

DFSMS/MVS Version 1 Release 5



NaviQuest User's Guide

DFSMS/MVS Version 1 Release 5



NaviQuest User's Guide

Note!

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

Third Edition (March 1999)

This edition applies to the Version 1 Release 5 of DFSMS/MVS (5695-DF1), Release 7 of OS/390 (5647-A01), and all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

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

A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to:

International Business Machines Corporation
RCF Processing Department
G26/050
5600 Cottle Road
SAN JOSE, CA 95193-0001
U.S.A.

Or you can send your comments electronically to starpubs@vnet.ibm.com.

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 1995, 1999. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	v
Trademarks	v
About This Book	vii
What You Should Know Before Reading This Publication	vii
How to Tell if this Book is Current	vii
Referenced Publications	viii
References to Product Names Used in DFSMS/MVS Publications	ix
Summary of Changes	xi
Third Edition, March 1999	xi
Chapter 1. What Is NaviQuest?	1
How to Start NaviQuest	2
Terminology	3
Methods for Collecting Test Data	4
Data Classification Methodology	5
Chapter 2. Testing ACS Routines	7
Setting Up the Test Environment	7
Testing Procedures	8
NaviQuest Testing Scenario	20
Data Testing	20
Method of Testing	21
Testing Example	21
Chapter 3. Performing Storage Administration Tasks in Batch	25
How to Run Storage Administration Tasks in Batch	25
ISMF Option 11.7 Batch Testing and Configuration Management	25
Sample JCL for Batch	26
NaviQuest CLISTS and REXX EXECs	31
Chapter 4. Creating DFSMS Online Reports	127
Creating Data Set Reports from Saved ISMF Lists	128
Creating Volume Reports from Saved ISMF Lists	129
Creating Customized Tape Reports from Saved ISMF Lists	130
Data Set Report from DCOLLECT Data	131
DASD Volume Report from DCOLLECT Data	132
SMS Configuration Report from DCOLLECT Data	132
Chapter 5. Additional Storage Administration Functions	135
QSAVE and QRETRIEV Commands: Saving and Retrieving ISMF Selection Criteria	135
How to Use the QSAVE and QRETRIEV Commands	135
Creating Model Commands	136
How to Create Model Commands	136
COPYFILT Macro: COPYLIB Facility for FILTLISTS	140
How to Setup the COPYFILT Macro	140
How to Use the COPYFILT MACRO after Initial Setup	141

Glossary of Terms	143
Index	149

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates.

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 of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the responsibility of the user.

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

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

IBM Corporation
Information Enabling Requests
Dept. DWZ
5600 Cottle Road
San Jose, CA 95193

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

Any pointers in this publication to non-IBM Web sites are provided for convenience only, and do not in any manner serve as an endorsement of these Web sites.

Trademarks

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

CICS
DFSMS
DFSMSdfp
DFSMSdss
DFSMShsm
DFSMS/MVS
ESA/370
ESA/390
IBM

MVS/DFP
MVS/ESA
MVS/SP
NaviQuest
RACF
Resource Measurement Facility
RMF
VM/ESA

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Windows is a trademark of Microsoft corporation.

About This Book

This book is intended for storage administrators who manage storage under DFSMS/MVS system-managed storage (SMS). It describes how, by using NaviQuest, storage administrators can easily perform the tasks associated with the initial implementation of SMS, and perform ongoing ACS and configuration maintenance.

What You Should Know Before Reading This Publication

To use this book effectively, you should be familiar with the following:

- Data and storage management concepts and the functions provided by DFSMS/MVS
- Interactive Storage Management Facility (ISMF)
- Interactive System Productivity Facility (ISPF) and ISPF/Program Development Facility (ISPF/PDF)

For information on ISMF, refer to *DFSMS/MVS Version 1 Release 5 Using the Interactive Storage Management Facility*.

Storage administrators can refer to *DFSMS/MVS Version 1 Release 5 DFSMSdfp Storage Administration Reference* for information dealing with storage administration tasks.

How to Tell if this Book is Current

IBM regularly updates its books with new and changed information. When first published, both hardcopy and BookManager softcopy versions of a book are identical, but subsequent updates might be available in softcopy before they are available in hardcopy. Here's how to determine the level of a book:

- Check the book's order number suffix (often referred to as the dash level). A book with a higher dash level is more current than one with a lower dash level. For example, in the publication order number SC26-4930-02, the dash level 02 means that the book is more current than previous levels, such as 01 or 00. Suffix numbers are updated as a product moves from release to release, as well as for hardcopy updates within a given release.
- Check to see if you are using the latest softcopy version. To do this, compare the last two characters of the book's file name (also called the book name). The higher the number, the more recent the book. For example, DGT1U302 is more recent than DGT1U301.
- Compare the dates of the hardcopy and softcopy versions of the books. Even if the hardcopy and softcopy versions of the book have the same dash level, the softcopy could be more current. This will not be apparent from looking at the edition notice. The edition notice number and date remain that of the last hardcopy version. When you are looking at the softcopy product bookshelf, check the date shown to the right of the book title. This will be the date that the softcopy version was created.

Also, an asterisk (*) is added next to the new and changed book titles in the CD-ROM booklet and the README files.

Vertical lines to the left of the text indicate changes or additions to the text and illustrations. For a book that has been updated in softcopy only, the vertical lines indicate changes made since the last printed version.

Referenced Publications

This book references the following publications:

Short Title	Publication Title	Order Number
<i>DFSMS/MVS V1R5 Access Method Services for ICF</i>	<i>DFSMS/MVS Version 1 Release 5 Access Method Services for Integrated Catalog Facility</i>	SC26-4906
<i>DFSMS/MVS V1R5 Using ISMF</i>	<i>DFSMS/MVS Version 1 Release 5 Using the Interactive Storage Management Facility</i>	SC26-4911
<i>DFSMS/MVS V1R5 DFSMSdfp Storage Administration Reference</i>	<i>DFSMS/MVS Version 1 Release 5 DFSMSdfp Storage Administration Reference</i>	SC26-4920
<i>OS/390 ISPF Dialog Developer's Guide and Reference</i>	<i>OS/390 Interactive System Productivity Facility Dialog Developer's Guide and Reference</i>	SC28-1273

References to Product Names Used in DFSMS/MVS Publications

DFSMS/MVS publications support DFSMS/MVS, 5695-DF1, as well as the DFSMSdfp base element and the DFSMSshsm, DFSMSdss, and DFSMSrmm features of OS/390, 5647-A01. DFSMS/MVS publications also describe how DFSMS/MVS interacts with other IBM products to perform the essential data, storage, program and device management functions of the operating system.

DFSMS/MVS publications typically refer to another IBM product using a generic name for the product. When a particular release level of a product is relevant, the reference includes the complete name of that product. This section explains the naming conventions used in the DFSMS/MVS library for the following products:

MVS can refer to:

- MVS/ESA SP Version 5, 5695-047 or 5695-048
- The MVS base control program (BCP) of OS/390, 5647-A01

All MVS book titles used in DFSMS/MVS publications refer to the OS/390 editions. Users of MVS/ESA SP Version 5 should use the corresponding MVS/ESA book. Refer to *OS/390 Information Roadmap* for titles and order numbers for all the elements and features of OS/390.

For more information about OS/390 elements and features, including their relationship to MVS/ESA SP and related products, please refer to *OS/390 Planning for Installation*.

RACF can refer to:

- Resource Access Control Facility (RACF), Version 2, 5695-039
- The RACF element of the OS/390 Security Server, an optional feature of OS/390

All RACF book titles refer to the Security Server editions. Users of RACF Version 2 should use the corresponding book for their level of the product. Refer to *OS/390 Security Server (RACF) Introduction* for more information about the Security Server.

CICS can refer to:

- CICS/MVS, 5665-403
- CICS/ESA, 5685-083
- The CICS element of the CICS Transaction Server for OS/390, 5665-147

All CICS book titles refer to the CICS Transaction Server for OS/390 editions. Users of CICS/MVS and CICS/ESA should use the corresponding books for those products. Please see *CICS Transaction Server for Planning for Installation* for more information.

Summary of Changes

Third Edition, March 1999

This publication is a major revision in support of the functional changes introduced with DFSMS/MVS Version 1 Release 5. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

This revision also includes maintenance and editorial changes.

The following summarizes changes to NaviQuest's batch support information:

- Added Define/Alter/Display keywords and samples for:
 - Aggregate groups
 - Base configuration
 - Data class
 - Management class
 - Storage class
- Added Define/Alter keywords and samples for new storage group types:
 - Dummy
 - Object
 - Object backup
 - VIO

Note: For other important updates to this book, please check information APAR II11474, a repository of DFSMS/MVS 1.5 information that was not available at the time DFSMS/MVS books were published for general availability.

Chapter 1. What Is NaviQuest?

DFSMS/MVS NaviQuest is a data and storage management tool for implementing, testing, and verifying the DFSMS SMS environment. NaviQuest is installed under the Interactive Storage Management Facility (ISMF) Primary Option Menu and uses the standard Interactive System Productivity Facility (ISPF) panel interface.

With NaviQuest you can:

- Create DFSMS implementation test cases

Initially you might use NaviQuest in the design and testing phase of your first automatic class selection (ACS) routines and SMS configuration.

- Run selected DFSMS testing functions in batch
- Update DFSMS configuration values in batch

NaviQuest lets you run jobs in batch to alter or define the space management or backup attributes for management class. You can also use batch mode to alter or define pool or tape storage groups.

- Create reports interactively or in batch

NaviQuest reports can be used during your DFSMS planning and design, for both DASD and for tape data migration.

- Perform ongoing storage administration activities

Use NaviQuest for production assurance prior to any changes to your DFSMS environment. The cross-reference facilities simplify the tasks of testing and verifying SMS configuration changes.

To simplify ongoing activities, the Model Command Generator option can be used to create a series of commands or control cards tailored with the data set name or volume serial number that is provided in an ISMF list or DCOLLECT report. For information about using the Model Command Generator option and other NaviQuest functions, see Chapter 5, “Additional Storage Administration Functions” on page 135.

Test ACS routines better, faster, and more completely with NaviQuest testing functions.

The power and capabilities of DFSMS require that ACS routines assign the correct data class, storage class, management class, and storage group to each data set. Implementing DFSMS or any change that affects the DFSMS environment must be tested. The results from the DFSMS configuration are unpredictable if the ACS routines have logic errors or if there are data sets that the routines are not coded to handle.

Before a DFSMS configuration is activated on the production system, you can:

- Easily create test cases to perform extensive testing against test data that represent actual data sets.
- Run the tests in batch, freeing the workstation for other work.
- Compare the test results against the *expected* results.

How to Start NaviQuest

Start NaviQuest by selecting option 11, *Enhanced ACS Management*, from the ISMF Primary Option Menu.

```
Panel Help
-----
DGTSMMD2      ISMF PRIMARY OPTION MENU - DFSMS/MVS 1.5
Enter Selection or Command ==> _____

Select one of the following options and press Enter:

 0 ISMF Profile           - Change ISMF user profile
 1 Data Set              - Perform Functions Against Data Sets
 2 Volume                - Perform Functions Against Volumes
 3 Management Class      - Specify Data Set Backup and Migration Criteria
 4 Data Class            - Specify Data Set Allocation Parameters
 5 Storage Class         - Specify Data Set Performance and Availability
 6 Storage Group         - Specify Volume Names and Free Space Thresholds
 7 Automatic Class Selection - Specify ACS Routines and Test Criteria
 8 Control Data Set      - Specify System Names and Default Criteria
 9 Aggregate Group       - Specify Data Set Recovery Parameters
10 Library Management    - Specify Library and Drive Configuration
11 Enhanced ACS Management - Perform Enhanced Test/Configuration Management
 C Data Collection       - Process Data Collection Function
 L List                  - Perform functions Against Saved ISMF Lists
 R Removable Media Manager - Perform Functions Against Removable Media
 X Exit                  - Terminate ISMF
Use HELP Command for Help; Use END Command or X to Exit.
```

Figure 1. ISMF Primary Option Menu for Storage Administrators, DGTSMMD2

Choosing option 11 from the ISMF Primary Option Menu takes you to the NaviQuest Primary Option Menu.

```
Panel Help
-----
ACBSMDP0      ENHANCED ACS MANAGEMENT - NaviQuest PRIMARY OPTION MENU
Enter Selection or Command ==> _____

Select one of the following options and press Enter:

 1 Test Case Generation
 2 ACS Test Listings Comparison
 3 Enhanced ACS Test Listing
 4 Test Case Update with Test Results
 5 SMS Report Generation
 6 Model Commands Generation
 7 Batch Testing/Configuration Management
 X Exit

Use HELP Command for Help; Use END Command or X to Exit.
```

Figure 2. NaviQuest Primary Option Menu, ACBSMDP0

When you select one of the options from the NaviQuest Primary Option Menu, each successive panel will guide you through the choices available for that function.

Terminology

Several new terms are unique to NaviQuest. You will need to familiarize yourself with them before you begin to use this facility.

The following terminology and concepts are used throughout this book:

base line test set

A special case test set, which includes all test cases for data sets not planned for SMS management. Expected values are nulls (' '); the associated special subtype prefix is NEVR.

data classification

The process of determining the data types in your installation and identifying data subtypes that require specific DFSMS data services.

data subtype Groupings of data sets from a data type needing identical DFSMS services, such as performance, backup, migration, or deletion. NaviQuest testing is managed at the subtype level.

data type Major groupings of data. TSO, test, batch production, online production, system, and temporary are examples of data types that most installations have.

errors Results that are different from the expected results. Errors are a subset of exceptions.

exceptions A test case whose results differ from the saved expected results. For a regression test case, this exception is also an error; for an initial test case, this exception might or might not be an error. If the result is equal to the expected value, there is no error.

expected results

Values that you want assigned by the ACS routines for data class, storage class, management class, and storage group for a specific subtype when the ACS routines perform correctly.

initial test First-time testing of a single data subtype performed prior to converting the data to SMS. The test may include regression tests for other data subtypes that have already been tested successfully.

phase or implementation phase

Data conversion to SMS management of one or more data subtypes with the same data type.

phase test set

Group of subtype test sets that define all data subtypes that make up a phase. Phase test sets are converted to SMS management within a single phase.

regression test

Testing of data subtypes that have already been successfully tested, along with the initial test of the current phase subtype.

results

Values that are assigned by the ACS routines for data class, storage class, management class, and storage group for a specific test case. These results might or might not be correct.

saved expected results

Results that have been saved in the test case after the successful initial test of the subtype test set. The results are used by the ACS comparison function during regression testing.

subtype prefix

A unique 1-to-4 character prefix associated with each data subtype. This prefix is used to relate the data classification data subtype to the subtype test set and to group all test cases together for a single subtype test set.

subtype test set

A group of test cases for all data sets associated with the same data subtype. All test cases within one subtype test set have the same expected results.

testbed library

A PDS that contains one member to define each test case.

test case

Parameters associated with a single data set that is tested using ISMF option 7.4.3.

Methods for Collecting Test Data

We suggest that you become familiar with the methods used to create test data before going to Chapter 2, "Testing ACS Routines" on page 7 .

The procedures for collecting test data are documented in "Testing Procedures" on page 8.

NaviQuest collects data for testing from any of four sources:

- ISMF lists
- DCOLLECT data
- SMF data
- VMA data

Although you can successfully use all four sources to create test data, there are restrictions or considerations associated with choosing one method over another.

ISMF Lists

Using the standard facilities of ISMF saved lists, you can select the data sets that you want NaviQuest to enter into the testbed library as DFSMS test cases. ISMF filtering capabilities help you to select data associated with one category of data (data subtype) at a time. When ISMF lists are used to create test cases, you can use NaviQuest's ISMF batch capabilities to generate and save the list.

Consideration: ISMF lists provide greater flexibility. For example, you can tailor the list by removing data sets.

DCOLLECT Data

DCOLLECT data can be used as input to create test cases. The DCOLLECT input is created as an independent batch job by using either the IDCAMS utility program or the ISMF “Data Collection” (option C).

Consideration: The IDCAMS DCOLLECT function runs in batch and is faster than ISMF lists.

SMF Data

Input from system management facility (SMF) data on new allocations can be used to create test cases. This SMF input data is created by using the storage class ACS exit code available in the samples distributed with the CBIPO optional materials. Once the exit is installed and a DFSMS minimal or null configuration is running, all new allocations are captured and written to SMF. CBIPO materials also include a program (ACSTST) for postprocessing of the SMF data, which NaviQuest uses to generate standard DFSMS test cases.

Restriction: Use the SMF data created out of the IGDACSSC exit in selected situations. For example, the extensive testing of system temporary data sets.

VMA Data

VMA data (tape data sets) can be used as input to create test cases. VMA data is gathered in the process of analyzing your tape data. This may be done in preparation for either a tape mount methodology (TMM) implementation or the design of your system-managed tape environment.

Restriction: Use only VMA data to create test cases for tape data sets.

Data Classification Methodology

Data classification is a methodology for DFSMS implementation that translates high-level requirements into data classes, storage classes, and management classes and identifies the decision points needed in the ACS routines.

Use the data classification technique to build a tree diagram where each major branch (data type) is a collection of data to be migrated to DFSMS management, each in a separate phase of the implementation process.

Figure 3 shows six data types to be managed by SMS.

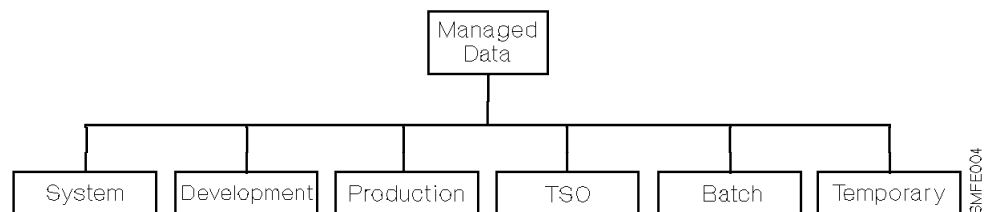


Figure 3. Typical Data Types

After the initial tree diagram has been built, each data type is split further until no more downward splits need to be made. Further splits result in the creation of *data subtypes*. Each data subtype is a distinct set of data sets requiring a unique set of assigned classes. Figure 4 on page 6 is an example of how a TSO data type can be split into different data subtypes.

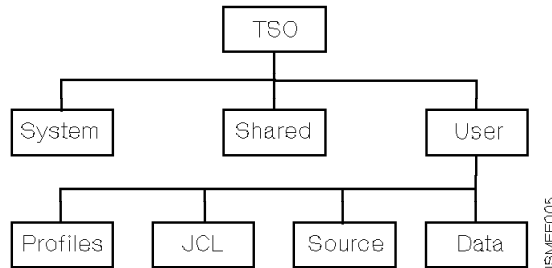


Figure 4. TSO Data Type with Data Subtypes

NaviQuest is designed to support testing of data subtypes determined during data classification. All data sets associated with a single data subtype will have the same expected ACS results. NaviQuest testing at the subtype level lets you manage all test cases for a data subtype and validate the results for that subtype.

Chapter 2. Testing ACS Routines

The following is an overview of the procedures required to complete the ACS routine testing phase. Apply this process to each new data subtype until all data has been migrated. For the specific procedure, see “Testing Procedures” on page 8.

- After a sample of the data sets that are to be tested has been chosen (or created), a base line test set is generated, to represent the ACS routines *before* they are changed.
- After the base line test set has been created, the ACS routines are changed (but not activated) to manage the data type. The same set of test cases is then run through ACS testing, generating a new ACS listing.
- A comparison test is run against the base line test set, and the new listing and exceptions are reported in a comparison report. If the change to the ACS routines was made correctly, the exceptions reported should contain only the new test cases.
- If no errors occur after converting the data to SMS management, the test cases are updated to reflect the expected results for future regression testing.
- The current configuration is saved and the new configuration is activated. After the activation, conversion of data to SMS management can proceed.

Setting Up the Test Environment

Prepare your DFSMS testing environment and create the test case library.

1. Perform data classification to determine the data types.

While this step is optional, if you do not perform data classification for the data types, you must still identify the data subtypes that you want NaviQuest to test.

2. Assign the subtype prefix.

If you performed data classification (step 1), you must assign a 1-to-4 character member name prefix to each data classification data subtype. Even if you do not use data classification, you still must identify the different types of tests you want done and assign them a subtype prefix, because each test runs against a different data subtype and produces a different expected result.

Note: For the base line test set, you must use the subtype prefix NEVR.

3. Determine the amount of testing to be done.

You can choose to conduct tests of all data sets in a given data subtype or you can conduct tests with a selected percentage of the data sets. If you use all the data sets, the testing is more extensive, with less chance for error. However, you can use a sample of the data to minimize the resources used while testing.

4. Determine when to place test cases into the testbed library.

The testbed library stores test cases for later use in bulk or regression testing.

You can either initially place test cases for all data subtypes into the testbed library or wait to add each subtype test set to the testbed library at the start of each phase. Fully loading the testbed library with all test cases prior to any

testing results in more thorough testing. Alternatively, adding the test cases at the start of each phase reduces the total time taken for test runs.

5. Determine how the data for the test cases is to be created.

You must also decide how you want to collect the data set test cases. For each data subtype, decide if you want ISMF lists, DCOLLECT, VMA, or SMF to be the source of the test cases. See “Methods for Collecting Test Data” on page 4 for an explanation of the data collection methods.

Note: The percentage of test data and method of collecting test case data can be tailored for each data subtype. This is independent of your choice of storing all test cases as you initially build the testbed library or at the start of each phase’s testing. For example, you may decide to test all production data sets for production data, but to use only a ten percent sample of your test data sets.

6. Determine the order of testing.

Because the order of testing is normally the same as the order of data conversion to SMS management, you must determine the order that you want to test data subtypes with NaviQuest. Only a single data subtype can be initially tested at one time.

Note: Other data subtypes that have already been tested are regression tested at this time. NaviQuest testing is usually done against all subtypes of one data type before starting the next.

7. Allocate the testbed library.

Note: You can either create a testbed library or let NaviQuest create one for you. The following instructions will guide you in how to create one.

To create a testbed library, create a PDS with the following attributes:

- LRECL = 80
- BLKSIZE = 0

Each test case is a member of the PDS. All test cases associated with a single data type have the same subtype prefix and have the identical expected results.

The amount of space allocated to the library depends on how many data sets the installation has and how many test cases you want to create. For example, on a 3380 device, 1000 test cases take approximately 8 cylinders and 52 directory blocks.

