMF*, GC27-0714
- *Installing and Managing QMF for VSE*, GC27-0721
- *QMF Reference*, SC27-0715
- *Installing and Managing QMF for VM*, GC27-0720
- *Developing QMF Applications*, SC27-0718
- *QMF Messages and Codes*, GC27-0717
- *Using QMF*, SC27-0716

#### **Query Management Facility (QMF) for Windows Publications**

- *Getting Started with QMF for Windows*, SC27-0723
- *Installing and Managing QMF for Windows*, GC27-0722

#### **DL/I DOS/VS Publications**

- *DL/I DOS/VS Application Programming*, SH24-5009

#### **COBOL Publications**

- *VS COBOL II Migration Guide for VSE*, GC26-3150
- *VS COBOL II Migration Guide for MVS and CMS*, GC26-3151
- *VS COBOL II General Information*, GC26-4042
- *VS COBOL II Language Reference*, GC26-4047
- *VS COBOL II Application Programming Guide*, SC26-4045
- *VS COBOL II Application Programming Debugging*, SC26-4049

- *VS COBOL II Installation and Customization for CMS*, SC26-4213
- *VS COBOL II Installation and Customization for VSE*, SC26-4696
- *VS COBOL II Application Programming Guide for VSE*, SC26-4697

#### **Systems Network Architecture (SNA) Publications**

- *SNA Transaction Programmer's Reference Manual for LU Type 6.2*, GC30-3084
- *SNA Format and Protocol Reference: Architecture Logic for LU Type 6.2*, SC30-3269
- *SNA LU 6.2 Reference: Peer Protocols*, SC31-6808
- *SNA Synch Point Services Architecture Reference*, SC31-8134

#### **Miscellaneous Publications**

- *IBM 3990 Storage Control Planning, Installation, and Storage Administration Guide*, GA32-0100
- *Dictionary of Computing*, ZC20-1699
- *APL2 Programming: Using Structured Query Language*, SH21-1056
- *ESA/390 Principles of Operation*, SA22-7201

#### **Related Feature Publications**

- *DB2 for VSE Control Center Operations Guide*, GC09-2992
- *DB2 Replication Guide and Reference*, SC26-9920



---

# Index

## A

- adding a monitor 72, 74
- adding objects to a group 63
- adding users to a group 65
- Analysis
  - of DBSPACES 2
- Application Group
  - adding objects to 63
  - defining 62
- Architecture
  - database connection 4
  - invocation 4
  - overview 1

## B

- backup 86
  - processing flow 87, 93
- Batch jobs
  - deleting 3
  - holding 3
  - releasing 3
- Benefits
  - automation 1
  - DDL generation 1
  - jobscheduling 1
  - QMF 1
  - reorganization 1
  - statistics updating 1

## C

- CEMS transaction ID 5
- changing monitor settings 78
- changing to a new database
  - overview 5
- CICS
  - report controller interface 3
  - TRANSIDs 4
- CICS ID 7
- CICS Report Controller Interface
  - a sample session 47
  - highlights 3
  - overview 47
- CLUSTERRATIO 31
- commitcount on reload 18
- console message monitoring 70
- Control Center Packages
  - rebinding 2
- Controlling access 1

## D

- DATA CAPTURE 90
- database v
- DATABASE DEFINITION PROC
  - parameter 22
- DATE tape file parameter 43
- DBA 1

- DBSPACE Analysis Tools
    - additional topics 37
    - initial execution 37
    - reorganization work space 37
    - before you begin 27
  - DBSPACE reorganization analysis tool 33
  - highlights 2
  - how they work 27
  - job options 36
  - overview 27
  - reorg criteria 31
  - Submit screen 35
  - the DBSPACE Analysis Utility
    - screen 29
    - functions 30
    - reorg criteria 31
    - selection options 31
  - the SQLMAINT table 28
  - update statistics analysis tool 32
- DBSPACE Reorganization
    - before you begin 13
    - scheduling 13
    - single user mode 22
    - when to reorg 13
  - DBSPACE REORGANIZATION screen
    - options 88
  - DBSPACE REORGANIZATION Screen 15
    - optional parameters 17
    - options 16
  - DBSPACE Reorganization Tool
    - about the tool 13
    - benefits 13
    - failure restart 24
    - features 14
    - highlights 2
    - how it works 14
      - processing flow 14
      - related files 127
    - problem analysis 24
    - scheduling 24
    - single user mode reorganization 22
      - before you choose it 22
      - parameters 22
    - special considerations 24
      - failure restart 24
      - problem analysis 24
      - scheduling 24
    - submit screen 20
      - optional parameters 21
    - unloading to tape 23
    - use 19, 90
      - generate DDL 19
      - reload DBSPACE 20
      - reorganize DBSPACE 20
      - unload DBSPACE 19
- DBSPACES
    - analyzing 2
    - DDL to recreate 2
    - listing reorganization candidates 30