Testing Procedures

When the set up tasks are complete (“Setting Up the Test Environment” on page 7), you can begin NaviQuest testing. The initial testing establishes the base line test set against data sets that will never be DFSMS managed. Because they are not managed, they have an expected result of null (' ') for each storage class, storage group, data class, and management class.

Note: For these test cases, you must use the subtype prefix NEVR.

After the base line test is complete, you can test each phase, or cycle, of the DFSMS implementation, one subtype at a time. Normally, a DFSMS implementation phase is made up of either a single data type or several data subtypes. Each data

subtype is tested independently. Once each data subtype tests correctly, you can begin data conversion for DFSMS for that phase.

Use the following procedure to test your base line test set or any other phase.

Recommendation

Put all test cases into the initial base line test set. This will save you from having to repeat adding data types or subtypes one at a time.

1. Collect data set information for input.

With NaviQuest, you can create many test cases at once using input from the following sources:

- ISMF lists
- DCOLLECT data
- SMF data created by a storage class ACS exit
- VMA data

The data set test cases must all be representative of the data type that you will migrate to DFSMS management and require the same DFSMS services.

From the ISMF Primary Option menu, “Data Set” (option 1) offers you two ways to generate data sets samplings: (1) from a saved listing, or (2) from a new listing created from the criteria you specify, such as a VTOC or catalog.

Notes:

- a. Saving tables of temporary data sets might produce errors in bulk test case creation (“Test Case Generation from Saved ISMF List” option 11.1.1). We recommend that you generate SMF test cases from the ACSTST program provided by CBIPO for temporary data sets.
- b. The multivolume variable is always set to “Yes” in an ISMF table if the data set is not open at the time the table is saved. The value is set correctly at the time the data set is opened, which can sometimes cause errors in the bulk test case generator.
- c. Always set the ACQUIRE DATA FROM VOLUME and ACQUIRE DATA IF DFHSM MIGRATED options under the ISMF “Data Set Selection Entry” panel (ISMF option 1) to Y before generating the list.

After you have created the list, enter the ISMF SAVE command on the command line to save the list into a table. For information on the SAVE command, to *DFSMS/MVS V1R5 Using ISMF*. Also, see “Generate a Data Set List: ACBQBAI2” on page 33.

2. Generate the test cases.

Use the “Test Case Generation Selection Menu” panel (option 11.1) to turn the ISMF list, DCOLLECT data, SMS data generated by the storage class ACS exit, and VMA data into standard DFSMS test cases.

```
Panel Help
-----
ACBSFLG4 TEST CASE GENERATION SELECTION MENU
Enter Selection or Command ==> _____

Select the input data to be used and press Enter:

1 Saved ISMF List
2 DCOLLECT Data
3 SMF Data
4 VMA Extract Data

Use HELP Command for Help; Use END Command to Exit.
```

Figure 5. Test Case Generation Selection Menu, ACBSFLG4

ISMF Lists

To generate test cases from saved ISMF tables, select option 1 (Saved ISMF List) and then use the “Test Case Generation from Saved ISMF List Entry Panel” (option 11.1.1). Enter the following information:

- Saved list name previously saved in step 1 on page 9
- Member name prefix (subtype prefix)
- PDS that contains the test cases
- Whether you want to replace the existing test cases with the output test cases

Also select additional values that you want included in the test cases.

Note: If you do not enter a PDS name, NaviQuest will generate one based on the format *userid.Tnn.TESTCASE*. It is recommended that you specify a name so that the test case library conforms to your installation’s naming standards.


```

Panel Help
-----
ACBDFLG1 TEST CASE GENERATION FROM SAVED ISMF LIST ENTRY PANEL
Command ==> _____

To generate test cases, specify the following information and press Enter:
Saved ISMF List . . . . . _____ (Data set list)

Member Name Prefix . . . . _____ (1 to 4 alpha characters)
Test Case PDS . . . . . _____
Replace Existing Prefix . _ (Y or N)

ACS Test Case Variables:
Applic . . . . _____ DD . . . . . _____ More: +
Def_dataclas . . . . _____ Def_mgmtclas . . . . _____
Def_storclas . . . . _____ Filenum . . . . _____
Group . . . . . _____ Job . . . . . _____
Label . . . . . _____ Libname . . . . _____
Msvgp . . . . . _____ Pgm . . . . . _____
Retpd . . . . . _____ Storgrp . . . . _____

Use DOWN Command to Scroll Forward; Use UP Command to Scroll Backward;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 6. Test Case Generation from Saved ISMF List Entry Panel, ACBDFLG1

DCOLLECT Data

To generate test cases from DCOLLECT data, select option 2 (DCOLLECT Data) and then use “Test Case Generation from DCOLLECT Data Entry Panel” (option 11.1.2).

```

Panel Help
-----
ACBDFLG5 TEST CASE GENERATION FROM DCOLLECT DATA ENTRY PANEL
Command ==> _____

To generate test cases, specify the following information and press Enter:
Data Set Containing DCOLLECT Data
====> _____

Member Name Prefix . . . . _____ (1 to 4 alpha characters)
Test Case PDS . . . . . _____
Replace Existing Prefix . _ (Y or N)
Number of Test Cases . . . _____ (1 to 9999, blank=no limit)

Use HELP Command for Help; Use END Command to Exit.

```

Figure 7. Test Case Generation from DCOLLECT Data Entry Panel, ACBDFLG5

Enter the following information:

- Data set name
- Number of test cases you want included
- Member name prefix (subtype prefix) of the DCOLLECT test cases
- PDS that contains the test cases
- Whether to replace the existing test cases with the output test cases

Before you can use this function, you must have DCOLLECT data that includes D (data set) records.

SMF Data

To generate test cases from the ACSTST program, select option 3 (SMF Data) and then use “Test Case Generation from SMF Data Entry Panel” (option 11.1.3).

```
Panel Help
-----
ACBDFLG3      TEST CASE GENERATION FROM SMF DATA ENTRY PANEL
Command ==>> _____

To generate test cases, specify the following information and press Enter:
Data Set Containing SMF Data (Type 127 records)
==>> _____

Test Case PDS . . . . . _____

Note: The IGDACSSC storage class CBIPO exit must exist and the type
      127 SMF records must have been extracted. Program ACSTST from
      CBIPO must be available.

Use HELP Command for Help; Use END Command to Exit.
```

Figure 8. Test Case Generation from SMF Data Entry Panel, ACBDFLG3

Enter both the data set name containing the system management facility (SMF) data and the name of the test case PDS.

Note: This function requires that you have the IGDACSSC storage class CBIPO exit installed and have extracted the SMF type 127 records. The ACSTST program from the CBIPO package is also required.

VMA Extract Data

To generate a test case from a VMA extract file, select option 4 (VMA Extract Data) and then use the “Test Case Generation from VMA Extract Data Entry Panel” option (11.1.4).

```

Panel  Help
-----
ACBDFLG2  TEST CASE GENERATION FROM VMA EXTRACT DATA ENTRY PANEL
Command ==> _____

To generate test cases, specify the following information and press Enter:
Data Set Containing VMA Extract Data (GFTAXTR output)
====> _____

Member Name Prefix . . . . ____ (1 to 4 alpha characters)
Test Case PDS . . . . . _____
Replace Existing Prefix . - (Y or N)
Number of Test Cases . . . ____ (1 to 9999, blank=no limit)
Program Name to Filter on . _____

Note: Before running this function you must have run GFTAXTR from
your saved SMF type 14, 15, 21, and type 30 records.

Use HELP Command for Help; Use END Command to Exit.

```

Figure 9. Test Case Generation from VMA Extract Data Entry Panel, ACBDFLG2

Enter the following information:

- Name of the data set containing the VMA Extract data
- Number of test cases you want generated
- Member name prefix (subtype prefix) of the VMA test cases
- Program name, if you want to test the implementation for a particular program

Also include the name of the test case PDS and whether to replace the existing test cases.

Note: This function requires that you have already run GFTAXTR from your saved SMF records (types 14, 15, 21 and 30). JCL for GFTAXTR can be found in SYS1.SAMPLIB member GFTAXTRP.

Batch Options for Creating Test Cases

In addition, you can also use the following batch options to create test cases:

- For option 11.1.1, use the ACBQBAG3 CLIST. See “Generate Test Cases from ISMF-saved Data Set Lists: ACBQBAG3” on page 98.
- For option 11.1.2, use the ACBQBAG1 EXEC. See “Generate Test Cases from DCOLLECT Data: ACBQBAG1” on page 99.
- For option 11.1.3, use the ACBQBAG1 EXEC. See “Generate Test Cases from SMF Data” on page 100.
- For option 11.1.4, use the ACBQBAO3 EXEC. See “Generate Test Cases from VMA Extract Data: ACBQBAO3” on page 101.

Add a 1-to-4 character subtype prefix to each test case member. The prefix must be unique for each data subtype. For example, the first group of TSO data could have subtype prefix TSOA, the second TSOB, and so on.

See step 1 on page 9 for creating the ISMF table. ACBJBAG2, ACBJBAG1, ACBJBAOW, and ACJBBAI1 in SYS1.SACBCNTL JCL library can perform this task in batch (see “How to Run Storage Administration Tasks in Batch” on page 25).

3. Make ACS routine and construct changes.

You must change (but not activate) the ACS code and constructs to reflect the new phase of implementation that you want to test. Before you change the ACS code and construct definitions contained in the source control data set (SCDS), save the old source in case it is needed for recovery.

For information on recovering the ACDS, refer to *DFSMS/MVS V1R5 DFSMSdfp Storage Administration Reference*.

You can now update the ACS routines to reflect the new data subtype you want migrated to DFSMS.

4. Update the FILTLISTS.

When the ACS code is changed, you might want to use the COPYFILT function of NaviQuest to update all the ACS routines from a common definition of the filter lists. You will be prompted to provide a change log entry that reflects changes you are making to the ACS routines. This entry will be automatically placed into the change log in the ACS routines.

To use the COPYFILT macro, see “COPYFILT Macro: COPYLIB Facility for FILTLISTS” on page 140.

5. Translate and validate the ACS routines.

You must translate and validate (but not activate) the ACS routines. The ISMF translate function transforms ACS routines into a table format. Translation checks for syntax errors and transforms the ACS routines into a format suitable for input to the validation.

The ISMF validate function verifies that all possible constructs that can be assigned via the ACS logic have been defined to the SCDS used for testing. ACS routines must be translated before they can be validated; however, validation of ACS routines is optional.

To translate and validate, you can either use the online ISMF functions or you can use the NaviQuest ISMF-in-batch EXEC.

For online translation and validation, choose option 7 (ACS Class Selection) from the ISMF Primary Option menu. To translate, choose option 2 (Translate). To validate, choose option 3 (Validate).

To use the translate facility in batch, see “ACS Routine Translate: ACBQBAO1” on page 84. To use the validation facility in batch, see “SCDS Validation: ACBQBAO2” on page 88.

For further information on translation or validation, refer to *DFSMS/MVS V1R5 DFSMSdfp Storage Administration Reference*.

6. Run the test cases.

Create a **new** ACS listing by using the ISMF “Test ACS Routines” option (7.4.3). The testbed library contains the test cases. Specify an asterisk (*) to run *all* test cases in the library.

The **new** ACS listing represents the SMS configuration *after* the ACS routines have been changed for the new data subtype.

Notes:

- a. We recommend including the prefix of the subtype tested in the ACS listing data set name, to make it easier to identify which data subtype the listing represents.
- b. We also recommend running test cases in batch whenever possible. To run option 7.4.3 in batch, see “Test ACS Routines: ACBQBAIA” on page 90.

7. Compare the results of the regression testing.

After the base line test, every test includes both testing of new data subtypes and regression testing of previously tested data subtypes, including the base line test set.

At this time, you use the NaviQuest ACS comparison test function to compare the results of all test cases in the testbed library with their expected results. The ACS comparison test produces a report of exceptions. Since you have not yet stored the expected results of this data subtype’s test cases, these test cases appear as exceptions. Later, in step 9 on page 18, you will store the expected results for the current data subtype test cases. But for now, the exceptions you get are either these valid initial (that is, first run) test cases, or they are errors.

To run the ACS comparison test, choose option 2 from the NaviQuest Primary Option Menu.

```
Panel Help
-----
ACBSMDP0  ENHANCED ACS MANAGEMENT - NaviQuest PRIMARY OPTION MENU
Enter Selection or Command ==> _____

Select one of the following options and press Enter:

 1 Test Case Generation
 2 ACS Test Listings Comparison
 3 Enhanced ACS Test Listing
 4 Test Case Update with Test Results
 5 SMS Report Generation
 6 Model Commands Generation
 7 Batch Testing/Configuration Management
 X Exit

Use HELP Command for Help; Use END Command or X to Exit.
```

Figure 10. NaviQuest Primary Option Menu, ACBSMDP0

On the “ACS Comparison Report” panel, enter the following information:

- Name of your base test case results
- Name of your new test case results
- PDS that contains the test cases
- PDS that contains the exception test cases
- Name of the comparison results data set

```

Panel Help
-----
ACBDFLC1      ACS TEST LISTINGS COMPARISON ENTRY PANEL
Command ==> _____

To compare ACS listings, specify the following information and press Enter:
Input Data Sets:
  Base ACS Test Listing (Before latest ACS routine changes)
  ==> _____
  New ACS Test Listing (After latest ACS routine changes)
  ==> _____
Reference Data Set for Compare:
  Test Case PDS (Test source for listings above)
  ==> _____

Output Data Sets:
  Comparison Results Data Set (Summary of exception test cases)
  ==> _____
  Replace Contents if DSN Exists . . _ (Y or N)
  Exception Test Case PDS (Contents of exception test cases)
  ==> _____
  Replace Contents if DSN Exists . . _ (Y or N)
  ==> _____

Use HELP Command for Help; Use END Command to Exit.

```

Figure 11. ACS Test Listings Comparison Entry Panel, ACBDFLC1

After running “ACS Test Listings Comparison Entry Panel” (option 11.2), you must verify the following items:

- The number of exceptions should be the same as the number of test cases you are currently testing.
- Exceptions should all have the same subtype prefix.
- Each listed test case should have the listed results that you expect.

If changes have been made correctly to the ACS routines, the differences between the two should be only the data subtype that is being initially tested.

Specify a comparison data set name to be used to store the results of the comparison. Also input whether you want to write over the data set specified if it already exists. If N is specified, and the data set name already exists, an error message will be returned. If Y is specified, the data set will be deleted, a new data set with the same name will be allocated, and the report will be written to this data set. Then press the Enter key.

You will be automatically placed into ISPF “browse” when the comparison completes. The comparison data set you are browsing lists only the test cases identified as exceptions.

If exceptions other than the test cases for the subtype you are initially testing are listed, you have probably made an error in coding the revisions to your ACS routines. Changes in coding that have caused errors must be corrected before you can proceed. This means repeating the operations until the test cases match the exceptions.

The following files are created or updated as output:

- Exception PDS
- Comparison data set

Important

Each test is an initial test for one data subtype but may include many regression tests for previously tested data subtypes. Expected values are not stored in the initially tested data subtype until its testing completes successfully.

The ACS comparison performed in step 7 has two functions:

- It validates the regression tests.

Current test results of each previously tested data subtype should match the saved expected results previously stored with the test cases. If the results are the same, the regression test is successful. If the results differ, there is an error in the new ACS logic; that is, the ACS routine is assigning different values.

- It indicates the subtype test set that is being initially tested.

Because this is an initial test, this test case has no expected results stored in the test cases, other than null. Thus, during the comparison in step 7 on page 15, all test cases for this new data subtype show an exception; that is, new results will no longer be null.

For more information about running the ACBQFLC1 EXEC in batch, see “ACS Test Listings Comparison: ACBQBAC1” on page 102.

8. Validate the test results and determine errors.

You must manually compare the new test cases to their expected results for the single data subtype that has been initially tested. This comparison determines if there are initial test errors. If the exceptions contain any test cases from the data subtypes previously tested correctly (in regression testing), these exceptions are also errors.

It is the manual verification of the results that makes sure that the values are the expected results. When all test cases are correct, the test values are stored in the test cases as save expected results, to be used for later regression testing.

If you find errors, you can generate the NaviQuest ACS cross-reference report for additional information about the specific test cases that produced the errors. Use this report to help you debug the ACS logic. If you find errors (from either step 7 or step 8), you must correct the ACS code before returning to step 3 on page 14 and retest until the data subtype results have no errors.

If you do not find errors, the test is complete, as all the test cases in the subtype test set have the correct expected results.

To create an ACS cross-reference report, choose “Enhanced ACS Test Listing Entry Panel” (option 3) from the NaviQuest Primary Option Menu (see Figure 2 on page 2). On the “Enhanced ACS Test Listing Entry Panel,” fill in the fields with the following names:

- ISMF test case listing (generated through option 7.4.3)
- Data set for cross-reference listing (name of data set to contain cross-reference)

```

Panel Help
-----
ACBDFLX1          ENHANCED ACS TEST LISTING ENTRY PANEL
Command ==> _____

To generate enhanced ACS test listing, specify the following and press Enter:
ACS Test Listing
==> _____

Enhanced ACS Test Listing
==> _____
Replace Contents if DSN Exists . . _ (Y or N)

Fields to Include in Enhanced ACS Test Listing: (Y or N)

DSN . . . . . _      JOBNAME . . . . . _
EXPDT . . . . . _   SIZE . . . . . _
UNIT . . . . . _    PROGRAM . . . . . _

Use HELP Command for Help; Use END Command to Exit.

```

Figure 12. Enhanced ACS Test Listing Entry Panel, ACBDFLX1

Indicate whether the specified data set should be written over if it already exists. If N is specified, and the data set name already exists, an error message will be returned. If Y is specified, the data set will be deleted, a new data set with the same name will be allocated, and the report will be written to this data set.

Specify with a Y or an N which variables you want included in your report. Once you have specified all variables that you want, press the Enter key and the report will be produced.

9. Save the expected results.

Once the subtype test is correct, you can use NaviQuest to place the results of the test (that is, the expected results for later regression testing) into the test case definition as the saved expected results for later regression testing. Test results are only saved after all test cases in the subtype test set have completed with the expected results for that data subtype. The saved expected results will be used for later regression testing, as explained in step 7 on page 15.

To save the test results, choose “Test Case Update With Test Results Entry Panel” (option 4) from the ISMF Primary Option menu, which takes you to the “Test Case Update With Test Results Entry Panel.”


```

Panel  Help
-----
ACBDFLU1      TEST CASE UPDATE WITH TEST RESULTS ENTRY PANEL
Command ==> _____

To update test cases with test results, specify the following and press Enter:

Input Data Sets:
  New ACS Test Listing (After latest ACS routine changes)
  ==> _____
  Comparison Results Data Set (Summary of exception test cases)
  ==> _____
  Exception Test Case PDS (Contents of exception test cases)
  ==> _____

Input/Output Data Set:
  Test Case PDS
  ==> _____

Use HELP Command for Help; Use END Command to Exit.

```

Figure 13. Test Case Update with Test Results Entry Panel, ACBDFLU1

Enter the names of the testbed PDS library, the exception test case PDS, the PDS created in the ACS comparison report, and the new ACS test case listing.

The test case members for the exceptions are read and copied into the testbed library. The saved expected results are obtained from the comparison report and are also saved in the testbed library.

You have now completed testing for this data subtype and can now start testing the next data subtype.

Delete the following data sets at the end of this step:

- Comparison report generated in step 7
- Exception PDS created in step 7
- Base and new ACS listing can be deleted (or printed and deleted)

For more information about running the ACBQFLU1 EXEC in batch, see “Update Test Cases with Expected Results: ACBQBAU1” on page 106.

10. Test the next data subtype in the current phase.

Continue NaviQuest testing for each data subtype in the current DFSMS implementation phase. This testing either repeats Steps 1 through 10 or repeats Steps 3 through 10, depending on whether all subtype test sets are initially placed into the testbed.

After the initial test of the base line, all additional tests include regression testing along with initial testing.

11. Activate your new DFSMS configuration.

Once an entire phase (that is, all the subtypes within the implementation phase) have tested correctly, you can activate the new configuration by using the SETSMS command at an MVS console.

For more information on activating your configuration, refer to the *DFSMS/MVS V1R5 DFSMSdfp Storage Administration Reference*.

You might want to use the NaviQuest reporting capabilities to determine the amount of DASD space required to convert the data in each phase, prior to

attempting conversion. Use this information to ensure that enough DASD is available for the conversion.

12. Convert data to SMS management.

After activation of your new configuration, you can now migrate the data to DFSMS management. There are several options for doing this data migration:

- DFDSS COPY
- DUMP/RESTORE
- Normal allocation processing
- MIGRATE/RECALL
- CONVERTV

NaviQuest Testing Scenario

This customer, initially running with a minimal configuration, plans to convert two data types to SMS management in two phases, one for each data type. The following scenario helps clarify the various testing phases.

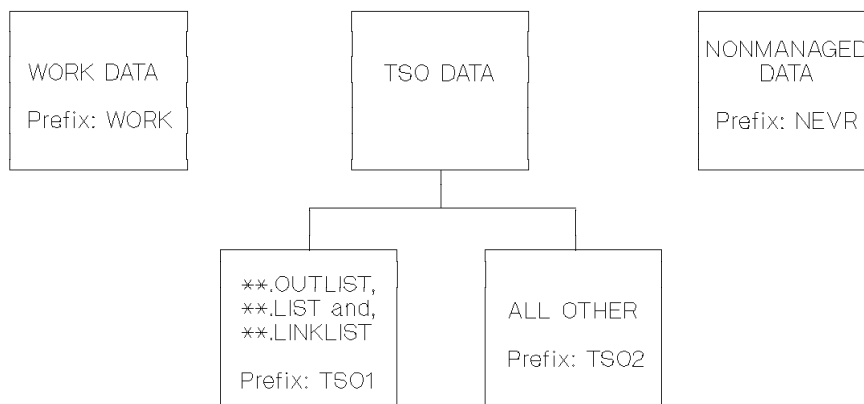
Data Testing

In this example, the customer wants to test two data types with NaviQuest: work data and TSO data. Because these two data types are the initial data types to be tested, establishing and testing non-SMS data must be done first to create the base line test.

The two data types are shown below with typical data classification assignments. Work data is made up of a single data subtype, each work data set having a single expected result from the ACS routines. TSO data is made up of two data subtypes, each subtype having a different set of expected results from the ACS routines.

The subtype prefix assigned for the work data type is WORK. The subtype prefix for the two TSO data subtypes are TSO1 and TSO2.

There is also a data type for all nonmanaged data. This data type is used to create the baseline test cases and must be assigned the subtype prefix NEVR.



Method of Testing

Testing with NaviQuest can be done with a sampling of data sets from each data subtype. In the following testing example, however, only two or three data sets are shown.

All test cases are built *before* any NaviQuest testing begins. As NaviQuest testing begins, there are four prefixes in the test bed library: NEVR, WORK, TSO1, and TSO2, with NEVR representing non-SMS-managed (base line) data sets.

NaviQuest is used to create the test cases from data set lists for each of the four subtype test sets, in this scenario.

Testing Example

Figure 14 on page 22 presents a flowchart of three data types as they progress through the NaviQuest testing phases.

Notes:

1. The term *systemp* means “system temporary data sets.” These are the WORK data sets.
2. The symbol (' ') means “null.”
3. Circled numbers represent corresponding steps in the testing procedure.

Refer to Figure 14 on page 22 for the following discussion.

Pass 1 establishes the testing *base line*. Data that is never to be SMS-managed must be included as test cases used in this pass. Additional test cases can also be included; in this example, all planned test data is included in pass 1. The objective of pass 1 is to set the expected results of all base line test cases to *nulls*. The null results establish a set of comparison values so that when future test runs (pass 2) are made, the storage administrator can see where changes occur by identifying the expected results that are no longer nulls.

Pass 2 tests the first set of data to be SMS-managed—in this example, temporary data sets (work data). The storage administrator makes logic changes to the test ACS routines to manage temporary data sets. These changes mean that the expected results for temporary data sets will no longer be null. During pass 2, the NaviQuest comparison function compares the results with the expected results that were generated and saved in pass 1. If the pass 2 tests are error-free, all the temporary data sets have expected results of nonnulls and they appear as the only exceptions in the comparison report. If pass 2 tests successfully, the storage administrator can save the expected results and begin managing temporary data sets by activating the test ACS routines.

Note: TSO data has two data subtypes: output listings and all other TSO data. Consequently, two passes are needed to complete the TSO testing.

Pass 3 tests the initial part of TSO data to be managed: list output data sets. Again the test ACS routines are changed to now manage both temporary data sets and TSO list output data sets. If pass 3 testing runs error-free, the comparison step has only the test cases for TSO list output. All other test cases have null results or have the expected results of the temporary data sets. If the only exceptions are the TSO list output test cases, pass 3 testing is successful and the expected results are saved. However, the TSO list data is not yet SMS-managed.

Pass 4 tests the rest of the TSO data. The test ACS routines are changed for the additional TSO data. This test *must* complete successfully before any TSO data is converted to SMS management. Again if pass 4 testing runs error free, only the new TSO test cases are identified as exceptions, NaviQuest saves these additional TSO test case expected results, and the storage administrator activates the test ACS configuration. All TSO data can then be converted to SMS management.

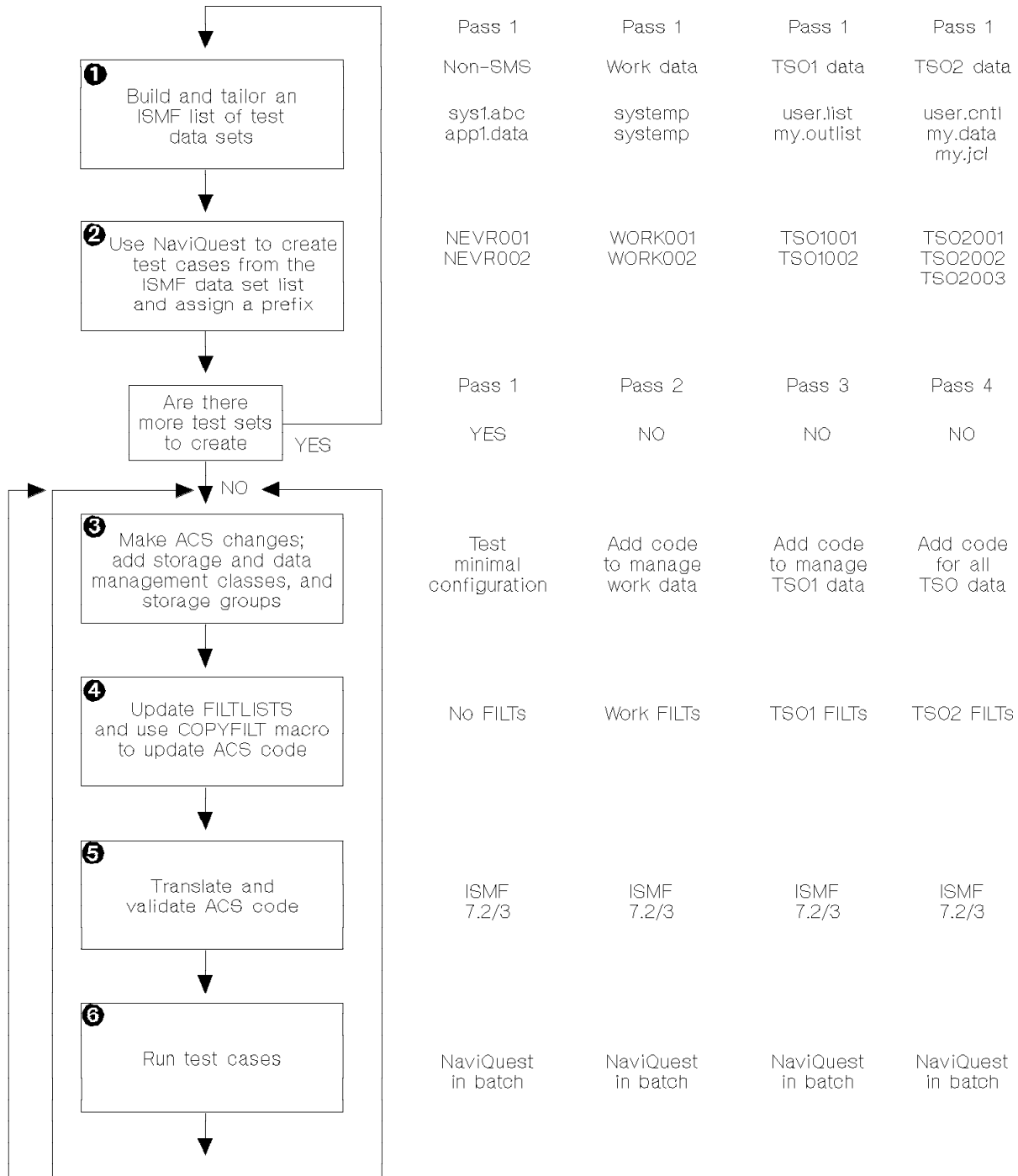


Figure 14 (Part 1 of 2). Testing Example Flowchart

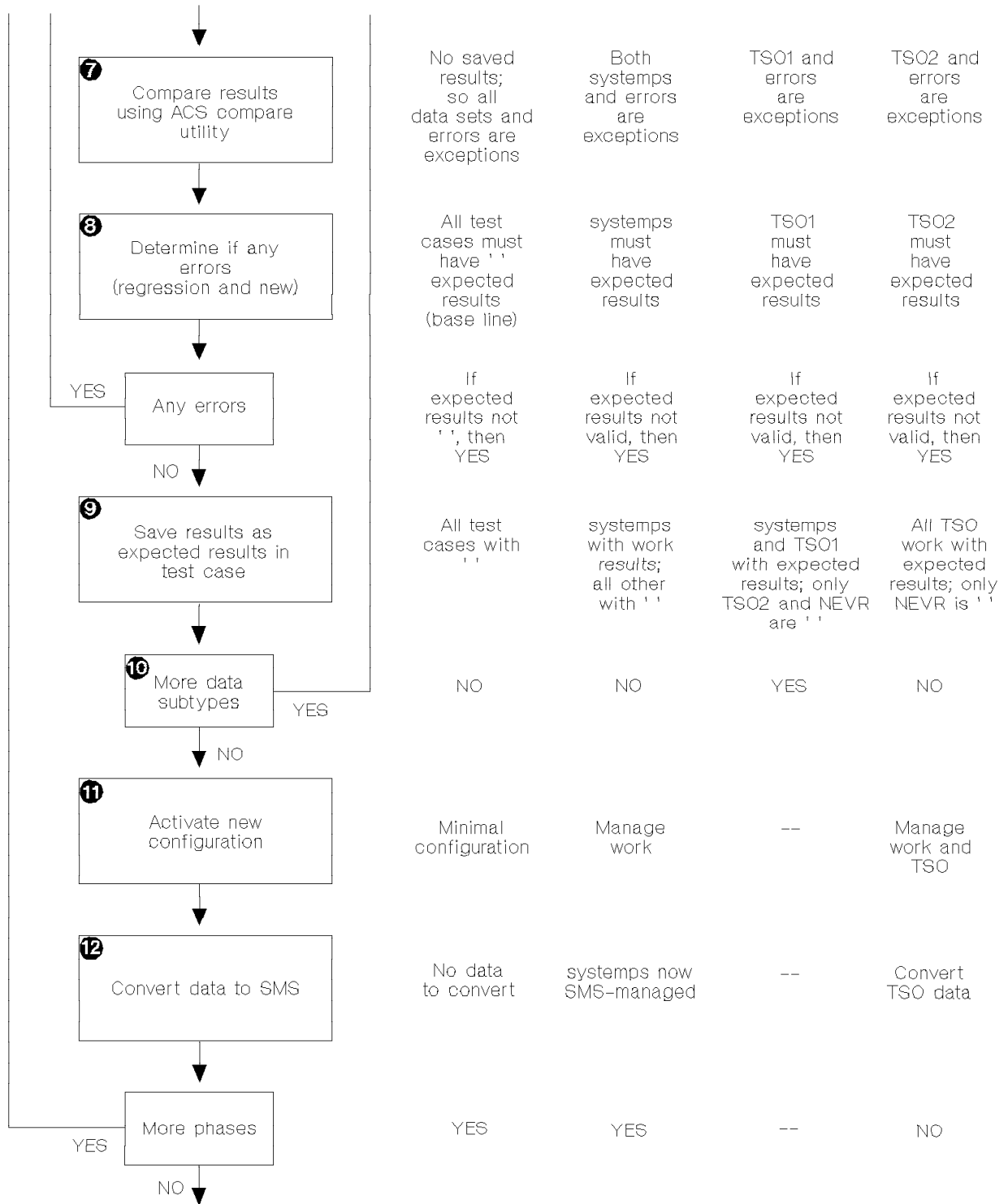


Figure 14 (Part 2 of 2). Testing Example Flowchart

Chapter 3. Performing Storage Administration Tasks in Batch

Storage administration tasks that are performed using ISMF options can also be done in batch with JCL, CLISTs, and REXX EXECs that are provided by NaviQuest.

For example:

- Testing the DFSMS configuration
- Performing data set and volume maintenance activities
- Diagnosing data set and volume problems

How to Run Storage Administration Tasks in Batch

To run in batch mode the user is required to:

- Have a TSO user profile
- Know the specific ISMF option that will be run in batch and the task it performs
- Provide the parameters that describe each task. These parameters can be modified by the user.
- Know the required ISPF statements. *Do not modify these ISPF statements.*

ISMF Option 11.7 Batch Testing and Configuration Management

ISMF Option 11.7 (Batch Testing/Configuration Management Selection Menu) lets you select and run batch jobs from an ISMF panel.

```
Panel Help
-----
ACBSMDJ1  BATCH TESTING/CONFIGURATION MANAGEMENT SELECTION MENU
Enter Selection or Command ==> _____

Select an option or enter Data Set to Edit and press Enter:

  1 Saved ISMF List Operations Batch Samples
  2 DCOLLECT Data Operations Batch Samples
  3 Configuration Changes Batch Samples
  4 VMA and SMF Batch Samples

Data Set to Edit . . . _____

Use HELP Command for Help; Use END Command to Exit.
```

Figure 15. Batch Testing/Configuration Management Selection Menu, ACBSMDJ1

1. Select an option and press Enter. You will be advanced to a menu of batch samples.

Or fill in the “Data Set to Edit” field with the name of the data set containing JCL you want to run. Press Enter. You will be placed in ISPF EDIT mode. Go to step 3.

2. Select one of the batch sample options or fill in the "Data Set to Edit" field with the name of the data set containing JCL you want to run. Press Enter. You will be placed in ISPF EDIT mode.
3. Edit the JCL if needed. While in ISPF EDIT, type SUBMIT at the command line and press ENTER to run the JCL.

Refer to Sample JCL for Batch, step 2, for information on modifying the JCL samples shipped with NaviQuest.

4. To save the edited JCL, press PF3 or the END command while in ISPF Edit. Fill in the fields and press Enter.

Sample JCL for Batch

NaviQuest provides sample JCL in the SYS1.SACBCNTL library. The JCL can be modified with the parameters for the task that is to be performed. Table 1 on page 27 lists SYS1.SACBCNTL members.

To use the JCL that is provided in SYS1.SACBCNTL:

1. Copy the SYS1.SACBCNTL library member that contains the sample JCL for the task that is to be run in batch. See Table 1 on page 27 for a list of members and the tasks they perform.
2. Update the JCL with the appropriate parameters for that task, after DD statement:

```
//SYSTSIN DD *
```

The commands, syntax, and parameters follow the discussion of each task, on pages 33 through 126.

Notes:

- a. Each member contains sample syntax and parameters. Change only the jobcard and the syntax and parameters. *Do not change any other JCL.*
- b. ACBJBAOB is called by the other SYS1.SACBCNTL library members during batch processing. *Do not modify ACBJBAOB when it is called by the other JCL members of SYS1.SACBCNTL.*

Figure 16 on page 29 shows the JCL in ACBJBAOB.

ISPSTART Batch Parameters for NaviQuest

The following ISPSTART batch parameters have been coded with the appropriate values for NaviQuest. Refer to the *ISPF V4R2 Dialog Developer's Guide and Reference* for additional information about these parameters.

BATSCRD Screen depth

BATSCRW Screen width

BDISPMAX Maximum number of panel displays for a session

Note: This value represents the total number of panel display calls. This value is coded to avoid loops.

BREDISPMAX Maximum number of times the same panel can be displayed

Note: The job will terminate if this limit is reached. This value is coded to avoid loops.

SYS1.SACBCNTL Sample JCL Library

The following table lists the sample job library members:

<i>Table 1 (Page 1 of 2). SYS1.SACBCNTL Sample JCL Library Member List</i>	
Member	Function
ACJBAA1	Aggregate group define/alter/display
ACJBAB1	Base configuration define/alter/display
ACJBAC1	ACS comparison report
ACJBAC2	Translate ACS routines, validate SCDS, test ACS routines, and generate ACS comparison report
ACJBAD1	Data class define/alter/display
ACJBAG1	Generate test cases from previously collected DCOLLECT data ('D' records)
ACJBAG2	Generate test cases from a previously saved table (data set list)
ACJBAA11	Generate test cases from CBIPO exit (extract type 127 records from SMF data using IFASMFDP and generate test cases from them using ACSTST program)
ACJBAA12	Generate data set list and save it in a table
ACJBAA14	Generate DASD volume list, save it in a table, and save the query
ACJBAA15	Generate DASD volume list and save it in a table
ACJBAA17	Generate data set list, save it in a table, and save the query
ACJBAA18	Generate DASD volume list, save it in a table, and generate a report from it
ACJBAA19	Generate DASD volume list using a previously saved query and save it in a table
ACJBAA1A	Generate ISMF mountable tape volume list, save it in a table, and generate report from it
ACJBAA1B	Alter storage group volume status
ACJBAA1C	Test ACS routines
ACJBAA1D	Generate mountable tape volume list and save it in a table
ACJBAA1H	Generate data set list using a previously saved query and save it in a table
ACJBAA1J1	Management class define/alter/display
ACJBAA1J2	Pool storage group define/alter
ACJBAA1J3	Tape storage group define/alter
ACJBAA1J7	Object backup storage group define/alter
ACJBAA1J8	VIO storage group define/alter
ACJBAA1JA	Dummy storage group define/alter
ACJBAA1JB	Object storage group define/alter
ACJBAA1M1	Generate model command from saved ISMF table (data set list)
ACJBAA1M2	Model commands from DCOLLECT data
ACJBAA1OD	Generate data set report from a previously saved table (data set list)
ACJBAA1OF	Generate volume report from a previously saved table (DASD volume list)

Table 1 (Page 2 of 2). SYS1.SACBCNTL Sample JCL Library Member List

Member	Function
ACBJBAOQ	Translate ACS Routines
ACBJBAOS	Validate SCDS
ACBJBAOT	Generate tape report from a previously saved table (tape volume list)
ACBJBAOU	Generate data set list, save it in a table, and generate report from it
ACBJBAOW	Generate test cases from VMA extract file
ACBJBAO7	Generate DCOLLECT output
ACBJBAR2	Generate SMS configuration report From DCOLLECT data
ACBJBARD	Generate data set report from DCOLLECT data
ACJBAS1	Storage Class define/alter/display
ACJBBAU2	Update test cases test cases with expected results
ACJBBAU4	Job to customize the job card for all NaviQuest jobs
ACJBAXV	Generate volume report from DCOLLECT data
ACJBAX1	ACS cross reference Report

ACBJBAOB JCL

Figure 16 shows the JCL for ACBJBAOB.

```
//ACBJBAOB PROC CLIST1='SYS1.DGTCLIB',
/*-----*/
/* CLIST1 SHOULD BE THE FILE #1 FROM THE INSTALLATION TAPE */
/* NOTE THAT THIS IS A FB CLIST LIBRARY; IF YOUR INSTALLATION */
/* USES VB CLIST LIBRARIES, YOU MUST CONVERT THEM YOURSELF */
/*-----*/
//          PLIB1='SYS1.DGTPLIB',
/*-----*/
/* PLIB1 SHOULD BE THE FILE #3 FROM THE INSTALLATION TAPE */
/*-----*/
//          LOAD1='SYS1.DGTLLIB',
/*-----*/
/* LOAD1 SHOULD BE THE FILE #5 FROM THE INSTALLATION TAPE */
/*-----*/
//          MLIB1='SYS1.DGTMLIB',
/*-----*/
/* MLIB1 SHOULD BE THE FILE #6 FROM THE INSTALLATION TAPE */
/*-----*/
//          TABL2='userid.TEST.ISPTABL'
/*-----*/
/* TABL2 IS THE DATA SET FOR SAVING ISMF TABLES; YOU SHOULD */
/* ALLOCATE THIS DATA SET WITH THE SAME DCB PARAMETERS AS THE */
/* ISMF DGTLLIB DATA SET; TABLE CAN BE LARGE - ALLOCATE A LARGE */
/* DATA SET. Be sure you change 'userid' to YOUR userid. */
/*-----*/
/*****
/*$MAC(ACBJBAOB) COMP(5695DF123): BATCH - CALLED PROC */
/* */
/* PROPRIETARY V3 STATEMENT */
/* LICENSED MATERIALS - PROPERTY OF IBM */
/* 5695-DF1 */
/* (C) COPYRIGHT 1995,1998 IBM CORP. */
/* END PROPRIETARY V3 STATEMENT */
/* */
/* CHANGE ACTIVITY: */
/* */
/*$K0=NAVIQUEST,HACS110,95/08/04,SNJADR: INITIAL VERSION @K0A*/
/*$K1=KNQ0003,HACS110,95/08/04,SNJTCS: COMMENT CORRECTION @K1C*/
/*02/29/96 TCS - HACS110: REMOVED SEQUENCE NUMBERS @WA18945*/
/*$L0=NAVIQUEST,HACS120,96/06/18,SNJTCS: RELEASE 2 CHANGES @L0A*/
/*$L1=KN20035,HACS120,96/07/12,SNJTCS: TYPO CORRECTED @L0A*/
/*10/14/96 CCY - HDZ11C0: Ship as SPE @WA22861*/
/*06/13/97 CCY - HDZ11D0: Change IBMUSER to 'userid' @WA27246*/
/*$T1=NQ5,HDZ11E0,061997,SNJTCS: ISP V3R5M0 Datasets deleted @T1A*/
/*08/27/98 TCS - HDZ11D0: BLKSIZE=0 fix @WA34895*/
/* */
/*****
```

Figure 16 (Part 1 of 2). SYS1.SACBCNTL member ACBJBAOB JCL for Batch Functions