- DBSPACES (*continued*)
  - listing update statistics candidates 30
  - reloading 2
  - reorganizing 2, 13
  - unloading 2
- DDL 1
- DDL STMTS 18
- deleting a monitor 72
- DEVICE CLASS 43
- displaying monitor definitions 72
- DISPosition
  - job card parameter 92

## F

- FILE # 37
- FILE SEQUENCE NUMBER 43
- function keys 7

## G

- Generate DDL 16, 85, 90
- GENERATION NUMBER 43
- granting
  - authority to a user group 66
- Group Authorization Tool
  - adding objects to a group 63
  - adding users to a group 65
  - defining a user group 64
  - defining an application group 62
  - granting authority to a user group 66
  - highlights 2

## H

- Help Facility
  - highlights 3
  - overview 51
  - TRANSID 4

## I

- Initializing the SQLMAINT Control Table 30
- Installation
  - before you begin... 5
  - overview 5
  - placement of code 5
  - prerequisite programs 3
  - preventive service planning 5
  - production library 5
- Interfaces
  - CICS report controller 3
  - operator command 2
- Invocation
  - using panels 4
  - using TRANSID 4

## J

- Job Scheduling
  - highlights 3
  - parameters 21
  - repetitive DBSPACE reorganizations 24
- job submission
  - monitor report 73

## K

- kernal monitor 71

## L

- Listing DBSPACE Reorganization
  - Candidates 30
- Listing Update Statistics Candidates 30
- LOGFULL condition 24
- LOGMODE parameter 23

## M

- Main Menu 7
- MODE 43
- modifying a monitor 72
- monitor
  - adding 72, 74
  - changing settings 78
  - console messages 70
  - database monitor 69
  - deleting 72
  - display definition 72
  - kernel 69
  - modifying 72
  - print report 73
  - reset 69
  - resetting data 72
  - specifying thresholds 77
  - starting 78
  - starting the kernel 71
  - stopping 80
  - stopping the kernel 71
  - threshold 70
  - view collected data 72, 79
- Monitor Utilities
  - highlights 2
- Moving DBSPACES 2
- Multiple database support 2

## N

- NDIRBUF parameter 23
- NOVERFLOW ROWS > n% 31
- NPAGBUF parameter 23
- NUMBER OF TRACKS/BLOCKS 41

## O

- Operation
  - function keys 7
  - getting started 7
  - operator command tool 9
  - signing on 8

## Operator Commands

- COUNTER 9
  - highlights 2
  - menu 9
  - SHOW 9
- Overview
  - of architecture 1
  - of installation 5
  - product 1

## P

- Package Utilities
  - highlights 3
  - package migration 54
  - rebind package 54
  - reload package 54
  - unload package 53
  - view package 54
- Parameters
  - DAYS SINCE 31
  - DBSPACE analysis 31
  - DBSPACE analysis jobs 36
  - DBSPACE migration 17
  - DBSPACE reorganization 17
  - disk work file label definition 41
  - FILE # 16, 88
  - job scheduling 21
  - rebinding 18, 37, 89
  - REORG CRITERIA 31
  - single user mode 1, 22
  - TABLE migration 88
  - TABLE reorganization 88
  - tape work file label definition 42
  - TLBL 3
  - WITH REFRESH 30
- printing monitor reports 73
- Product
  - benefits 1
  - description 4
  - software requirements 3
- Production library 5

## Q

- QMF 3, 7

## R

- Rebind Package 18, 89
- REBIND PACKAGES 37
- RELATIVE TRACK/BLOCK 41
- Reload DBSPACE 17
- reload TABLE 91
- Reorg Criteria 31
- REORG\_DATE 29
- REORG\_ELAPSED 29
- REORG\_FREEPCT 29
- REORG\_PCTINDX 29
- REORG\_STATUS 29
- REORG\_TIME 29
- REORG\_WEIGHT 29
- Reorganize DBSPACE 16
- reorganize table 91
- Requirements
  - program 3