```

/*****
/* PROC STEP STEP1 - INVOKES IKJEFT01 */
/*****
//STEP1 EXEC PGM=IKJEFT01,DYNAMNBR=50,REGION=(6144K),TIME=(300)
//*-----*/
//STEPLIB DD DSN=&LOAD1,DISP=SHR
//*-----*/
/* IN ISPPLIB, INCLUDE THE ISPF, PDF, ISMF AND NAVIQUEST PANELS */
//*-----*/
//ISPPLIB DD DSN=&PLIB1,DISP=SHR,BLKSIZE=0
// DD DSN=ISP.SISPPENU,DISP=SHR /* ISPF PANELS */
//* DD DSN=SYS1.DGTPLIB,DISP=SHR /* ISMF PANELS */
//*-----*/
/* IN ISPMLIB, INCLUDE THE ISPF, PDF, ISMF AND NAVIQUEST MESSAGES */
//*-----*/
//ISPMLIB DD DSN=&MLIB1,DISP=SHR,BLKSIZE=0
// DD DSN=ISP.SISPMENU,DISP=SHR /* ISPF MESSAGES */
//* DD DSN=SYS1.DGTMLIB,DISP=SHR /* ISMF MESSAGES */
//*-----*/
/* IN ISPSLIB, INCLUDE THE ISPF, PDF AND ISMF SKELETONS */
//*-----*/
//ISPSLIB DD DSN=ISP.SISPSENU,DISP=SHR /* ISPF SKELETONS */
// BLKSIZE=0
// DD DSN=SYS1.DGTSLIB,DISP=SHR /* ISMF SKELETONS */
//*-----*/
/* IN ISPTLIB, INCLUDE THE ISPF, PDF, ISMF AND NAVIQUEST TABLES */
//*-----*/
//ISPTLIB DD DSN=&&TEMP,DISP=NEW,UNIT=3380,BLKSIZE=0,
// SPACE=(TRK,(1,1,1)),DCB=(ISP.SISPTENU)
// DD DSN=&TABL2,DISP=SHR
// DD DSN=ISP.SISPTENU,DISP=SHR /* ISPF TABLES */
// DD DSN=SYS1.DGTTLIB,DISP=SHR /* ISMF TABLES */
//*-----*/
/* IN SYSPROC, INCLUDE THE ISPF, PDF, ISMF AND NAVIQUEST CLISTS */
//*-----*/
//SYSPROC DD DSN=&CLIST1,DISP=SHR,BLKSIZE=0
// DD DSN=ISP.SISPCLIB,DISP=SHR /* ISPF CLISTS */
//* DD DSN=SYS1.DGTCLIB,DISP=SHR /* ISMF CLISTS */
//*-----*/
//ISPTABL DD DSN=&TABL2,DISP=SHR,BLKSIZE=0
//SYSPRINT DD SYSOUT=(,)
//SYSUDUMP DD SYSOUT=*
//SYSTSPRT DD SYSOUT=(,),OUTLIM=20000
//ISPLLOG DD SYSOUT=(,),DCB=(LRECL=125,BLKSIZE=129,RECFM=VA)
//ISPPROF DD DSN=&&PROF,DISP=(NEW,DELETE,DELETE),
// DCB=(ISP.SISPTENU),SPACE=(TRK,(1,1,1)),UNIT=3380
// PEND

```

Figure 16 (Part 2 of 2). SYS1.SACBCNTL member ACBJBAOB JCL for Batch Functions

NaviQuest CLISTS and REXX EXECs

The CLISTS and REXX EXECs that are called by the sample JCL shipped with NaviQuest, can also be called by JCL that *you* code, to perform storage administration tasks. Your JCL must call the appropriate CLIST or REXX EXEC for the task that is to be performed. Use Table 2 to find the CLIST or REXX EXEC you want to use, then update the ISPSTART statement with its name and parameters.

Refer to the JCL sample jobs in SYS1.SACBCNTL to see the complete syntax for each task.

Notes:

1. It is recommended that you **do not** modify these REXX EXECs and CLISTS.
2. The REXX EXECs and CLISTS do not create a listing; they create an ISMF-saved table (except the test ACS routines task), which is similar to running ISMF interactively and then issuing a SAVE 'xxxxxxx' command. Save the table, then use the ISMF 11.5.n options to produce a flat file of the table for printing.

Table 2 (Page 1 of 2). CLISTS and REXX EXECs for Storage Administration Tasks

Interactive ISMF Option	Storage Administration Task	CLIST or REXX EXEC	Page
1	Generate data set list	ACBQBAl2	33
2.1	Generate DASD or optical device volume list	ACBQBAl4	38
2.3	Generate tape volume list	ACBQBAl6	42
3.2	Display management class	ACBQBAl1	44
3.3	Define management class	ACBQBAl1	44
3.4	Alter management class	ACBQBAl1	44
4.2	Display data class	ACBQBAD1	52
4.3	Define data class	ACBQBAD1	52
4.4	Alter data class	ACBQBAD1	52
5.2	Display storage class	ACBQBAS1	61
5.3	Define storage class	ACBQBAS1	52
5.4	Alter storage class	ACBQBAS1	61
6.2	Define dummy storage group	ACBQBAlA	65
	Define object backup storage group	ACBQBAl7	66
	Define object storage group	ACBQBAlB	69
	Define pool storage group	ACBQBAl2	72
	Define tape storage group	ACBQBAl3	76
	Define VIO storage group	ACBQBAl8	79

Table 2 (Page 2 of 2). CLISTs and REXX EXECs for Storage Administration Tasks

Interactive ISMF Option	Storage Administration Task	CLIST or REXX EXEC	Page
6.3	Alter dummy storage group	ACBQBAJA	65
	Alter object backup storage group	ACBQBAJ7	66
	Alter object storage group	ACBQBAJB	69
	Alter pool storage group	ACBQBAJ2	72
	Alter tape storage group	ACBQBAJ3	76
	Alter VIO storage group	ACBQBAJ8	79
6.4	Storage group add/delete volume	ACBQBAl9	82
7.2	ACS routine translate	ACBQBAO1	84
7.3	SCDS validation	ACBQBAO2	88
7.4.3	Test ACS Routines	ACBQBAIA	90
8.1	Display base configuration	ACBQBAB1	92
8.2	Define base configuration	ACBQBAB1	92
8.3	Alter base configuration	ACBQBAB1	92
9.2	Display aggregate group	ACBQBAA1	95
9.3	Define aggregate group	ACBQBAA1	95
9.4	Alter aggregate group	ACBQBAA1	95
11.1.1	Test cases from ISMF-saved list	ACBQBAG3	98
11.1.2	Test cases from DCOLLECT data	ACBQBAG1	99
11.1.3	Test cases from SMF data	None	100
11.1.4	Test cases from VMA extract data	ACBQBAO3	101
11.2	ACS test listings comparison	ACBQBAC1	102
11.3	Enhanced ACS test listing	ACBQBAX1	104
11.4	Update test cases with expected test results	ACBQBAU1	106
11.5.1	Data set report from ISMF-saved list	ACBQBAR1	107
11.5.2	DASD volume report from ISMF-saved list	ACBQVAR1	110
11.5.3	Tape volume report from ISMF-saved list	ACBQBAR4	113
11.5.4	Data set report from DCOLLECT data	ACBQBAR7	116
11.5.5	DASD volume report from DCOLLECT data	ACBQBAR6	120
11.5.6	SMS configuration report from DCOLLECT data	ACBQBAR8	123
11.6.1	Model command from ISMF-saved list	ACBQBAM1	125
11.6.2	Model command from DCOLLECT data	ACBQBAM2	126

Generate a Data Set List: ACBQBAI2

ACBQBAI2 is called by the following SYS1.SACBCNTL members to generate the data set list in batch:

ACBJBAI2 Generate data set list and save it in a table
 ACBJBAI7 Generate data set list, save it in a table, and save the query
 ACBJBAOU Generate data set list, save it in a table, and generate report

See Figure 17 on page 36 for the sample JCL and parameters.

Save or Delete the Data Set List: You can specify whether or not the ISMF table should be saved after it has been generated.

ISPSTART CMD(ACBQBAI2 SAVE|DELETE *tablnm parameters ...*)

SAVE *tablnm* Indicates that the generated data set list is to be saved in data set *tablnm* with the specified parameters.

DELETE Indicates that the specified data sets in the generated list are to be deleted.

Attention: Use the DELETE option carefully. Before you use this option, generate the table (with the SAVE option), print it, and then examine the table to see which data sets are to be deleted.

Use the following parameters with ACBQBAI2. At least one OP and one value should be included when specifying a parameter.

Parameters	Description
ALLOCSP(OP1 nnn1 BOOL OP2 nnn2)	Where 'OP1' is EQ, NE, GT, LE, 'nnn1' is the allocate space value, in kilobytes (KB); BOOL is AND or OR; OP2 has the same values as OP1; and 'nnn2' has the same values as nnn1.
BLKSIZE(OP1 nnn1 BOOL OP2 nnn2)	Where 'OP1' and 'OP2' are EQ, GT, LE, or NE; 'nnn1' and 'nnn2' are the block size values; BOOL is AND or OR.
BLKUNUSED(OP1 nnn1 BOOL OP2 nnn2)	Where 'OP1' and 'OP2' are EQ, GE, GT, LE, LT, or NE; 'nnn1' and 'nnn2' are blocks unused values; and BOOL is AND or OR.
CATNAME(catalog name)	Where 'catalog name' is the name of the catalog to be searched for the dsns.
CATVOL(DDDDDD)	Where 'DDDDDD' is the volume serial. This is the variable used when you are generating a dataset list from the catalog and you want to limit the data sets generated to those on a particular volume.
CHGIND (OP DD1 DD2 DD3 DD4)	Where 'OP' is EQ or NE, and 'DD1' thru 'DD4' are either YES or NO for change indicator bit setting for the DSN.
COMPfmt (OP DD1 DD2 DD3 DD4)	Where 'OP' is EQ or NE, and 'DD1' thru 'DD4' are either YES or NO for compressed format for the dataset.

Parameters	Description
CREATEDT (OP1 dat1 BOOL OP2 dat2)	Where 'OP1' and 'OP2' are EQ, GE, GT, LE, LT, or NE; 'dat1' and 'dat2' are dates in the yyyy/mm/dd format; BOOL is AND or OR.
DATACLS(OP DDD1 DDD2 DDD3 DDD4)	Where 'OP' is EQ, NE, GT, LE, and so on, and 'DDD1' through 'DDD4' are the data classes.
DEVTYPE(OP DDD1 DDD2 ... DDD8)	Where 'OP' is EQ, NE, GT, LE, and so on, and 'DDD1' through 'DDD8' are the device types.
DSN(list of dsns)	Where "list of dsns" is the format used by ISMF for generating lists of data sets and volumes. The default is '*.*'.
DSORG(OP DD1 DD2 ... DD8)	Where 'OP' is EQ, NE, GT, LE, and so on, and 'DD1' through 'DD8' are the data set organizations.
ENTRYTYP(OP DDD1 DDD2 ... DDD12)	Where 'OP' is EQ, NE, GT, LE, and so on, and 'DDD1' through 'DDD12' are the entry types, (for example, DEFERRED, AIX, CLUSTER, and GDG).
EXPIREDT (OP1 dat1 BOOL OP2 dat2)	Where 'OP1' and 'OP2' are EQ, GE, GT, LE, LT, or NE; 'dat1' and 'dat2' are dates in the yyyy/mm/dd format; BOOL is AND or OR.
HSMDATA(Y N)	This specifies whether or not the user wants to use the ACQUIRE DATA FROM DFHSM option. The default is 'N'.
LASTBKUP (OP1 dat1 BOOL OP2 dat2)	Where 'OP1' and 'OP2' are EQ, GE, GT, LE, LT, or NE; 'dat1' and 'dat2' are dates in the yyyy/mm/dd format; BOOL is AND or OR.
LASTREF (OP1 dat1 BOOL OP2 dat2)	Where 'OP1' and 'OP2' are EQ, GE, GT, LE, LT, or NE; 'dat1' and 'dat2' are dates in the yyyy/mm/dd format; BOOL is AND or OR.
LRECL(OP1 nnn1 BOOL OP2 nnn2)	Where 'OP' is EQ, NE, GT, LE, and 'nnn1' is the lrecl specification; BOOL is AND or OR; OP2 has the same values as OP1, and 'nnn2' has the same values as nnn1.
MGMTCLS(OP nnnnn1 ... nnnnn4)	Where 'OP' is EQ, NE, and so on, and 'nnnnn1' through 'nnnnn4' are the management classes.
MULTVOL (OP DD1 DD2 DD3 DD4)	Where 'OP' is EQ or NE, and 'DD1' thru 'DD4' are either YES or NO if dataset is multivolume.
NOTUSED%(OP1 nn1 BOOL OP2 nnn2)	Where 'OP' is EQ, NE, GT, LE, and 'nn1' is the % of space not used; BOOL is AND or OR; OP2 has the same values as OP1; and 'nnn2' has the same values as nnn1.
NUMEXT(OP1 nnn1 BOOL OP2 nnn2)	Where 'OP1' is EQ, NE, GT, LE, and 'nnn1' is the extent specification; BOOL is AND or OR; OP2 has the same values as OP1; and 'nnn2' has the same values as nnn1.
NUMSTRIPE (OP1 nnn1 BOOL OP2 nnn2)	Where 'OP1' and 'OP2' are EQ, GE, GT, LE, LT, or NE; 'nnn1' and 'nnn2' are stripe number values; and BOOL is AND or OR.
OPTIMAL (OP1 nnn1 BOOL OP2 nnn2)	Where 'OP1' and 'OP2' are EQ, GE, GT, LE, LT, or NE; 'nnn1' and 'nnn2' are optimal blksize values; and BOOL is AND or OR.
OWNER (OP DD1 DD2 DD3 DD4)	Where 'OP' is EQ or NE, and 'DD1' thru 'DD4' are owners of the the datasets.

Parameters	Description
QSAVE(nnnnnnnn)	Where 'nnnnnn' is the query name to be created with all saved variables.
QUERY(nnnnnnnn)	Where 'nnnnnn' is the query name to be used for all the variables.
REBLOCK(OP DDD1 DDD2 ... DDD3)	Where 'OP' is EQ and 'DDD1' through 'DDD3' are either YES or NO.
RECFMT(OP DDD1 DDD2 .. DDD8)	Where 'OP' is EQ, NE, GT, LE, and so on, and 'DDD1' through 'DDD8' are the record formats. List built from the specified criteria.
SECALLOC (OP1 nnn1 BOOL OP2 nnn2)	Where 'OP1' and 'OP2' are EQ, GE, GT, LE, LT, or NE; 'nnn1' and 'nnn2' are secondary allocation values; and BOOL is AND or OR.
SOURCEGL(1 2)	Specifies whether the generated list comes from a saved list or is a new generated list. The default is '2'. When generating a list from the VTOC (SOURCENL is 1), you must specify VTOCVSER. The following parameters will be ignored: CATNAME, CATVOL, VTOCDATA, and HSMDATA. When generating a list from the catalog (SOURCENL is 2), use CATVOL to for the volume serial. Specify VTOCDATA and HSMDATA if needed. VTOCVSER is ignored.
SOURCENL(1 2)	Specifies whether the generated list comes from 1 - VTOC , or 2 - Catalog. The default is '2'.
STORCLS(OP nnnnnn1 .. nnnnnn4)	Where 'OP' is EQ, NE, and so on, and 'nnnnnn1' through 'nnnnnn4' are the storage classes.
USEDSPC(OP1 nnn1 BOOL OP2 nnn2)	Where 'OP' is EQ, NE, GT, LE, and 'nnn1' is the amount of used space in KB; BOOL is AND or OR; OP2 has the same values as OP1, and 'nnn2' has the same values as nnn1.
USERDATAREDUCT% (OP1 nnn1 BOOL OP2 nnn2)	Where 'OP1' and 'OP2' are EQ, GE, GT, LE, LT, or NE; 'nnn1' and 'nnn2' are% of user data reduction; and BOOL is AND or OR.
VTOCDATA(Y N)	Specifies whether the user desires the ACQUIRE DATA FROM VOLUME OPTION. The default is 'N'.
VTOCVSER(VVVVVV)	Specifies the volsers whose VTOCs are to be searched. The user may specify from 1–6 alphanumeric characters and an asterisk for filtering.

Sample JCL for Generating a Data Set List

```
//*****  
//*  
//* SAMPLE JCL TO CREATE ISMF DATA SET LIST IN BATCH AND SAVE IT *  
//*  
//* INSTRUCTIONS BEFORE SUBMITTING: *  
//*  
//* CHANGE JOBCARD *  
//* CHANGE PREFIX *  
//* CHANGE PARAMETERS *  
//*  
//* PARAMETERS: *  
//*  
//* PARAMETER FOLLOWING SAVE - NAME OF THE SAVED LIST (OUTPUT) *  
//* ALLOCSP - 0 to 9999999 (in Kilo Bytes) *  
//* BLKSIZE - 0 to 99999 (in Bytes) *  
//* BLKUNUSED - 0 to 9999999 (in Kilo Bytes) *  
//* CATNAME - ICF CATALOG name *  
//* CATVOL - VOLUME used in catalog list *  
//* CHGIND - YES/NO *  
//* COMPFMT - YES/NO *  
//* CREATEDT - YYYY/MM/DD (1900/01/01 to 2155/12/31) *  
//*          YY/MM/DD ( (19)00/01/01 to (19)99/12/31 ) *  
//* DATACLS - DATA CLASS name *  
//* DEVTYPE - 3380/3390/9345 for DASD types *  
//*          - 3480/3480X/3490/3590-1 for TAPE types *  
//* DSN - DATA SET NAME *  
//*          ('**' requires catalog name) *  
//* DSORG - DA/DAU/IS/ISU/PO/POU/PS/PSU/UN/V *  
//* ENTRYTYP - AIX/ALIAS/CATALOG/DATA/INDEX/NONVSAM/PATH/ *  
//*            DEFERRED/GDG/GDS/ROLLOFF *  
//* EXPIREDT - YYYY/MM/DD (1900/01/01 to 2155/12/31) or *  
//*          - YY/MM/DD ( (19)00/01/01 TO (19)99/12/31 ) or *  
//*          - NEVER or 1999/00/00 *  
//* HSMDATA - Y / N (Catalog List) *  
//* LASTBKUP - YYYY/MM/DD (1900/01/01 to 2155/12/31) *  
//*          YY/MM/DD ( (19)00/01/01 to (19)99/12/31 ) *  
//* LASTREF - YYYY/MM/DD (1900/01/01 to 2155/12/31) *  
//*          YY/MM/DD ( (19)00/01/01 to (19)99/12/31 ) *  
//* LRECL - 0 to 99999 (in Bytes) *  
//* MGMTCLS - MANAGEMENT CLASS name *  
//* MULTVOL - YES/NO *  
//* NOTUSED% - 0 to 100 *  
//* NUMEXT - 0 to 999 *  
//* NUMSTRIPE - 1 to 16 *  
//* OPTIMAL - 0 to 99999 (in Bytes) *  
//* OWNER - OWNER of the datasets *  
//* QSAVE - QUERY Name to be saved *  
//* QUERY - QUERY Name to be used *  
//* REBLOCK - YES/NO *  
//* RECFMT - A/B/D/F/M/FS/V *  
//* SECALLOC - 0 to 99999999 (in Kilo Bytes) *  
//* SOURCEGL - 1 / 2 (2-new list) *
```

Figure 17 (Part 1 of 2). Sample JCL for ACBJBAI2.

```

/** SOURCENL - 1 / 2 (1-VTOC 2-catalog) *
/** STORCLS - STORAGE CLASS name *
/** USEDSPC - 0 to 9999999 (in Kilo Bytes) *
/** USERDATAREDUCT% - 0 to 99 *
/** VTOCDATA - Y / N (Catalog List) *
/** VTOCVSER - VTOC VOLUME SERIAL NUMBERS (VTOC list) *
/** *
/*******
//SAVELIST EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAI2 SAVE DSNLIST +
VTOCDATA(Y) HSMDATA(Y) +
ALLOCS(500) DSORG(NE PS) +
DSN('IBMUSER.**')) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISP(9999999)
/*
//*****

```

Figure 17 (Part 2 of 2). Sample JCL for ACBJBAI2.

Generate a DASD or Optical Device Volume List: ACBQBAI4

ACBQBAI4 is called by the following SYS1.SACBCNTL members to generate the list of DASD and mountable optical device volumes in batch:

ACBJBAI4 Generate a DASD volume list, save it in a table, and save the query
 ACBJBAI5 Generate a DASD volume list and save it in a table
 ACBJBAI8 Generate a DASD volume list, save it in a table, and generate a report from it

ACBQBAI6 (see “Generate a Tape Volume List: ACBQBAI6” on page 42) generates the mountable tape volume list.

See Figure 18 on page 40 for the sample JCL and parameters.

Use the following parameters with ACBQBAI4. At least one OP and one value should be included when specifying a parameter.

Parameters	Description
ALLOCSP(OP1 nnn1 BOOL OP2 nnn2)	Specifies the amount of allocated space. Accepted values for nnn1 are from 0 to 9999999 (in KB); 'OP' is EQ, NE, GT, LE, and so on; BOOL is AND or OR; OP2 has the same values as OP1, and 'nnn2' has the same values as nnn1.
CDSNAME(cds.name)	The SCDS that extracts the volume information from for the volume list.
CFWSTAT(OP1 nnn1 nnn2 nnn3 nnn4)	Limits the volumes included in the list to those with specified cached-write status. Accepted values for nnn# are NONE, ACTIVE, INACTIVE, PENDING, or PINNED; 'OP' is EQ or NE.
DEVTYPE(XXXXXXXX)	Device type to use for the volume list. Note: If the user specifies the DEVTYPE parameter, NaviQuest attempts to determine the device type of the volume (for example, 3380-K). Currently NaviQuest only determines the correct device type if the volume is an entire MVS volume, that is, the using volume is not a VM minidisk. If the type cannot be determined, then the user receives a value of xxxx-?, where xxxx is the generic device type.
DFWSTAT(OP1 nnn1 nnn2 nnn3 nnn4)	Limits the volumes included in the list to those with specified DASD fast-write status. Accepted values for nnn# are NONE, ACTIVE, INACTIVE, PENDING, or PINNED; 'OP' is EQ or NE.
DUPLXSTAT(OP1 nnn1 nnn2 nnn3 nnn4)	Limits the volumes included in the list to those with specified duplex status. Accepted values for nnn# are NONE, SIMPLEX, PRIMARY, SECONDARY, PRI-PEN, SEC-PEN, PRI-SUS, and SEC-SUS; 'OP' is EQ or NE.
FRAG(OP1 nnn1 BOOL OP2 nnn2)	Specifies the fragmentation index. Accepted values for nnn1 are from 0 to 999; 'OP' is EQ, NE, GT, LE, and so on; BOOL is AND and OR; OP2 has the same values as OP1, and 'nnn2' has the same values as nnn1.

Parameters	Description
FREEDSCB(OP1 nnn1 BOOL OP2 nnn2)	Specifies the number of free DSCBs. Accepted values for nnn1 are from 0 to 99999;. 'OP' is EQ, NE, GT, LE, and so on; BOOL is AND or OR; OP2 has the same values as OP1, and 'nnn2' has the same values as nnn1.
FREESPC(OP1 nnn1 BOOL OP2 nnn2)	Specifies the amount of free space. Accepted values for nnn1 are from 0 to 9999999 (in KB); 'OP' is EQ, NE, GT, LE, and so on; BOOL is AND and OR; OP2 has the same values as OP1, and 'nnn2' has the same values as nnn1.
FREESPC%(OP1 nnn1 BOOL OP2 nnn2)	Limits the volumes included in the list to those with specified proportions of free space. Accepted values for nnn1 are from 0 to 100, with no % sign; 'OP' is EQ, NE, GT, LE, and so on; BOOL is AND or OR; OP2 has the same values as OP1, and 'nnn2' has the same values as nnn1.
FREEVIR(OP1 nnn1 BOOL OP2 nnn2)	Specifies the amount of free VIRs. Accepted values for nnn1 are from 0 to 99999; 'OP' is EQ, NE, GT, LE, and so on; BOOL is AND or OR; OP2 has the same values as OP1, and 'nnn2' has the same values as nnn1.
FREEXT(OP1 nnn1 BOOL OP2 nnn2)	Specifies the number of free extents. Accepted values for nnn1 are from 0 to 99999; 'OP' is EQ, NE, GT, LE, and so on; BOOL is AND or OR; OP2 has the same values as OP1, and 'nnn2' has the same values as nnn1.
FROMDEV(nnn)	First device number in the range to be listed in the volume list. Used with the LASTDEV parameter.
INDEX(OP1 nnn1 nnn2 nnn3 nnn4)	Limits the volumes included in the list to those with acceptable VTOC status. Accepted values for nnn# are ENABLED, DISABLED, or NONE; 'OP' is EQ or NE.
LASTDEV(nnn)	Last device number in the range to be listed in the volume list. Used with the FROMDEV parameter.
LRGEXT(OP1 nnn1 BOOL OP2 nnn2)	Limits the volumes in the list to those with largest extents of a specified size, in KB. Accepted values for nnn1 are from 1 to 9999999; 'OP' is EQ, NE, GT, LE, and so on; BOOL is AND, OR, OP2 has the same values as OP1, and 'nnn2' has the same values as nnn1.
OTHERDEV(OP1 nnn1 BOOL OP2 nnn2)	Specifies the address of the duplex copy secondary device. Accepted values for nnn1 are from 0 to FFF; 'OP' is EQ, NE, GT, LE, and so on; BOOL is AND or OR; OP2 has the same values as OP1, and 'nnn2' has the same values as nnn1.
PHYDATA(Y N)	Specifies whether or not the user desires the ACQUIRE PHYSICAL DATA OPTION. The default is 'N'.
PHYSTAT(OP1 nnn1 nnn2 nnn3 nnn4)	Limits the volumes included in the list to those with a specific SMS status. Accepted values for nnn# are INITIAL, CONVERT, NONSMS, or UNKNOWN; 'OP' is EQ or NE.
QSAVE (nnnnnnnn)	Where 'nnnnnnnn' is the query name to be created with all saved variables.
QUERY (nnnnnnnn)	Where 'nnnnnnnn' is the query name to be used for all variables.

Parameters	Description
RDCACHE(OP1 nnn1 nnn2 nnn3 nnn4)	Limits the volumes included in the list to those with a specified caching status. Accepted values for nnn# are NONE, ACTIVE, INACTIVE, or PENDING; 'OP' is EQ or NE.
SHRDASD(OP1 nnn1 nnn2 nnn3 nnn4)	Limits the volumes included in the list to those with DASD that either are or are not shareable between multiple CPUs. Accepted values for nnn# are YES or NO; 'OP' is EQ or NE.
SOURCEGL(1 2)	Specifies whether the generated list come from (1) a saved list or (2) is a new list built from the the specified criteria. The default is '2'.
SOURCENL(1 2)	Specifies whether the generated list comes from PHYSICAL(1) or DFSMS(2). The default is '1'. When generating an SMS volume list (SOURCENL is '2'), STORGRP defaults to '*' and CDSNAME defaults to 'ACTIVE' if no values are specified.
SPCDATA(Y N)	Specifies whether or not the user desires the ACQUIRE SPACE DATA OPTION. The default is 'N'.
STORGRP(XXXXXXXX)	This is the STORGRP to extract the volume information from.
SUBSYSID(OP1 nnn1 BOOL OP2 nnn2)	Limits the volumes included in the list to those with specified subsystems. You may specify a single subsystem number or a range of subsystem numbers. Accepted values for nnn1 are from 0001 to 00FF; 'OP' is EQ, NE, GT, LE, and so on; BOOL is AND or OR; OP2 has the same values as OP1, and 'nnn2' has the same values as nnn1.
USEATTR(OP1 nnn1 nnn2 nnn3 nnn4)	Limits the volumes included in the list to those with allowable DASD attributes. Accepted values for nnn# are PUB, PRIV, or STOR; 'OP' is EQ or NE.
VOL (list of VOLS)	Where "list of VOLS" is the format used by ISMF for generating lists of data sets and volumes. The default is '*'.
VOLSTYPE(1 2 3)	Specifies whether the generated list will come from ONLINE(1), NOT ONLINE(2), or EITHER (3). The default is '1'.

Sample JCL for Generating a Volume List

```
//*****
//*
//* SAMPLE JCL TO GENERATE ISMF DASD VOLUME LIST IN BATCH, SAVE IT, *
//* AND SAVE THE QUERY ALSO *
//* *
//* INSTRUCTIONS BEFORE SUBMITTING: *
//* *
//* CHANGE JOBCARD *
//* CHANGE PREFIX *
//* CHANGE PARAMETERS *
```

Figure 18 (Part 1 of 2). Sample JCL for ACBJBA14.

```

//*
//* PARAMETERS:
//*
36A/C @WA29014
//* PARAMETER FOLLOWING SAVE - NAME OF SAVED LIST (OUTPUT)
//* ALLOCSP - 0 to 9999999 (in Kilo Bytes)
//* CDSNAME - Control Dataset Name
//* CFWSTAT - ACTIVE/INACTIVE/NONE/PENDING/PINNED
//* DEVTYPE - 3380/3390/9345
//*
- Installation defined esoteric names(like SYSDA ..)
//* DFWSTAT - ACTIVE/INACTIVE/NONE/PENDING/PINNED
//* DUPLXSTAT - NONE / PPRI-FAI / PPRI-PEN / PPRI-SUS / PPRIMARY/
PRI-PEN / PRI-SUS / PRIMARY / PSEC-FAI / PSEC-PEN/
//* PSEC-SUS /PSECNDRY/ SEC-PEN / SEC-SUS / SECONDRY/
SIMPLEX / SPAR-BRK / SPAR-PEN / SPARE
//* FRAG - 1 to 999
//* FREEDSCB - 0 to 99999
//* FREESPC - 0 to 9999999 (in Kilo Bytes)
//* FREESPC% - 0 to 100 (with no % sign)
//* FREEVIR - 0 to 99999
//* FREEEXT - 0 to 99999
//* FROMDEV - 1 to 4 Hexadecimal digits
//* INDEX - DISABLED/ENABLED/NONE
//* LASTDEV - 1 to 4 Hexadecimal digits (should be >= FROMDEV)
//* LRGEXT - 1 to 9999999 (in Kilo Bytes)
//* OTHERDEV - 1 to 4 Hex Decimal digits
//* PHYDATA - Y / N
//* PHYSTAT - CONVERT/INITIAL/NONSMS/UNKNOWN
//* QSAVE - Query Name to be saved
//* QUERY - Query Name to be used
//* RDCACHE - ACTIVE/INACTIVE/NONE/PENDING
//* SHRDASD - YES / NO
//* SOURCEGL - 1 / 2 (2-New List)
//* SOURCENL - 1 / 2 (1-Physical, 2-SMS)
//* SPCDATA - Y / N
//* STORGRP - Stotage Group name (SMS only)
//* SUBSYSID - 0001 to FFFF (in Hexadecimal)
//* USEATTR - PRIV/PUB/STOR
//* VOL - Volume serial
//* VOLSTYPE - 1 / 2 / 3 (1-On Line 2-Not On Line 3-Either)
//*
//*****
//DASDLST EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER) MSGID
ISPSTART CMD(ACBQBIA4 +
SAVE CHKFRAG QSAVE(CHKFRAG) +
SPCDATA(Y) PHYDATA(Y) +
VOL(TSC*) FRAG(GT 450)) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*****

```

Figure 18 (Part 2 of 2). Sample JCL for ACBJBAI4.

Generate a Tape Volume List: ACBQBAI6

ACBQBAI6 is called by the following SYS1.SACBCNTL members to list the mountable tape volumes in batch:

ACBJBAID Creates ISMF table of scratch tapes in a library, and then prints the table.
 ACBJBAIA Creates ISMF table of all tapes in a library.

See Figure 19 for the sample JCL and parameters.

Use the following parameters with ACBQBAID:

Parameters	Description
LIBNAME (nnn)	Specifies the library name to query against. The default is '*'. *
SOURCEGL(1 2)	Specifies whether the generated list comes from (1) a saved list or (2) a new list built from the specified criteria. The default is '2'. *
STORGRP(XXXXXXXX)	Specifies the storage group to extract the volume information from. The default is '*'. *
VOL(list of VOLS)	Where "list of VOLS" is in the format for ISMF. The default is '*'. *

Sample JCL for Generating a Tape List:

```

//*****
//*
//* SAMPLE JCL TO GENERATE AN ISMF MOUNTABLE TAPE VOLUME LIST,
//* SAVE IT, AND GENERATE A TAPE VOLUME REPORT FROM IT
//*
//* INSTRUCTIONS BEFORE SUBMITTING:
//*
//* CHANGE JOBCARD
//* CHANGE PROFILE PREFIX COMMAND
//*
//*****
//*****
//*
//* STEP TO GENERATE TAPE VOLUME LIST AND SAVE IT
//*
//* VOL - TAPE VOLUMES TO BE SELECTED
//* LIBNAME - LIBRARY NAMES TO BE SELECTED
//* STORGRP - STORAGE GROUPS TO BE SELECTED
//* PARAMETER FOLLOWING SAVE - NAME FOR THE SAVED LIST (OUTPUT)
//*
//*****

```

Figure 19 (Part 1 of 2). Sample JCL for ACBJBAIA.


```

//GENLIST EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
DELETE TAPEVOL.REPORT
ISPSTART CMD(ACBQBAI6 SAVE TAPELIST +
VOL(*) LIBNAME(L*) STORGRP(*)) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*****
//*
//* STEP TO GENERATE THE TAPE VOLUME REPORT FROM THE SAVED LIST *
//*
//* PARAMETER FOR ACBQBAR4 - SAVED LIST NAME (FROM PREVIOUS STEP) *
//* ISPFIL - TAPE VOLUME REPORT (OUTPUT) *
//* SYSIN - KEY WORDS SPECIFYING COLUMNS TO BE INCLUDED IN THE *
//* GENERATED REPORT *
//*****
//GENREP EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAR4 TAPELIST) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//ISPFIL DD DSN=IBMUSER.TAPEVOL.REPORT,DISP=(NEW,CATLG),
// BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA
//SYSIN DD *
VOLSER
USEATTR
VOLERROR
CKPT
LIBNAME
STORGRP
MEDIA
RECTECH
COMPTYPE
LASTWRITE
LASTMOUNT
LASTEJECT
EXPDTVOL
CRTDTVOL
VOLLOC
TITLE=STATUS OF TAPES AS ON 06/01/96
/*
//*****
//*
//* COPY THE REPORT *
//*
//* SYSUT1 - INPUT (FROM PREVIOUS STEP) *
//* SYSUT2 - OUTPUT *
//*****
//TAPGEN EXEC PGM=IEBGENER
//SYSUT1 DD DSN=IBMUSER.TAPEVOL.REPORT,DISP=SHR
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//*****

```

Figure 19 (Part 2 of 2). Sample JCL for ACBJBAIA.

Define/Alter/Display Management Class: ACBQBAJ1

SYS1.SACBCNTL member ACBJBAJ1 calls ACBQBAJ1 to define, alter, or display data set management class in batch.

See Figure 20 on page 45 for the sample JCL and parameters.

Use the following parameters with ACBQBAJ1:

Parameters	Description
ACPYTECH	A backup copy technique
AUTOBKUP	Automatic backup
BACKUPFR	Backup Frequency
BKUPTECH	Backup copy technique
CMDBKUP	Administration or user command backup
CPYSERLN	Copy serialization
CMDORAUT	Command or automatic migrate
DESCR	Task description
EXPDTDY	Expire after date or days
EXPNOUSE	Expire after days of non-usage
GDGROLL	Rolled-off GDG Action
LVINOUSE	Level days of non-usage
MGMTCLAS	Management class name
NUMBKDSD	Number of backup versions (data set deleted)
NUMBKDSE	Number of backup versions (data set exists)
PARTREL	Partial release
PMTHODAY	Periodic monthly on day
PQUAIMTH	Periodic quarterly in month
PQUAODAY	Periodic quarterly on day
PRIGDGEL	Number of GDG elements in primary
PRINOUSE	Primary days nonusage
PYRLIMTH	Periodic yearly in month
PYRLODAY	Periodic yearly on day
RETDYDSD	Retain days only backup version (data set deleted)
RETDYEXT	Retain days extra backup version
RETNLIM	Retention limit
RTEVUNIT	Retain extra versions unit
RTNEVERS	Retain extra versions
RTNOVERS	Retain only version
RTOVUNIT	Retain only version unit
SCDS	CDS name
TMSCDYS	Time since creation days

Parameters	Description
TMSCMTH	Time since creation months
TMSCYRS	Time since creation years
TMSLUDYS	Time since last use days
TMSLUMTH	Time since last use months
TMSLUYRS	Time since last use years
VERSIONS	Number of versions

Sample JCL for Define/Alter/Display Management Class

```

//*****
//*
//* SAMPLE JCL TO DEFINE/ALTER/DISPLAY MANAGEMENT CLASSES IN BATCH
//*
//* INSTRUCTIONS BEFORE SUBMITTING:
//*
//*   CHANGE JOBCARD
//*   CHANGE PREFIX
//*   CHANGE PARAMETERS
//*
//* PARAMETER FOLLOWING ACBQBAJ1 - DEFINE or ALTER or DISPLAY
//* SCDS - SCDS in which MANAGEMENT CLASS is to be DEF/ALT/DISP
//* MGMTCLAS - MANAGEMENT CLASS to be DEFINED/ALTERED/DISPLAYED
//* (Define or Alter)
//*
//* DESCR      : Type in remarks about the MGMTCLAS which is being
//*              defined/alterd, not exceeding 120 chars.
//*
//* EXPNOUSE   : The datasets will expire if they are not used for
//*              the number of days specified here.
//*
//*              Possible values 1 - 9999, NOLIMIT. If NOLIMIT is
//*              specified the DS would not expire.
//*              Valid only if retention period or expiration date
//*              is not specified by the end user or is not derived
//*              from the data class.
//*
//* EXPDTDY    : Datasets expires after DATE/DAYS entered here.
//*              Possible values 0 - 9999,YYYY/MM/DD or NOLIMIT.
//*
//* RETNLIM    : Possible values 0 - 9999, NOLIMIT.
//*              Use this field to control what a user or Data class
//*              can specify for retention period or expiration date
//*              during allocation. The affect of the values entered
//*              in this field are explained below.
//*
//*              0          -> Do not use the RETPD and EXPDT that
//*              the user or Dataclass specified.
//*              1 - 9999 -> Use this value only if the RETPD or
//*              EXPDT is more than this limit.

```

Figure 20 (Part 1 of 7). Sample JCL for ACBJBAJ1.

```

/**          NOLIMIT  -> Do not set a limit to RETPD or EXPDT. *
/**
/** PARTREL  : Possible values Y, C, YI, CI or N .           *
/**          Use this field (PARTIAL RELEASE) to specify whether *
/**          allocated but unused space can be released for DSs *
/**          in this MGMTCLS. This one applies only to VSAM DSs *
/**          in extended format or NON-VSAM datasets. The values *
/**          entered would have following results.             *
/**
/**          Y  -> Release unused space automatically during the *
/**          Space Management cycle.                            *
/**
/**          C  -> Unused space can be released automatically *
/**          only if a secondary allocation exists for the *
/**          dataset.                                           *
/**
/**          YI -> Release unused space when a dataset is closed *
/**          or during the Space Management cycle, whichever *
/**          comes first.                                       *
/**
/**          CI -> Unused space for data sets with secondary *
/**          allocation is released either when a data set *
/**          is closed or during the Space Management *
/**          cycle, whichever comes first.                       *
/**
/**          N  -> Do not release unused space.                 *
/**
/** PRINOUSE : Use this field to specify when to migrate the DSs *
/**          in this class. The possible values are             *
/**
/**          0      -> To Migrate data sets as soon as the *
/**          space management function of DFSMSHsm *
/**          is run and data integrity age is met.             *
/**
/**          1 to 9999 -> Migrate data sets out of primary *
/**          storage if they have been unused for *
/**          this number of days or longer.                     *
/**
/**          BLANK  ->                                         *
/**
/** LVINOUSE : Use this field to specify whether DSs can migrate *
/**          to LEVEL 1 storage and how long they can remain *
/**          there. The possible values are,                     *
/**
/**          0      -> No migration to Level 1. DSs migrate *
/**          directly from primary storage to LVL 2 *
/**
/**          1 to 9999 -> The total number of consecutive days *
/**          that datasets must remain unaccessed *
/**          before becoming eligible to migrate *
/**          from LVL 1 to LVL 2.                               *
/**
/**          NOLIMIT -> Datasets can not migrate to LEVEL 2 *
/**          automatically, and remain in LVL 1 for *
/**          an unlimited period.                                *

```

Figure 20 (Part 2 of 7). Sample JCL for ACBJBAJ1.

```

/**          BLANK    ->                                     *
/**
/** CMDORAUT : If migration is allowed, this field determines how *
/** the migration is initiated. Possible values are,          *
/**
/**          BOTH     -> DSs can migrate either automatically *
/**                   or by command.                          *
/**          COMMAND  -> Data sets can migrate by command only. *
/**
/**          NONE     -> Data sets cannot migrate.             *
/**
/** PRIGDGEL : Valid for Generation Data Group (GDG) DSs only. *
/** This field specifies how many of the newest                *
/** generations of a GDG are to have normal priority.        *
/** Possible values are 0 - 255 or blank. For Example         *
/** enter 100 if you want GDG generations older              *
/** than the most recent 100, to migrate before non          *
/** generation datasets.                                     *
/**
/** GDGROLL  : This field specifies whether the Generation DSs *
/** in this MGMTCLS will expire or migrate after they        *
/** have been removed from the GDG. The possible             *
/** values are, MIGRATE, EXPIRE or blank.                    *
/**
/** BACKUPFR : This field specifies the backup frequency. The *
/** possible values are,                                     *
/**
/**          0         -> Backup each dataset only when the volume *
/**                   it resides on is backed up.              *
/**          1 - 9999 -> If dataset is changed in the interval *
/**                   between backups, extend the interval    *
/**                   for atleast this many number of days.   *
/**          BLANK     ->                                     *
/**
/** NUMBKDSE : Maximum number of Backups that can be kept *
/** concurrently. Possible values are, 1 - 100, BLANK.        *
/**
/** NUMBKDSD : Specifies the maximum no of Backups to keep after *
/** the dataset is deleted. Possible values are                *
/**
/**          0         -> All backups that were created are erased *
/**                   after the dataset is deleted.            *
/**          1 - 100  -> The maximum no. of backups to keep after *
/**                   a dataset has been deleted.              *
/**          BLANK     ->                                     *
/**
/** RETDYDSD : Specifies how long a most recent backup version of *
/** a deleted dataset will be kept. Possible values are        *
/**
/**          1 - 9999 -> After a dataset is deleted keep its most *
/**                   recent backup version for these many    *
/**                   days.                                    *
/**
/**          NOLIMIT  -> The backup version will be kept for *
/**                   unlimited period.                        *

```

Figure 20 (Part 3 of 7). Sample JCL for ACBJBAJ1.

```

//*          BLANK  ->                                     *
//*
//* RETDYEXT : Specifies the retention period for a dataset that *
//* pre-date the most recent backup. Possible values *
//* are, *
//* *
//*          1 - 9999 -> Each backup version of a dataset other *
//* than the recent copy will be kept for *
//* these many days. *
//* *
//*          NOLIMIT -> All backup versions will be kept for *
//* unlimited period. *
//* *
//*          BLANK  ->                                     *
//*
//* CMDBKUP  : Specifies who will have authority to perform command*
//* backups. Possible values are, *
//* *
//*          ADMIN -> Only Storage Administrator , *
//* *
//*          BOTH  -> Both Storage Administrator and end users. *
//* *
//*          NONE  -> Neither end user nor Storage Administrator*
//* *
//* AUTOBKUP : Specifies whether the datasets in this MGMTCLS are *
//* eligible for automatic backup. Possible values are *
//* Y -> Yes , N -> No *
//* *
//* BKUPTECH : Specifies BACKUP COPY TECHNIQUE to be used. *
//* Possible values are, *
//* *
//*          R -> Concurrent copy technique must be used. *
//*          P -> Concurrent copy technique should be used. *
//*          S -> With out the concurrent copy technique. *
//* *
//* TMSCYRS  : No of years that must pass since the creation date *
//* before class transition occurs. Possible values are *
//* 0 - 9999, or BLANK. *
//* *
//* TMSCMTH  : No of months that must pass since the creation date *
//* before class transition occurs. Possible values are *
//* 0 - 9999, or BLANK. *
//* *
//* TMSCDYS  : No of days that must pass since the creation date *
//* before class transition occurs. Possible values are *
//* 0 - 9999, or BLANK. *
//* *
//* TMSLUYRS : No of years that must pass since the last reference *
//* date before class transition occurs. Possible values *
//* are 0 - 9999, or BLANK. *
//* *
//* TMSLUMTH : No of months that must pass since the last reference*
//* date before class transition occurs. Possible values *
//* are 0 - 9999, or BLANK. *
//*

```

Figure 20 (Part 4 of 7). Sample JCL for ACBJBAJ1.

```

/** TMSLUDYS : No of days that must pass since the last reference *
/**          date before class transition occurs.Possible values *
/**          are 0 - 9999, or BLANK. *
/** *
/** PMTHODAY : The day of the month that class transition occurs. *
/**          Possible values, 1 - 31, FIRST, LAST or BLANK *
/** *
/** PQUAODAY : The day of the each quarter the CT occurs. *
/**          Possible values, 1 - 92, FIRST, LAST or BLANK *
/** *
/** PQUAIMTH : Month of each quarter the CT occurs. *
/**          Possible values, 1 - 3, or BLANK *
/** *
/** PYRLODAY : The day of each year the CT occurs. *
/**          Possible values, 1 - 366,FIRST,LAST or BLANK *
/** *
/** PYRLIMTH : The month of each year the CT occurs. *
/**          Possible values, 1 - 12, or BLANK *
/** *
/** VERSIONS : Specify how many versions of an aggregate group *
/**          associated with the management class are to be *
/**          maintained. Possible values are 1 - 9999, NOLIMIT or *
/**          BLANK. If BLANK is specified no aggregate group BKP *
/**          is maintained. *
/** *
/** RTNOVERS : Specify how long the only version of an aggregate *
/**          group is kept. Possible values are 1 - 9999,NOLIMIT *
/**          or BLANK. *
/** *
/** RTOVUNIT : Specify the unit of measure for the length of time*
/**          specified in the above field. Possible values are *
/**          D -> Days, W -> Weeks, M -> Months, Y -> Years and *
/**          BLANK. *
/** *
/** RTNEVERS : Specify the time periods for which backup versions*
/**          of an aggregate group are to be kept.Possible values*
/**          are 1 - 9999,NOLIMIT and BLANK. *
/** *
/** RTEVUNIT : Specify the unit of measure for the length of time *
/**          specified in the above field. Possible values are *
/**          D -> Days, W -> Weeks, M -> Months, Y -> Years and *
/**          BLANK. *
/** *
/** CPYSERLN : Specifies whether you want processing of a backup *
/**          copy of an aggregate group to continue if a shared *
/**          enqueue cannot be obtained for the datasets being *
/**          backed up. Possible values are, *
/**          C -> Continue, F -> Fail or BLANK. *
/** *
/** ACPYTECH : Specifies ABACKUP COPY TECHNIQUE to be used. *
/**          Possible values are, *
/** *
/**          R -> Concurrent copy technique must be used. *
/**          P -> Concurrent copy technique should be used. *
/**          S -> With out the concurrent copy technique. *

```

Figure 20 (Part 5 of 7). Sample JCL for ACBJBAJ1.