## Requirements (continued)

- software 3
- reset monitor 69
- resetting monitor data 72

## S

- Scheduling
  - DBSPACE analysis jobs 35
  - DBSPACE reorganizations 13
  - UPDATE STATISTICS jobs 35
- Scheduling jobs 3
- setting monitor thresholds 70
- Single User Mode
  - before you choose it 22
  - DBSPACE reorganization 22
  - reorganization parameters 22
  - startuppparameters 1
- Software requirements 3
- specifying monitor thresholds 77
- SQB01, example using 19, 105, 106, 115, 123
- SQB02, example using 105, 106, 109, 115, 123
- SQB60, example using 90, 119, 120
- SQB62, example using 120
- SQDR transaction ID 5, 15
- SQFM transaction ID 5
- SQGA transaction ID 5, 61
- SQHM transaction ID 5
- SQL Master for VSE
  - and DB2 4
  - highlights 1
- SQL/DS
  - and Control Center for VSE 4
- SQLMAINT
  - updated during reorgs 15
- SQLMAINT table
  - description 28
  - initializing 30
- SQM transaction ID 5
- SQMDAT 128
- SQMDDL 127, 129
  - definition 127
- SQMDDLn 128
- SQMM transaction ID 5
- SQMMESG 128
- SQMPPARM 24, 127
  - definition 127
- SQMRDAT 127
  - definition 127
- SQMTAPE 128
- SQMWORK 39, 41, 42, 43, 128, 129
- SQOM transaction ID 5
- SQPM transaction ID 5, 54
- SQTU transaction ID 5
- starting a monitor 78
- Startup parameters 1
- stopping a monitor 80
- stopping the kernal monitor 71
- stored procedures 101
- System
  - software 3

## T

- table job submission
  - optional parameters 97, 98
  - required parameters 91
- table reorganization
  - submit screen 91
- Table REORGANIZATION screen 88
  - optional parameters 88
- table utility
  - CREATE TABLE function 93
  - creating table parameters 96
  - description 81
  - DROP TABLE function 84
  - inserting and deleting data 95
  - invocation 81
  - LIST TABLES function 83
  - REORGANIZE TABLE function 85
  - UPDATE STATISTICS function 98
- table utility tool 3
- TAPE FILE NAME 37
- transactions
  - CEMS 4, 5
    - for CICS report controller 4
    - for dbspace analysis menu 4
    - for dbspace reorganization 4
    - for group authorization tool 4
    - for help facility 4
    - for main menu 4
    - for operator commands menu 4
    - for package utility tool 4
    - for work file label definition 4
  - SQDR 4, 5, 15
  - SQFM 4, 5
  - SQGA 4, 5, 61
  - SQHM 4, 5
  - SQM 4, 5
  - SQMM 5
  - SQMM1 4
  - SQOM 4, 5
  - SQPM 4, 5, 54
  - SQTU 5, 81

## U

- UNCLUSTERED INDEX 31
- Unload DBSPACE 16
- unload table 90
- UPDATE ALL STATISTICS 36
- update statistics 18
- UPSTAT\_DATE 29
- UPSTAT\_ELAPSED 29
- UPSTAT\_TIME 29
- User Group
  - adding users to 65
  - defining 64
- User ID 7
- using monitors 74

## V

- VERSION NUMBER 43
- view monitor collected data 72, 79
- VOLUME SEQUENCE NUMBER 43
- VOLUME SERIAL NUMBER 41, 42

## W

- Work File Label Definition
  - disk work file label definition
    - fields 41
  - disk work file label definition screen 40
  - highlights 3
  - how it works 39
  - overview 39
  - package messages work file 44
  - package work file 43
  - special considerations 45
    - size of defined files 45
  - tape work file label definition
    - fields 42
  - tape work file label definition screen 42
- Work File Label Definition screen 39





---

## Contacting IBM

Before you contact DB2 customer support, check the product manuals for help with your specific technical problem.

For information or to order any of the DB2 Server for VSE & VM products, contact an IBM representative at a local branch office or contact any authorized IBM software remarketer.

If you live in the U.S.A., then you can call one of the following numbers:

- 1-800-237-5511 for customer support
- 1-888-426-4343 to learn about available service options

---

## Product information

DB2 Server for VSE & VM product information is available by telephone or by the World Wide Web at <http://www.ibm.com/software/data/db2/vse-vm>

This site contains the latest information on the technical library, product manuals, newsgroups, APARs, news, and links to web resources.

If you live in the U.S.A., then you can call one of the following numbers:

- 1-800-IBM-CALL (1-800-426-2255) to order products or to obtain general information.
- 1-800-879-2755 to order publications.

For information on how to contact IBM outside of the United States, go to the IBM Worldwide page at <http://www.ibm.com/planetwide>

In some countries, IBM-authorized dealers should contact their dealer support structure for information.



File Number: S370/4300-50  
Program Number: 5697-F42

Printed in U.S.A.

GC09-2992-01



Spine information:



DB2 Server for VSE & VM

Control Center Operations Guide for VSE

Version 7 Release 3