```

//*
//*
//*****
//*****
//*
//*   STEP1 - SET UP PARAMETERS
//*
//*****
//STEP1 EXEC ACBJBAOB,TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD SYSOUT=*
//TEMPFILE DD DSN=&&TEMPFILE,DISP=(NEW,PASS),
// SPACE=(TRK,(1,1)),LRECL=300,RECFM=F,BLKSIZE=300
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAJ1 DEFINE/ALTER +
SCDS(TEMP.SCD) +
MGMTCLAS() +
DESCR() +
EXPNOUSE() +
EXPDTDY() +
RETNLIM() +
PARTREL() +
PRINOUSE() +
LVINOUSE() +
CMDORAUT() +
PRIGDGEL() +
GDGROLL() +
BACKUPFR() +
NUMBKDSE() +
NUMBKDSD() +
RETDYDSD() +
RETDYEXT() +
CMDBKUP() +
AUTOBKUP() +
BKUPTECH() +
TMSCYRS() +
TMSCMTH() +
TMSCDYS() +
TMSLUYRS() +
TMSLUMTH() +
TMSLUDYS() +
PMTHODAY() +
PQAODAY() +
PQUAIMTH() +
PYRLODAY() +
PYRLIMTH() +
VERSIONS() +
RTNOVERS() +
RTOVUNIT() +
RTNEVERS() +
RTEVUNIT() +
CPYSERLN() +
ACPYTECH() +

```

Figure 20 (Part 6 of 7). Sample JCL for ACBJBAJ1.


```

)
/*
/*****
/*
//*   STEP2 - EXECUTE THE DEFINE
//*
//*****
//STEP2  EXEC ACBJBAOB
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  DSN=&&TEMPFILE,DISP=(OLD,DELETE,DELETE)
/*****
//*
//*   STEP3 - SET UP PARAMETERS
//*
//*****
//STEP3  EXEC ACBJBAOB,TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD  SYSOUT=*
//TEMPFILE DD  DSN=&&TEMPFILE,DISP=(NEW,PASS),
//  SPACE=(TRK,(1,1)),LRECL=300,RECFM=F,BLKSIZE=300
//SYSTSIN  DD  *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAJ1 DISPLAY +
SCDS(TEMP.SCDS) +
MGMTCLAS() +
)
/*

```

Figure 20 (Part 7 of 7). Sample JCL for ACBJBAJ1.

Define/Alter/Display Data Class: ACBQBAD1

SYS1.SACBCNTL member ACBJBAD1 calls ACBQBAD1 to define, alter, or display the data class in batch.

See Figure 21 on page 53 for the sample JCL and parameters.

Use the following parameters with ACBQBAD1:

Parameters	Description
ADDDVOLAM	Additional volume amount
BWO	Backup while open
CISZDATA	Csize data
COMPTN	Compaction
DCNAME	Data class name
DESCR	Task description
DSNM TYP	Data set name type
EXADDRS	Extended addressability
FRSPCCA	% Freespace CA
FRSPCCI	% Freespace CI
IFEXT	If extended
IMBED	Imbed
INILOAD	Initial load
KEYLEN	Key length
KEYOFF	Key off
LOG	Log
LOGSTID	Log stream id
LRECL	Record length
MDTYPE	Media type
RECACCB	Record access bias
RECFM	Record format
RECO RG	Record organization
RECTECH	Record technology
REDSPCUT	Reduce space up to (%)
REPLCAT	Replicate
REUSE	Reuse
REXP PDT	Retpd or expdt
SCDS	CDS name
SHRXREG	Shareoptions Xregion
SHRXSYS	Shareoptions Xsystem
SPANONSP	Spanned or nonspanned
SPCAVREC	Space average record

Parameters	Description
SPCAVVAL	Space average value
SPCCONRL	Space constraint relief
SPCDIR	Space directory
SPCPRM	Space primary
SPCSEC	Space secondary
VOLCNT	Volume count

Sample JCL for Define/Alter/Display Data Class

```

//*****
//*
//* SAMPLE JCL TO DEFINE/ALTER/DISPLAY DATA CLASSES IN BATCH
//*
//* INSTRUCTIONS BEFORE SUBMITTING:
//*
//* CHANGE JOBCARD
//* CHANGE PREFIX
//* CHANGE PARAMETERS
//*
//* PARAMETER FOLLOWING ACBQBAD1 - DEFINE OR ALTER OR DISPLAY
//*****
//* SCDS : Specify the name of the CDS that contains the
//* dataclass you want to Define/Alter/Display.
//*
//* Possible values : Valid CDS name .
//*
//* DCNAME : Name of the Dataclass.
//*
//* Possible values : 1 - 8 characters
//*
//* DESCR : Remarks about the DC being defined/changed.
//* 1-120 characters.
//*
//* REORG : Specify how the records in the Datasets will be
//* organized during allocation.
//*
//* Possible values :
//*
//* KS -> VSAM Key Sequenced
//* ES -> VSAM Entry Sequenced
//* RR -> VSAM Relative Record
//* LS -> VSAM Linear Space
//* BLANK -> PS or PDS
//*
//* RECFM : Specify the format of records for Non VSAM DSS
//* in this dataclass.
//*
```

Figure 21 (Part 1 of 8). Sample JCL for ACBQBAD1.

```

/**          Possible Values :          *
/**          *                          *
/**          U[A|M]  -> Undefined        *
/**          V[A|M]  -> Variable          *
/**          VS[A|M] -> Variable Spanned *
/**          VB[A|M] -> Variable Blocked  *
/**          VBS[A|M]-> Variable Blocked & Spanned *
/**          F[A|M]  -> Fixed              *
/**          FS[A|M] -> Fixed Standard    *
/**          FB[A|M] -> Fixed Blocked     *
/**          FBS[A|M]-> Fixed Blocked Standard *
/**          BLANK   -> Specify no record format *
/**          *                          *
/**  LRECL      : Specify the logical record length of records in *
/**              this dataclass.For variable length or undefined *
/**              records this is the maximum length of a record. *
/**          *                          *
/**          Possible Values :          *
/**          *                          *
/**          1 to 32760 or BLANK if RECORG is BLANK *
/**          *                          *
/**          1 to 32761 or BLANK if RECORG is ES, KS, or RR *
/**          *                          *
/**  KEYLEN     : Specify the length of the key field for records *
/**              in this dataclass. *
/**          *                          *
/**          Possible Values :          *
/**          *                          *
/**          0 to 255 or blank if RECORG is blank *
/**          *                          *
/**          1 to 255 or blank if RECORG is KS *
/**          *                          *
/**  KEYOFF     : Specify key offset for key sequenced datasets *
/**              in this DC. *
/**          *                          *
/**          Possible Values : 0 - 32760, BLANK *
/**          *                          *
/**  SPCAVREC   : Specify space units. *
/**          *                          *
/**          Possible Values :          *
/**          *                          *
/**          K      -> Kilo Bytes *
/**          M      -> Mega Bytes *
/**          U      -> Bytes *
/**          BLANK -> *
/**          *                          *
/**  SPCAVVAL   : Average length of each record in bytes. *
/**          *                          *
/**          Possible Values : 0 - 65535 or blank *
/**          *                          *
/**  SPCPRM     : Specify no. of records Primary storage will *
/**              contain. *
/**          *                          *
/**          Possible Values : 0 - 999999 or blank *
/**          *                          *

```

Figure 21 (Part 2 of 8). Sample JCL for ACBJBAD1.

```

/** SPCSEC : Specify no. of records Secondary storage will *
/** contain. *
/** *
/** Possible Values : 0 - 999999 or blank *
/** *
/** SPCDIR : Specify no. of directory blocks to be allocated *
/** for PDS. *
/** *
/** Possible Values : 0 - 999999 or blank *
/** *
/** REXPPDT : Specify the default retention period or *
/** expiration date of datasets in this DC. *
/** *
/** Possible Values : *
/** *
/** 0 to 9999 -> Datasets expire in no.of days *
/** *
/** yyyy/mm/dd -> Datasets expiry date *
/** yyyy => 1900 - 2155 *
/** yyyy/00/00 -> Special value *
/** yyyy => 1900 - 2155 *
/** BLANK -> No expiration date specified *
/** *
/** VOLCNT : Maximum no of volumes you expect to a DS in *
/** this DC. *
/** *
/** Possible Values : 1 - 59, BLANK *
/** *
/** ADDVOLAM : Specify the allocation amount when a VSAM DS *
/** in extended format begins allocation on *
/** subsequent volumes. *
/** *
/** Possible Values : *
/** *
/** P Use primary allocation amount *
/** S Use secondary allocation amount *
/** BLANK Use default value of primary *
/** *
/** IMBED : Specify the location of the sequence-set *
/** record for each control area. *
/** *
/** Possible Values : *
/** *
/** Y -> Imbed sequence-set record as many times *
/** as it will fit on the first track *
/** adjacent to the data control area. *
/** *
/** N -> Don't imbed sequence-set records *
/** *
/** BLANK -> Use VSAM defaults. *
/** *
/** REPLCAT : Specify whether key sequenced VSAM datasets in *
/** this DC will use index replication. *
/** *

```

Figure 21 (Part 3 of 8). Sample JCL for ACBJBAD1.

```

/*           Possible Values :                               *
/*
/*           Y   -> Write each index record on a track as  *
/*                many times as it will fit.                *
/*
/*           N   -> Write each index record on a track only *
/*                one time.                                  *
/*
/*           BLANK -> No REPLICATE value is specified.      *
/*
/*           CISZDATA : Specify the size of each Control Interval for DS *
/*                in this data class. Applies only to VSAM DSs *
/*                with RECOrg of KS,ES,LS or RR only.       *
/*
/*           Possible Values : 1 - 32768, BLANK             *
/*
/*           FRSPCCI : Percentage of free space you want to reserve in *
/*                the CI to avoid splits. Valid only for VSAM KSDS. *
/*
/*           Possible Values : 0 - 100                      *
/*
/*           FRSPCCA : Percentage of free space you want to reserve in *
/*                the CA to avoid splits. Valid only for VSAM KSDS. *
/*
/*           Possible Values : 0 - 100                      *
/*
/*           SHRREG : Specify how the data will be shared with in one *
/*                system. Applies to VSAM DSs with RECOrg of *
/*                KS,ES,LS and RR only.                    *
/*
/*           Possible Values :                               *
/*
/*           1 -> All users can read the DS when no one is *
/*                writing to it.                             *
/*
/*           2 -> All users can read the DS even if one *
/*                user is writing to it.                     *
/*
/*           3 -> All users can both read and write to the *
/*                DS. VSAM doesn't ensure the data integrity. *
/*
/*           4 -> All users can both read and write to the *
/*                DS. VSAM provides some assistance to ensure *
/*                data integrity.                            *
/*
/*           BLANK -> No share options specified.           *
/*
/*           SHRXSYS : Specify how the data will be shared among the *
/*                systems. Applies to VSAM DSs with RECOrg *
/*                KS,ES,LS and RR only.                    *
/*
/*           Possible values : (See above)                  *
/*
/*           COMPTN : Specify whether tape volumes or DASD datasets *
/*                associated with this DC are to be compacted or *
/*                compressed.                               *

```

Figure 21 (Part 4 of 8). Sample JCL for ACBJBAD1.

```

/**
/**          Possible values :
/**          Y      -> DSs are compressed and Tape volumes are
/**                compacted.
/**          N      -> DSs are not compressed and Tape volumes
/**                are not compacted unless requested by
/**                USER on JCL/DYNAMIC allocation.
/**          BKANK -> DSs are not compressed and Tape volumes
/**                are not compacted unless requested by
/**                USER on JCL/DYNAMIC allocation or by the
/**                installation through parmlib specification.*
/**
/** MDTYPE   : Specify mountable tape cartridge type used for
/**          DSs associated with this DC.
/**
/**          Possible values :
/**
/**          1      -> For MEDIA1 ( Cartridge System Tape)
/**          2      -> For MEDIA2 ( Enhanced Capacity CST)
/**          3      -> For MEDIA3 ( High Performance CT)
/**          4      -> For MEDIA4 ( Extended HPCT)
/**          BLANK -> Cartridge type not specified.
/**
/** RECTECH  : Specify the number of tracks on mountable tape
/**          cartridges used for DSs associated with this DC.
/**          Optional.
/**
/**          Possible values :
/**
/**          18     -> An 18-track cartridge is used.
/**          36     -> A 36-track cartridge is used.
/**          128    -> A 128-track cartridge is used.
/**          BLANK  -> The system default is used.
/**
/** DSNMTYP  : Specify the format of the datasets in this DC.
/**
/**          Possible values :
/**
/**          EXTENDED -> Extended format datasets
/**          HFS      -> Hierarchical File System Datasets
/**          LIBRARY  -> Datasets are allocated as PDSEs
/**          PDS      -> Datasets are allocated as PDSs
/**          BLANK    -> Not specified
/**
/** IFEXT    : Specify whether this DC requires the DSs to be
/**          allocated only in extended sequential format.
/**          Required only if DSNMTYP is specified as EXT.
/**
/**          Possible values :
/**
/**          P      -> Preferred
/**          R      -> Required
/**          BLANK  -> DSNMTYP is not EXT.
/**

```

Figure 21 (Part 5 of 8). Sample JCL for ACBJBAD1.

```

/** EXTADDRS : Specify this field to provide datasets with      *
/** addressability of more than 4 GB.                          *
/**                                                         *
/** Possible values :                                         *
/**                                                         *
/** Y -> Provides extended addressability if                 *
/** IFEXT => P or R                                          *
/** N -> Doesn't provide extended addressability.           *
/**                                                         *
/** RECACCB : Use this field to specify to VSAM, how the     *
/** buffers are to be chosen and the manner by              *
/** which they are processed.                                *
/**                                                         *
/** Possible values :                                         *
/**                                                         *
/** S -> VSAM to determine buffering algorithm              *
/** U -> Buffers are obtained in the same manner           *
/** as is used with out SYSTEM MANAGED                     *
/** BUFFERING.                                              *
/** BLANK -> U assumed if DSNMTYP => EXT                   *
/**                                                         *
/** REUSE : Specify whether or not the users can open the   *
/** cluster again and again as a new cluster.               *
/**                                                         *
/** Possible values :                                         *
/**                                                         *
/** Y -> Reusable                                           *
/** N -> Non Reusable                                       *
/**                                                         *
/** INILOAD : Specify whether or not storage allocated to the *
/** data component is to be preformatted before            *
/** records are inserted during initial load.              *
/**                                                         *
/** Possible values :                                         *
/**                                                         *
/** S -> No preformat                                        *
/** R -> Data components control areas written             *
/** with records that indicate EOF.                        *
/** BLANK -> Default R is assumed                          *
/**                                                         *
/** SPANONSP : Specify whether a data record is allowed to  *
/** cross Control Interval boundaries. This                 *
/** attribute cannot be defined while defining              *
/** linear dataset cluster.                                 *
/**                                                         *
/** Possible values :                                         *
/**                                                         *
/** S -> record can be spanned across CIs                   *
/** N -> Record must be contained in one CI                 *
/** BLANK -> Default N is assumed.                         *
/**                                                         *
/** BWO : Specify this field if Backup-While-Open (BWO)    *
/** is allowed for sphere. BWO applies only to             *
/** SMS VSAM datasets and can not be used with             *
/** TYPE(LINEAR).                                           *

```

Figure 21 (Part 6 of 8). Sample JCL for ACBJBAD1.


```

/*
/*          Possible values :
/*
/*          TC    -> Use TYPECICS for CICS VSAM file control
/*                  datasets. CICS use BWO for RLS and FVT
/*                  for Non RLS processing.
/*          TI    -> Use TYPEIMS for IMS VSAM datasets
/*          NO    -> BWO doesn't apply to the cluster
/*          BLANK -> CICS uses FCT definition for RLS and
/*                  non RLS processing.
/*
/* LOG      : Specify whether the sphere to be accessed with
/*            VSAM record level sharing(RLS) is recoverable
/*            or non-recoverable.
/*
/*          Possible values :
/*
/*          N[one] -> Non recoverable
/*          U[ndo] -> recoverable
/*          All    ->
/*          Blank  ->
/*
/* LOGSTID : This field identifies the CICS recovery log stream.
/*
/*          Possible values :
/*
/*          Dataset name of the recovery log stream
/*          BLANK
/*
/* SPCCONRL : Specify the DC attributes that will be used by
/*            the system to retry allocation and extension
/*            failures for SMS datasets.
/*
/*          Possible values :
/*
/*          Y -> SMS will re-drive the allocation after
/*                reducing the required space quantity
/*                based on the REQUIRED SPACE UPTO parm.
/*          N -> Default; SMS will not attempt a retry
/*
/* REDSPCUT : specify this field ( REDUCE SPACE UPTO) in order
/*            to reduce the amount of requested quantity by x%
/*            and re-drive the best fit allocation.
/*
/*          Possible values : 0 - 99, BLANK.
/*
/*
/*
/******
/******
/******
//STEP1 EXEC ACBJBAOB,TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAD1 +
DEFINE/ALTER +

```

Figure 21 (Part 7 of 8). Sample JCL for ACBJBAD1.

```

SCDS(TEST.CDS) +
DCNAME() +
DESCR() +
RECORG() +
RECFM() +
LRECL() +
KEYLEN() +
KEYOFF() +
SPCAVREC() +
SPCAVVAL() +
SPCPRM() +
SPCSEC() +
SPCDIR() +
REXPPDT() +
VOLCNT() +
ADDVOLAM() +
IMBED() +
REPLCAT() +
CISZDATA() +
FRSPCCI() +
FRSPCCA() +
SHRXREG() +
SHRXSYS() +
COMPTN() +
MDTYPE() +
RECTECH() +
DSNMTYP() +
IFEXT() +
EXTADDRS() +
RECACCB() +
REUSE() +
INILOAD() +
SPANONSP() +
BWO() +
LOG() +
LOGSTID() +
SPCCONRL() +
REDSPCUT() +
) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(999999)
/*
//STEP2 EXEC ACBJBAOB,TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAD1 +
DISPLAY +
SCDS(TEST.CDS) +
DCNAME() +
) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(999999)
//

```

Figure 21 (Part 8 of 8). Sample JCL for ACBJBAD1.

Define/Alter/Display Storage Class: ACQBAS1

SYS1.SACBCNTL member ACBJBAS1 calls ACQBAS1 to define, alter, or display the storage class in batch.

See Figure 22 for the sample JCL and parameters.

Use the following parameters with ACQBAS1:

Parameters	Description
ACCSBTY	Accessibility
AVALBTY	Availability
CFCACSTN	CF cache set name
CFDTWGHT	CF direct weight
CFSEQWHT	CF sequential weight
DESCR	Task description
DMSRESP	Direct millisecond response
DRTBIAS	Direct bias
GUASYNWR	Guaranteed synchronous write
GURNTSPC	Guaranteed space
INIARESS	Initial access response seconds
SCDS	CDS name
SEQBIAS	Sequential bias
SEQMSRES	Sequential millisecond response
STCNAME	Storage class name
SUSDTRT	Sustained data rate (MB/sec)

Sample JCL for Define/Alter/Display Storage Class

```

//*****
//*
//* SAMPLE JCL TO DEFINE/ALTER/DISPLAY STORAGE CLASSES IN BATCH
//*
//* INSTRUCTIONS BEFORE SUBMITTING:
//*
//* CHANGE JOBCARD
//* CHANGE PREFIX
//* CHANGE PARAMETERS
//*
//* PARAMETER FOLLOWING ACQBAS1 - DEFINE OR ALTER OR DISPLAY
//***** ADD BEG *****
//*
//* SCDS : Name of the SCDS, 1-44 characters.
//*
//* STCNAME : Storage class being defined/alterd.
//*

```

Figure 22 (Part 1 of 4). Sample JCL for ACBJBAS1.

```

/**          1-8 alphanumeric characters , beginning with      *
/**          alphabet.                                         *
/**          *                                                 *
/** DESCRC   : Remarks about the SC being created. 1-120      *
/**          alphanumeric characters.                          *
/**          *                                                 *
/** DMSRESP  : Use DIRECT MILLISECOND RESPONSE field to specify *
/**          how quickly the system will read or write data in *
/**          4K blocks on direct access devices. Enter the     *
/**          response time in milliseconds. All input and output *
/**          requests are processed in single 4k bytes.        *
/**          *                                                 *
/**          Possible Values : 1 - 999, BLANK                  *
/**          *                                                 *
/** DRTBIAS  : Specify this field ( DIRECT BIAS) to indicate  *
/**          whether the majority of the I/O scheduled for DSs *
/**          in this SC will be READ,WRITE or UNKNOWN.        *
/**          *                                                 *
/**          Possible Values : R, W, BLANK                     *
/**          *                                                 *
/** SEQMSRES : Use SEQUENTIAL MILLISECOND RESPONSE field to *
/**          specify how quickly the system will read or       *
/**          write data in 4K blocks on sequentially accessed  *
/**          devices. Enter the response time in milliseconds. *
/**          All I/O requests are processed in single 4K blocks. *
/**          *                                                 *
/**          Possible Values : 1 - 999, BLANK                  *
/**          *                                                 *
/** SEQBIAS  : Specify this field ( SEQ BIAS) to indicate     *
/**          whether the majority of the I/O scheduled for DSs *
/**          in this SC will be READ,WRITE or UNKNOWN.        *
/**          *                                                 *
/**          Possible Values : R, W, BLANK                     *
/**          *                                                 *
/** INIARESS : Use this field (INITIAL ACCESS RESPONSE SECONDS) *
/**          to specify the desired response time in SECS     *
/**          to locate, mount, and prepare media for          *
/**          data transfer.                                    *
/**          *                                                 *
/**          Possible Values : 0 - 9999, BLANK                 *
/**          *                                                 *
/** SUSDTRT  : Use this field (SUSTAINED DATA RATE (MB/SEC)) to *
/**          specify the sequential data transfer rate you     *
/**          want for striped datasets in this SC. The system  *
/**          uses this value to determine the no.of stripes it *
/**          will attempt to allocate for the datasets. If you *
/**          enter ZERO or BLANK the system will attempt to   *
/**          allocate them with one stripe. Only extended     *
/**          sequential datasets can be striped.              *
/**          *                                                 *
/**          Possible Values : 0 - 999, BLANK                  *
/**          *                                                 *
/** AVALBTY  : Specify whether dataset processing should continue *
/**          after device failures.                            *
/**          *

```

Figure 22 (Part 2 of 4). Sample JCL for ACBJBAS1.

```

/**          Possible Values :          *
/**          *                          *
/**          C -> Continue to process a DS if a device failure *
/**          severs communications with the volumes that *
/**          contains the dataset. *
/**          S -> Continuous processing is unavailable after *
/**          device failures. *
/**          P -> Data may be placed on devices that support *
/**          continuous processing. *
/**          N -> Data is placed on any volume. There is no *
/**          preference among volumes. *
/**          *                          *
/**          ACCSBTY : ACCESSIBILITY field specifies whether the datasets *
/**          in this SC be allocated to volumes supported by *
/**          concurrent copy. When used with the MGMTCLS *
/**          ABACKUP/BACKUP COPY TECHNIQUE attributes, this *
/**          field determines if DS should retain write access *
/**          during backup. *
/**          *                          *
/**          Possible Values :          *
/**          *                          *
/**          C -> Datasets must be allocated to the volumes *
/**          supported by concurrent copy. *
/**          S -> Datasets should be allocated to the volumes *
/**          supported by concurrent copy. *
/**          P -> Datasets should be allocated to the volumes *
/**          not supported by concurrent copy. *
/**          N -> Datasets should be allocated to the volumes *
/**          whether the volumes support concurrent copy *
/**          or not. *
/**          *                          *
/**          GURNTSPC : GUARANTEED SPACE field specifies whether to use *
/**          the JCL VOL=SER= parameter to reserve space on *
/**          specific volumes for datasets in this SC. *
/**          *                          *
/**          Possible Values :          *
/**          *                          *
/**          Y -> Reserve space for datasets on the volumes *
/**          specified in the JCL VOL=SER= parameter. *
/**          N -> Don't use volumes requested in the JCL *
/**          VOL=SER= parameter. *
/**          *                          *
/**          GUASYNWR : GUARANTEED SYNCHRONOUS WRITE field indicates *
/**          whether the system should return from a BSAM *
/**          CHECK (or WAIT) issued for a WRITE against a *
/**          partitioned dataset extended (PDSE) member or a *
/**          compressed format data set before (unsynchronized) *
/**          or after (synchronized) the data has been written *
/**          to a storage device. *
/**          *                          *
/**          Possible Values :          *
/**          *                          *
/**          Y -> Indicates synchronized write. *
/**          *                          *
/**          N -> Indicates no synchronization. *

```

Figure 22 (Part 3 of 4). Sample JCL for ACBJBAS1.

```

/*
/* CFCACSTN : Coupling Facility Cache set name.
/*
/* Possible Values :
/*
/* 1-8 alphanumeric characters with beginning char
/*
/* CFDTWGHT : Specifies relative importance of data in a SC
/* when it is accessed directly.
/*
/* Possible Values : 1 - 11, BLANK
/*
/* CFSEQWHT : Specifies relative importance of data in a SC
/* when it is accessed sequentially.
/*
/* Possible Values : 1 - 11, BLANK
/*
/****** ADD END *****
/******
//STEP1 EXEC ACBJBAOB,TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACQBAS1 DEFINE/ALTER +
SCDS(TEMP.SCDS) +
STCNAME() +
DESCR() +
DMSRESP() +
DRTBIAS() +
SEQMSRES() +
SEQBIAS() +
INIARESS() +
SUSDTRT() +
AVALBTY() +
ACCSBTY() +
GURNTSPC() +
GUASYNWR() +
CFCACSTN() +
CFDTWGHT() +
CFSEQWHT() +
)
/*
//STEP2 EXEC ACBJBAOB,TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACQBAS1 DISPLAY +
SCDS(TEMP.SCDS) +
STCNAME() +
)
//

```

Figure 22 (Part 4 of 4). Sample JCL for ACJBAS1.

Define/Alter Dummy Storage Group: ACBQBAJA

SYS1.SACBCNTL member ACBJBAJA calls ACBQBAJA to define or alter dummy type storage groups in batch.

See Figure 23 for the sample JCL and parameters.

Use the following parameters with ACBQBAJA:

Parameters	Description
DESCR	Task description
SCDS	CDS name
STORGRP	Storage group name

Sample JCL for Define/Alter Dummy Storage Group

```

//*****
//*
//* SAMPLE JCL TO DEFINE/ALTER DUMMY STORAGE GROUPS IN BATCH
//*
//* INSTRUCTIONS BEFORE SUBMITTING:
//*
//* CHANGE JOBCARD
//* CHANGE PREFIX
//* CHANGE PARAMETERS
//*
//* PARAMETER FOLLOWING ACBQBAJA - DEFINE OR ALTER
//* SCDS : Name of SCDS, length 1-44 characters
//*
//* STORGRP : Name of the storage group
//*
//* DESCR : Remarks about the Storage group being defined /
//* altered. Maximum length up to 120 characters.
//*
//*
//*****
//*****
//STEP1 EXEC ACBJBAOB,TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAJA +
DEFINE/ALTER +
SCDS(TEST.CDS) +
STORGRP() +
DESCR() +
)
/*

```

Figure 23. Sample JCL for ACBJBAJA.

Define/Alter Object Backup Storage Group: ACBQBAJ7

SYS1.SACBCNTL member ACBJBAJ7 calls ACBQBAJ7 to define or alter object backup type storage groups in batch.

See Figure 24 for the sample JCL and parameters.

Use the following parameters with ACBQBAJ7:

Parameters	Description
DESCR	Task description
DRVSTHRS	Drive start threshold
LIBNAME	Library names Note: LIBNAME can accept up to eight values separated by commas
SCDS	CDS name
SGSTATUS	SMS storage group status Note: SGSTATUS can accept up to 32 values separated by commas. It is up to the user to ensure that the status of all the system or sysgroup names listed in the base configuration are listed here. The user can also use an asterisk (*) to leave the status unmodified for a particular system or sysgroup. The following example will change the status of the third system or sysgroup. The status of systems or sysgroups 1, 2, 4, and so on, will not be altered. SGSTATUS(*,*,ENABLE)
STORGRP	Storage group name
VOLFTHRS	Volume full threshold
VOLFWERR	Volume full at write error

Sample JCL for Define/Alter Object Backup Storage Group

```
//*****
//*
//* SAMPLE JCL TO DEFINE/ALTER OBJ BACKUP TYPE STORAGE GROUP IN BATCH*
//*
//* INSTRUCTIONS BEFORE SUBMITTING:
//*
//* CHANGE JOBCARD
//* CHANGE PREFIX
//* CHANGE PARAMETERS
//*
//* PARAMETER FOLLOWING ACBQBAJ7 - DEFINE OR ALTER
//*****
```

Figure 24 (Part 1 of 3). Sample JCL for ACBJBAJ7.


```

/** SCDS      : Name of SCDS, length 1-44 characters      *
/**
/** STORGRP   : Name of the storage group being defined/  *
/**
/** DESCR    : Remarks about the Storage group being defined/ *
/**            altered.  Maximum length up to 120 characters. *
/**
/** LIBNAME   : Library name that represents a defined optical *
/**            drive configuration or Pseudo library name that *
/**            represents stand-alone optical drives and shelf *
/**            resident optical volumes.                    *
/**
/**            Possible values ;                            *
/**
/**            You can specify up to 8 libraries of 1-8 *
/**            alphanumeric chars each.                    *
/**
/** VOLFTHRS  : No. of free sectors required for an optical *
/**            volume (VOLUME FULL THRESHOLD) within this SG. *
/**
/**            Possible values ; 0 - 9999                  *
/**
/** DRVSTHRS  : Max no of outstanding object write requests *
/**            for an optical drive (DRIVE START THRESHOLD). *
/**
/**            Possible values ; 0 - 9999                  *
/**
/** VOLFWERR  : This field VOLUME-FULL-AT-WRITE ERROR indicates *
/**            whether a " mark volume full on first write *
/**            failure" criteria applies to optical volumes *
/**            in this SG.                                *
/**
/**            Possible values ;                            *
/**
/**            Y -> OAM marks an optical volume full the first *
/**            time an attempt to write an object on the *
/**            optical volume fails, because insufficient *
/**            space.                                       *
/**            N -> OAM marks an object volume full only when *
/**            the no. of available sectors in the user *
/**            data area falls below the VOLUME FULL *
/**            THRESHOLD.                                   *
/**
/** SGSTATUS  : This field sets the status of this object or object *
/**            backup storage group, on each system in the SMS *
/**            complex.                                     *
/**
/**            Possible values ;                            *
/**
/**            NOTCON -> The system can not process this object *
/**            backup storage group. NOTCON is default. *
/**
/**            ENABLE -> The system can access the Volume set. *
/**

```

Figure 24 (Part 2 of 3). Sample JCL for ACBJBAJ7.

```

|      /*          DISNEW -> The system can use the OSREQ functions      *
|      /*          RETRIEVE and DELETE but not STORE.                    *
|      /*
|      /*          DISALL -> The system can use the OSREQ functions      *
|      /*          RETRIEVE, DELETE or STORE.                            *
|      /*          Up to 32 statuses can be specified separated by      *
|      /*          commas to match the 32 systems. If a status is      *
|      /*          skipped, the system status that falls in between two*
|      /*          commas will have default value of ENABLE.            *
|      /*
|      /******
|      /******
|      /*STEP1 EXEC ACBJBAOB,TABL2=userid.TEST.ISPTABL
|      /*SYSUDUMP DD SYSOUT=*
|      /*SYSTSIN DD *
|      PROFILE PREFIX(IBMUSER)
|      ISPSTART CMD(ACBQBAJ7 +
|      DEFINE/ALTER +
|      SCDS(TEST.CDS) +
|      STORGRP(SGOBJBKP) +
|      DESCR() +
|      LIBNAME() +
|      VOLFTHRS() +
|      DRVSTHRS() +
|      VOLFWERR() +
|      SGSTATUS() +
|      )
|      /*
|      //

```

Figure 24 (Part 3 of 3). Sample JCL for ACBJBAJ7.

Define/Alter Object Storage Group: ACBQBAJB

SYS1.SACBCNTL member ACBJBAJB calls ACBQBAJB to define or alter object type storage groups in batch.

See Figure 25 for the sample JCL and parameters.

Use the following parameters with ACBQBAJB:

Parameters	Description
CYCLEET	Cycle end time
CYCLEST	Cycle start time
DESCR	Task description
DRVSTHRS	Drive start threshold
LIBNAME	Library names Note: LIBNAME can accept up to eight values separated by commas
QUALFR	Qualifier
SCDS	CDS name
SGSTATUS	SMS storage group status Note: SGSTATUS can accept up to 32 values separated by commas. It is up to the user to ensure that the status of all the system or sysgroup names listed in the base configuration are listed here. The user can also use an asterisk (*) to leave the status unmodified for a particular system or sysgroup. The following example will change the status of the third system or sysgroup. The status of systems or sysgroups 1, 2, 4, and so on, will not be altered. SGSTATUS(*,*,ENABLE)
STORGRP	Storage group name
VOLFTHRS	Volume full threshold
VOLFWERR	Volume full at write error

Sample JCL for Define/Alter Object Storage Group

```
//*****
//*
//* SAMPLE JCL TO DEFINE/ALTER OBJECT TYPE STORAGE GROUPS IN BATCH
//*
//* INSTRUCTIONS BEFORE SUBMITTING:
//*
//* CHANGE JOBCARD
//* CHANGE PREFIX
```

Figure 25 (Part 1 of 3). Sample JCL for ACBJBAJB.

```

/**      CHANGE PARAMETERS      *
/**      *
/**      PARAMETER FOLLOWING ACBQBAJB - DEFINE OR ALTER      *
/**      *****      *
/**      SCDS      : Name of SCDS, length 1-44 characters      *
/**      *
/**      STORGRP   : Name of the storage group being defined/changed.      *
/**      *
/**      DESCR    : Remarks about the Storage group being defined /      *
/**      altered. Maximum length up to 120 characters.      *
/**      *
/**      QUALFR   : Designates both the DB2 database and High      *
/**      level qualifier (HLQ) for the DB2 tables in      *
/**      in the database for that SG.      *
/**      *
/**      Possible values ; 1 - 8 characters.      *
/**      *
/**      CYCLEST   : Beginning of a window of time in which the      *
/**      OAM can begin storage management processing      *
/**      for this SG.      *
/**      *
/**      Possible values ; 0 - 23 (HR of DAY), Blank      *
/**      *
/**      CYCLEET   : End of a window of time in which the OAM      *
/**      can begin storage management processing for      *
/**      this SG.      *
/**      *
/**      Possible values ; 0 - 23 (HR of DAY), Blank      *
/**      *
/**      LIBNAME   : Library name that represents a defined optical      *
/**      drive configuration or Pseudo library name that      *
/**      represents stand-alone optical drives and shelf      *
/**      resident optical volumes.      *
/**      *
/**      Possible values ;      *
/**      *
/**      You can specify up to 8 libraries of 1-8      *
/**      alphanumeric chars each.      *
/**      *
/**      VOLFTHRS  : No of free sectors required for an optical      *
/**      volume (VOLUME FULL THRESHOLD) within this SG.      *
/**      *
/**      Possible values ; 0 - 9999      *
/**      *
/**      DRVSTHRS  : Max no of outstanding object write requests      *
/**      for an optical drive (DRIVE START THRESHOLD).      *
/**      *
/**      Possible values ; 0 - 9999      *
/**      *
/**      VOLFWERR  : This field VOLUME-FULL-AT-WRITE ERROR indicates      *
/**      whether a " mark volume full on first write      *
/**      failure" criteria applies to volumes in this SG.      *
/**      or not.      *
/**      *
/**      Possible values ;      *

```

Figure 25 (Part 2 of 3). Sample JCL for ACBQBAJB.

```

/**
/**          Y -> OAM marks an optical volume full the first
/**          time an attempt to write an object on the
/**          optical volume fails, because of insufficient
/**          space.
/**          N -> OAM marks an object volume full only when
/**          the no.of available sectors in the user
/**          data area falls below the VOLUME FULL
/**          THRESHOLD.
/**
/** SGSTATUS : This field sets the status of this object storage
/**          storage group, on each system in the SMS complex.*
/**
/**          Possible values ;
/**
/**          NOTCON -> The system can not process this object
/**          storage group. NOTCON is default.
/**
/**          ENABLE -> The system can access the object storage
/**          hierarchy for object storage groups.
/**
/**          DISNEW -> The system can use the OSREQ functions
/**          RETRIEVE and DELETE but not STORE.
/**
/**          DISALL -> The system can use the OSREQ functions
/**          RETRIEVE, DELETE or STORE.
/**          Up to 32 statuses can be specified separated by
/**          commas to match the 32 systems. If a status is
/**          skipped, the system status that falls in between two*
/**          commas will have default value of ENABLE.
/**
/**
/*******
/*******
//STEP1 EXEC ACBJBAOB,TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAJB +
DEFINE/ALTER +
SCDS(TEST.CDS) +
STORGRP() +
DESCR() +
QUALFR() +
CYCLEST() +
CYCLEET() +
LIBNAME() +
VOLFTHRS() +
DRVSTHRS() +
VOLFWERR() +
SGSTATUS() +
)
/*
//

```

Figure 25 (Part 3 of 3). Sample JCL for ACBJBAJB.

Define/Alter Pool Storage Group: ACBQBAJ2

SYS1.SACBCNTL member ACBJBAJ2 calls ACBQBAJ2 to define or alter pool type storage groups in batch.

See Figure 26 on page 73 for the sample JCL and parameters.

Use the following parameters with ACBQBAJ2:

Parameters	Description
AUTOBKUP	Automatic backup
AUTODUMP	Automatic dump
AUTOMIG	Automatic migration
BKUPSYS	Backup system or system group name
DESCR	Task description
DUMPCLAS	Dump class Note: DUMPCLAS can accept up to 5 values separated by commas.
DMPYSNM	Dump system or system group name
GUARBKFR	Guaranteed backup frequency
HIGHTHRS	High allocation/migration threshold
LOWTHRS	Low allocation/migration threshold
SCDS	CDS name
SGSTATUS	SMS storage group status Note: SGSTATUS can accept up to 32 values separated by commas. It is up to the user to ensure that the status of all the system or sysgroup names listed in the base configuration are listed here. The user can also use an asterisk (*) to leave the status unmodified for a particular system or sysgroup. The following example will change the status of the third system or sysgroup. The status of systems or sysgroups 1, 2, 4, and so on, will not be altered. SGSTATUS(*,*,ENABLE)
STORGRP	Storage group name

Sample JCL for Define/Alter Pool Storage Group

```
//*****  
//*  
//* SAMPLE JCL TO DEFINE/ALTER POOL TYPE STORAGE GROUPS IN BATCH *  
//*  
//* INSTRUCTIONS BEFORE SUBMITTING: *  
//*  
//* CHANGE JOBCARD *  
//* CHANGE PREFIX *  
//* CHANGE PARAMETERS *  
//*****  
//* SCDS : Name of SCDS, length 1-44 characters *  
//*  
//* STORGRP : Name of the storage group *  
//*  
//* DESCR : Remarks about the Storage group being defined / *  
//* altered. Maximum length up to 120 characters. *  
//*  
//* AUTOMIG : Specifies whether the datasets on volumes in this *  
//* storage group can be moved to DASD or TAPE by the *  
//* primary space management and interval migration *  
//* functions of DFSMSshm. Possible values are, *  
//*  
//* Y -> Datasets are eligible for primary space *  
//* management migration. *  
//*  
//* N -> Datasets are not eligible for automatic *  
//* migration. *  
//*  
//* I -> Datasets are eligible for primary space *  
//* management and interval migration. *  
//*  
//* P -> Datasets are eligible for primary space *  
//* management but not interval migration. *  
//*  
//* MIGSYSNM : Name of the system or system group where automatic *  
//* migration and space management of the volumes in *  
//* this storage group would be performed. Possible *  
//* values: System or system group names 1 - 8 *  
//* alphanumeric characters in length. *  
//*  
//* AUTOBKUP : Specifies whether all the volumes in the storage *  
//* are eligible for automatic backup. Possible values *  
//* are Y -> Yes and N -> No *  
//*  
//* BKUPSYS : Name of the system or system group where automatic *  
//* backup function will be processed. *  
//* Possible values: System or system group names 1 - 8 *  
//* alphanumeric characters in length. *  
//*  
//* AUTODUMP : Specifies whether the volumes in this storage group *  
//* are to be eligible for automatic dumping. Possible *  
//* values are Y -> Yes and N -> No. *  
//*
```

Figure 26 (Part 1 of 3). Sample JCL for ACBJBAJ2.

```

/** DMPSYSNM : Name of the system or system group where volumes *
/**           in this storage group will automatically dump to *
/**           backup devices. *
/** *
/** DUMPCLAS : Use this field to specify 1 to 5 dump classes. ISMF *
/**           neither processes nor verifies the values of DUMP *
/**           CLASS. 1 - 8 alphanumeric characters. *
/**           Classes will be separated by ','. *
/** *
/** HIGHTHRS : High value of allocation/migration threshold to *
/**           optimize the use of DASD space in a pool SG. *
/**           Possible values 1-99. *
/** *
/** LOWTHRS  : Low value of allocation/migration threshold to *
/**           optimize the use of DASD space in a pool SG. *
/**           Possible values 0-99. *
/** *
/** GUARBKFR : Specifies the maximum number of days that can elapse *
/**           between backups. Possible values 1 - 9999,NOLIMIT. *
/** *
/** SGSTATUS : specify this field to designate the relationship or *
/**           status between storage groups and the systems in a *
/**           a complex. Possible values are, *
/** *
/**           ENABLE -> System or System Group can allocate *
/**                   and access datasets in SG. Default. *
/**           DISALL -> System or System Group can't allocate *
/**                   or access datasets in SG. *
/**           DISNEW -> System or System Group can't allocate *
/**                   new datasets in the SG. *
/**           NOTCON -> System or System Group is physically *
/**                   disconnected form the SG and can not *
/**                   allocate datasets in it. *
/**           QUIALL -> If JES3, system can not schedule jobs *
/**                   that allocate or access datasets in *
/**                   the SG. In JES2 system uses volumes *
/**                   in the SG for new allocations only if *
/**                   other volumes are not available. *
/**           QUINEW -> If JES3, system can not schedule jobs *
/**                   that allocate new datasets in the SG. *
/**                   In case of JES2, system uses volumes *
/**                   in the SG for new allocations only if *
/**                   other volumes are not available. *
/** *
/**           Up to 32 statuses can be specified separated by *
/**           commas to match the 32 systems. If a status is *
/**           skipped, the system status that falls in between two *
/**           commas will have default value of ENABLE. *
/** *
/** *****
/** *****
/** *
//STEP1 EXEC ACBJBAOB,TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *

```

Figure 26 (Part 2 of 3). Sample JCL for ACBJBAJ2.


```

PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAJ2 +
DEFINE/ALTER +
SCDS(TEST.CDS) +
STORGRP(POOLSTGP) +
DESCR(THIS IS TO DEFINE POOL STORAGE GROUP POOLSTGP) +
AUTOMIG() +
MIGSYSNM() +
AUTOBKUP() +
BKUPSYS() +
AUTODUMP() +
DMPSYSNM() +
DUMPCLAS() +
HIGHTHRS() +
LOWTHRS(50) +
GUARBKFR(NOLIMIT) +
SGSTATUS() +
)
/*
//TEMPFILE DD DSN=&&TEMPFILE,DISP=(NEW,PASS),
// SPACE=(TRK,(1,1)),LRECL=300,RECFM=F,BLKSIZE=300
//*
//STEP2 EXEC ACBJBAOB,TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD DSN=&&TEMPFILE,DISP=(OLD,DELETE,DELETE)
//

```

Figure 26 (Part 3 of 3). Sample JCL for ACBJBAJ2.

Define/Alter Tape Storage Group: ACBQBAJ3

SYS1.SACBCNTL member ACBJBAJ3 calls ACBQBAJ3 to define or alter tape type storage groups in batch.

See Figure 27 for the sample JCL and parameters.

Use the following parameters with ACBQBAJ3:

Parameters	Description
DESCR	Task description
LIBNAME	<p>Library name</p> <p>Note: LIBNAME can accept up to 8 values separated by commas. The user can also use an asterisk (*) to leave the status unmodified for a particular library.</p> <p>The following example will change the status of <i>lib1</i>, <i>lib2</i>, and <i>lib4</i>. The status of <i>lib3</i> and any remaining libraries will not be altered.</p> <p style="text-align: center;">LIBNAME(<i>lib1,lib2,* ,lib4</i>)</p>
SCDS	CDS name
SGSTATUS	<p>SMS storage group status</p> <p>Note: SGSTATUS can accept up to 32 values separated by commas. It is up to the user to ensure that the status of all the system or sysgroup names listed in the base configuration are listed here. The user can also use an asterisk (*) to leave the status unmodified for a particular system or sysgroup.</p> <p>The following example will change the status of the third system or sysgroup. The status of systems or sysgroups 1, 2, 4, and so on, will not be altered.</p> <p style="text-align: center;">SGSTATUS(*,* ,ENABLE)</p>
STORGRP	Storage group name

Sample JCL for Define/Alter Tape Storage Group

```
//*****
//*
//* SAMPLE JCL TO DEFINE/ALTER TAPE TYPE STORAGE GROUPS IN BATCH
//*
//* INSTRUCTIONS BEFORE SUBMITTING:
//*
//* CHANGE JOBCARD
//* CHANGE PREFIX
//* CHANGE PARAMETERS
//*****
```

Figure 27 (Part 1 of 3). Sample JCL for ACBJBAJ3.

```

/** SCDS      : Name of SCDS, length 1-44 characters          *
/**
/** STORGRP   : Name of the storage group that is being defined / *
/**           altered.                                         *
/**
/** DESCR    : Remarks about the Storage group being defined/altere*
/**           Maximum length up to 120 characters.           *
/**
/** LIBNAME   : Specify the name of the tape library to which the *
/**           volume belongs. Possible values, library name of *
/**           up to 1 - 8 characters each. Up to eight libraries *
/**           can be specified separated by commas.           *
/**
/** SGSTATUS  : Use this field to set the status of the TAPE    *
/**           storage group on the system or system groups     *
/**           in the complex.                                   *
/**
/**           Possible Values :                                 *
/**
/**           ENABLE  -> System or System Group can allocate *
/**                   and access datasets in SG. Default.    *
/**           DISALL  -> System or System Group can't allocate *
/**                   or access datasets in SG.               *
/**           DISNEW  -> System or System Group can't allocate *
/**                   new datasets in the SG.                  *
/**           NOTCON  -> System or System Group is physically *
/**                   disconnected form the SG and can not     *
/**                   allocate or access datasets in it.      *
/**           QUIALL  -> If JES3, system can not schedule jobs *
/**                   that allocate new datasets in the SG.   *
/**                   In case of JES2, system uses volumes    *
/**                   in the SG for new allocations only if   *
/**                   other volumes are not available.         *
/**
/**           Up to 32 statuses can be specified separated by *
/**           commas to match the 32 systems. If a status is  *
/**           skipped, the system status that falls in between two*
/**           commas will have default value of ENABLE.        *
/**
/**
/**
/*******
/*******
/**
//STEP1 EXEC ACBJBAOB,TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAJ3 +
DEFINE/ALTER +
SCDS(TEST.CDS) +
STORGRP(TAPESTGP) +
DESCR(THIS IS TO DEFINE TAPE STORAGE GROUP TAPESTGP) +
LIBNAME() +
SGSTATUS() +
)
/*

```

Figure 27 (Part 2 of 3). Sample JCL for ACBJBAJ3.

```
//TEMPFILE DD DSN=&&TEMPFILE,DISP=(NEW,PASS),
// SPACE=(TRK,(1,1)),LRECL=300,RECFM=F,BLKSIZE=300
//*
//STEP2 EXEC ACBJBAOB,TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD DSN=&&TEMPFILE,DISP=(OLD,DELETE,DELETE)
//
```

Figure 27 (Part 3 of 3). Sample JCL for ACBJBAJ3.

Define/Alter VIO Storage Group: ACBQBAJ8

SYS1.SACBCNTL member ACBJBAJ8 calls ACBQBAJ8 to define or alter VIO type storage groups in batch.

See Figure 28 for the sample JCL and parameters.

Use the following parameters with ACBQBAJ8:

Parameters	Description
DESCR	Task description
SCDS	CDS name
SGSTATUS	<p>SMS storage group status</p> <p>Note: SGSTATUS can accept up to 32 values separated by commas. It is up to the user to ensure that the status of all the system or sysgroup names listed in the base configuration are listed here. The user can also use an asterisk (*) to leave the status unmodified for a particular system or sysgroup.</p> <p>The following example will change the status of the third system or sysgroup. The status of systems or sysgroups 1, 2, 4, and so on, will not be altered.</p> <p style="text-align: center;">SGSTATUS(*,*,ENABLE)</p>
STORGRP	Storage group name
VIOMAX	VIO maximum size
VIUNIT	VIO unit

Sample JCL for Define/Alter VIO Storage Group

```
//*****
//*
//* SAMPLE JCL TO DEFINE/ALTER VIO TYPE STORAGE GROUPS IN BATCH
//*
//* INSTRUCTIONS BEFORE SUBMITTING:
//*
//*   CHANGE JOBCARD
//*   CHANGE PREFIX
//*   CHANGE PARAMETERS
//*
//* PARAMETER FOLLOWING ACBQBAJ8 - DEFINE OR ALTER
//*****
```

Figure 28 (Part 1 of 3). Sample JCL for ACBJBAJ8.

```

/** SCDS      : Name of SCDS, length 1-44 characters          *
/**
/** STORGRP   : Name of the storage group  being defined.    *
/**
/** DESCR    : Remarks about the Storage group being defined / *
/**            altered. Maximum length of up to 120 characters. *
/**
/** VIOMAX   : Specify in KB the largest VIO dataset that can be *
/**            created for the SG.                             *
/**
/**            Possible Values : 8 - 2000000                  *
/**
/** VIOUNIT  : Specify this field to select the storage device *
/**            type to be simulated by the VIO SG.           *
/**
/**            Possible Values :Any valid device type        *
/**                               (3380, 3390 etc.)          *
/**
/** SGSTATUS : Use this field to set the status of the VIO   *
/**            storage group on the system or system groups   *
/**            in your complex.                               *
/**
/**            Possible Values :                             *
/**
/**            ENABLE -> System or System Group can allocate *
/**                    and access datasets in SG. Default.   *
/**            DISALL -> System or System Group can't allocate *
/**                    or access datasets in SG.             *
/**            DISNEW -> System or System Group can't allocate *
/**                    new datasets in the SG.                *
/**            NOTCON -> System or System Group is physically *
/**                    disconnected form the SG and can not    *
/**                    allocate datasets in it.                *
/**            QUIALL -> If JES3, system can not schedule jobs *
/**                    that allocate or access datasets in    *
/**                    the SG. In JES2 system uses volumes    *
/**                    in the SG for new allocations only if   *
/**                    other volumes are not available.        *
/**            QUINEW -> If JES3, system can not schedule jobs *
/**                    that allocate new datasets in the SG.  *
/**                    In case of JES2, system uses volumes   *
/**                    in the SG for new allocations only if   *
/**                    other volumes are not available.        *
/**
/**            Up to 32 statuses can be specified separated by *
/**            commas to match the 32 systems. If a status is *
/**            skipped, the system status that falls in between two *
/**            commas will have default value of ENABLE.      *
/**
/**
/*******

```

Figure 28 (Part 2 of 3). Sample JCL for ACBJBAJ8.

```
| //STEP1 EXEC ACBJBAOB,TABL2=userid.TEST.ISPTABL  
| //SYSUDUMP DD SYSOUT=*  
| //SYSTSIN DD *  
| PROFILE PREFIX(IBMUSER)  
| ISPSTART CMD(ACBQBAJ8 +  
| DEFINE/ALTER +  
| SCDS(TEST.CDS) +  
| STORGRP(SGVIO) +  
| DESCR() +  
| VIOMAX() +  
| VIOUNIT() +  
| SGSTATUS() +  
| )  
| /*  
| //
```

| *Figure 28 (Part 3 of 3). Sample JCL for ACBJBAJ8.*

Storage Group Volume Add/Delete: ACBQBAI9

SYS1.SACBCNTL member ACBJBAIB calls ACBQBAI9 to add SMS volumes from a storage group. When performing an ADD or a DELETE on an SMS volume in an SMS complex, run ACBQBAI9 for each MVS image in the SMS complex.

See Figure 29 for the sample JCL and parameters.

Use one of the following DDNAMES to indicate which operation is to occur:

VOLADD lists the volumes to add to a storage group in an SCDS

VOLDEL lists the volumes to delete from a storage group in an SCDS

Sample JCL for Storage Group Volume Add/Delete

Note: To delete a volume, specify:

- SCDSNAME
- SG
- VOL
- VOLDEL for the DDNAME

```

/*****
/*
/* SAMPLE JCL TO ADD NEW VOLUMES AND THEIR STATUS
/*
/* INSTRUCTIONS BEFORE SUBMITTING:
/*
/* CHANGE JOBCARD
/* CHANGE PREFIX
/* CHANGE PARAMETERS
/*
/* TEMPFIL - LISTING DATA SET (OUTPUT)
/* VOLADD - VOLUMES TO BE ADDED (INPUT)
/*
/* PARAMETERS:
/*
/* SCDSNAME - SOURCE CONTROL DATA SET
/* VOL - VOLUME
/* SG - STORAGE GROUP
/* STATUS - STATUS (ENABLE/NOTCON/DISALL/DISNEW/QUIALL/
/* QUINNEW)
/* Up to 32 statuses can be specified separated by
/* commas to match the 32 systems. If a status is
/* skipped, the system status that falls in between
/* 2 commas will have default value of ENABLE.
/* To DELETE volume, specify SCDSNAME, VOL and SG,
/* and use VOLDEL for DDname.
*****/
```

Figure 29 (Part 1 of 2). Sample JCL for ACBJBAIB.


```

//ADDVOL1 EXEC ACBJBAOB,PLIB1='SYS1.DGTPLIB'
//SYSUDUMP DD SYSOUT=*
//TEMPFILE DD DSN=&&VOLADDS,DISP=(NEW,KEEP),
// SPACE=(TRK,(1,1)),LRECL=80,RECFM=F,BLKSIZE=80
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAI9) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//VOLADD DD *
SCDSNAME(MYCDS) VOL(SMS900) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS901) SG(PRIMARY) STATUS(DISALL, +
ENABLE,,DISNEW,,NOTCON,DISALL,ENABLE,QUIALL)
SCDSNAME(MYCDS) VOL(SMS902) SG(PRIMARY) +
STATUS(DISNEW)
SCDSNAME(MYCDS) VOL(SMS903) SG(PRIMARY) +
STATUS(DISALL,DISNEW,ENABLE,NOTCON,DISALL,DISNEW,DISALL,QUIALL, +
DISALL,QUINEW,ENABLE,NOTCON,QUIALL,DISNEW,DISALL,QUIALL, +
DISNEW,ENABLE,ENABLE,DISNEW,QUIALL,DISALL,DISALL,NOTCON, +
NOTCON,QUINEW,ENABLE,NOTCON,QUIALL,DISNEW,ENABLE,QUIALL)
SCDSNAME(MYCDS) VOL(SMS904) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS905) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS906) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS907) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS908) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS909) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS910) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS911) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS912) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS913) SG(PRIMARY) STATUS(ENABLE)
/*
//ADDVOL2 EXEC ACBJBAOB,PLIB1='SYS1.DGTPLIB'
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD DSN=&&VOLADDS,DISP=(OLD,DELETE)
//*****

```

Figure 29 (Part 2 of 2). Sample JCL for ACBJBAIB.

ACS Routine Translate: ACBQBAO1

You can use the ISMF translate option to translate the ACS routines and store them in an SCDS. After successful translation and validation, that SCDS can be activated and used by the system to manage storage. ACBQBAO1 is called by the following SYS1.SACBCNTL members to translate the ACS routines:

ACBJBAC2 Translate ACS routines into an SCDS, validate the SCDS, test ACS routines, and run the NaviQuest ACS comparison utility.
ACBJBAOQ Translate ACS routines into an SCDS, followed by validating the SCDS.

See Figure 30 for the sample JCL and parameters.

Sample JCL for ACS Routine Translate

```
//*****  
//*                                                                 *  
//*  SAMPLE JCL TO DO FOUR OPERATIONS:                             *  
//*                                                                 *  
//*    1. TRANSLATE ACS ROUTINES (ISMF OPTION 7.2)                 *  
//*    2. VALIDATE ACS ROUTINES (ISMF OPTION 7.3)                 *  
//*    3. TEST ACS ROUTINES (ISMF OPTION 7.4.3)                   *  
//*    4. COMPARE BASE & NEW ACS LISTINGS                          *  
//*                                                                 *  
//*  INSTRUCTIONS BEFORE SUBMITTING:                               *  
//*                                                                 *  
//*    CHANGE JOBCARD                                              *  
//*    CHANGE PREFIX                                              *  
//*    CHANGE PARAMETERS                                          *  
//*                                                                 *  
//*****  
//*****  
//*                                                                 *  
//*  TRANSLATE STEP:                                               *  
//*                                                                 *  
//*  ACSSRC   - PDS CONTAINING ACS ROUTINES TO BE TRANSLATED (INPUT)*  
//*  MEMBER   - MEMBER NAME OF THE ROUTINE TO BE TRANSLATED (INPUT) *  
//*  SCDSNAME - NAME OF SCDS INTO WHICH THE ACS ROUTINES ARE TO BE *  
//*            TRANSLATED (OUTPUT)                                  *  
//*  LISTNAME - TRANSLATE LISTING (OUTPUT)                          *  
//*                                                                 *  
//*****  
//TRANSLAT EXEC ACBJBAOB,PLIB1='SYS1.DGTPLIB'  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)
```

Figure 30 (Part 1 of 4). Sample JCL for ACBJBAC2.

```

DEL DATACLAS.LISTING
DEL MGMTCLAS.LISTING
DEL STORGRP.LISTING
DEL STORCLAS.LISTING
ISPSTART CMD(ACBQBA01 +
ACSSRC(ACS.SOURCE) MEMBER(DATACLAS) +
SCDSNAME(MYSCDS) LISTNAME(DATACLAS.LISTING)) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)

ISPSTART CMD(ACBQBA01 +
ACSSRC(ACS.SOURCE) MEMBER(STORCLAS) +
SCDSNAME(MYSCDS) LISTNAME(STORCLAS.LISTING)) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)

ISPSTART CMD(ACBQBA01 +
ACSSRC(ACS.SOURCE) MEMBER(MGMTCLAS) +
SCDSNAME(MYSCDS) LISTNAME(MGMTCLAS.LISTING)) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)

ISPSTART CMD(ACBQBA01 +
ACSSRC(ACS.SOURCE) MEMBER(STORGRP) +
SCDSNAME(MYSCDS) LISTNAME(STORGRP.LISTING)) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)

/*
//*****
//*
//* COPY TRANSLATE LISTINGS
//*
//* SYSUT1 - INPUT (FROM PREVIOUS STEP)
//* SYSUT2 - OUTPUT
//*
//*****
//TRANGEN EXEC PGM=IEBGENER
//SYSUT1 DD DSN=IBMUSER.DATACLAS.LISTING,DISP=SHR
// DD DSN=IBMUSER.STORCLAS.LISTING,DISP=SHR
// DD DSN=IBMUSER.MGMTCLAS.LISTING,DISP=SHR
// DD DSN=IBMUSER.STORGRP.LISTING,DISP=SHR
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//*****
//*
//* VALIDATE STEP:
//*
//* SCDSNAME - NAME OF SCDS THAT CONTAINS THE TRANSLATED ACS
//* ROUTINES TO BE VALIDATED (INPUT)
//* TYPE - TYPE OF ACS ROUTINE TO BE VALIDATED (INPUT)
//* LISTNAME - VALIDATE LISTING (OUTPUT)
//*
//*****
//VALIDAT EXEC ACBJBAOB,PLIB1='SYS1.DGTPLIB'
//SYSTSIN DD *

```

Figure 30 (Part 2 of 4). Sample JCL for ACBJBAC2.

```

DEL VALIDAT.LISTING
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBA02 SCDSNAME(MYSCDS) TYPE(*) +
LISTNAME(VALIDAT.LISTING)) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*****
//*
//*   COPY VALIDATE LISTING
//*
//*   SYSUT1 - INPUT (FROM PREVIOUS STEP)
//*   SYSUT2 - OUTPUT
//*
//*****
//VALGEN   EXEC   PGM=IEBGENER
//SYSUT1   DD    DSN=IBMUSER.VALIDAT.LISTING,DISP=SHR
//SYSUT2   DD    SYSOUT=*
//SYSIN    DD    DUMMY
//SYSPRINT DD    SYSOUT=*
//*****
//*
//*   TEST STEP
//*
//*   SCDSNAME - NAME OF SCDS (INPUT)
//*   TESTBED  - PDS CONTAINING TEST CASES (INPUT)
//*   MEMBER   - MEMBERS TO BE TESTED IN TESTBED (INPUT)
//*   DC,SC,MC,SG - ROUTINES TO BE TESTED (INPUT)
//*   LISTNAME - TEST LISTING (OUTPUT)
//*
//*****
//TEST     EXEC ACBJBAOB,PLIB1='SYS1.DGTPLIB'
//SYSTSIN  DD *
DEL NEW.LISTING
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAIA +
SCDSNAME(MYSCDS) +
TESTBED(TESTCASE.LIBRARY) MEMBER(*) +
DC(Y) SC(Y) MC(Y) SG(Y) +
LISTNAME(NEW.LISTING)) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
//*****
//*
//*   COPY TEST LISTING
//*
//*   SYSUT1 - INPUT (FROM PREVIOUS STEP)
//*   SYSUT2 - OUTPUT
//*
//*****
//TSTGEN   EXEC   PGM=IEBGENER
//SYSUT1   DD    DSN=IBMUSER.NEW.LISTING,DISP=SHR
//SYSUT2   DD    SYSOUT=*
//SYSIN    DD    DUMMY
//SYSPRINT DD    SYSOUT=*

```

Figure 30 (Part 3 of 4). Sample JCL for ACBJBAC2.

```

//*****
//*
//* COMPARE ACS LISTINGS
//*
//* PARAMETERS:
//*
//* BASELIST - BASE ACS TEST LIST (INPUT)
//* NEWLIST - NEW ACS TEST LIST (INPUT)
//* TESTBED - PDS CONTAINING TEST CASES (REFERENCE INPUT)
//* RSLTDSN - COMPARISON RESULTS DATA SET (OUTPUT)
//* XCPTPDS - EXCEPTION TEST CASE PDS (OUTPUT)
//* XCPSPACE - SPACE values of Except DS (Optional) 3a@WA32832*
//* Values: (Primary Tracks,Secondary Tracks,Directory Blocks) *
//* which are positional and optional. Defaults: (3,1,20). *
//*
//* NOTE: If you receive message IEC217I B14-0C on your exception *
//* data set, you need to increase your data set size by using *
//* the XCSPACE parameter (specially the directory blocks) *
//*
//*****
//COMPARE EXEC ACBJBAOB,PLIB1=ISMFMVS.ISMFE.TEST.ISPPLIB,
// TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
DEL PDS4
DEL RESULTS.LISTING
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(%ACBQBAC1 +
BASELIST(BASE.TESTLIST) +
NEWLIST(NEW.TESTLIST) +
TESTBED(TESTCASE.LIBRARY) +
RSLTDSN(COMPARE.LISTING) +
XCPTPDS(TESTCASE.EXCP) +
XCPSPACE(5,3,30)) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*****
//*
//* COPY COMPARISON RESULTS
//*
//* SYSUT1 - INPUT (FROM PREVIOUS STEP)
//* SYSUT2 - OUTPUT
//*
//*****
//CMPGEN EXEC PGM=IEBGENER
//SYSUT1 DD DISP=SHR,DSN=IBMUSER.COMPARE.LISTING
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//*

```

Figure 30 (Part 4 of 4). Sample JCL for ACBJBAC2.

SCDS Validation: ACBQBAO2

ACBQBAO2 is called by the following SYS1.SACBCNTL members to validate the constructs:

ACBJBAOQ Translate all 4 ACS routines into an SCDS and validate SCDS.
ACBJBAC2 Translate all 4 ACS routines into an SCDS, validate the SCDS, run option 7.4.3 (test ACS routines) in batch, run the NaviQuest ACS comparison utility.
ACBJBAOS ISMF batch EXEC for validating the SCDS.

See Figure 31 for the sample JCL and parameters.

Sample JCL for SCDS Validation

```
//*****  
//*                                                                 *  
//* SAMPLE JCL TO PERFORM ISMF ACS VALIDATE IN BATCH                *  
//*                                                                 *  
//* INSTRUCTIONS BEFORE SUBMITTING:                                *  
//*                                                                 *  
//* CHANGE JOBCARD                                                *  
//* CHANGE PREFIX                                                 *  
//* CHANGE PARAMETERS                                             *  
//*                                                                 *  
//*****  
//*****  
//*                                                                 *  
//* VALIDATE STEP:                                                *  
//*                                                                 *  
//* SCDSNAME - NAME OF SCDS THAT CONTAINS THE TRANSLATED ACS      *  
//*          ROUTINES TO BE VALIDATED (INPUT)                     *  
//* TYPE     - TYPE OF ACS ROUTINE TO BE VALIDATED (INPUT)       *  
//* LISTNAME - VALIDATE LISTING (OUTPUT)                           *  
//*                                                                 *  
//*****  
//VALIDAT EXEC ACBJBAOB,PLIB1='SYS1.DGTPLIB'  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
DEL VALIDAT.LISTING  
ISPSTART CMD(ACBQBAO2 +  
SCDSNAME(DSNREPN) TYPE(*) +  
LISTNAME(VALIDAT.LISTING)) +  
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)  
/*
```

Figure 31 (Part 1 of 2). Sample JCL for ACBJBAOS.

```

//*****
//*
//* COPY THE VALIDATE LISTING DATA SET
//*
//*   SYSUT1 - INPUT (FROM PREVIOUS STEP)
//*   SYSUT2 - OUTPUT
//*
//*****
//STEP2 EXEC PGM=IEBGENER
//SYSUT1 DD DSN=IBMUSER.VALIDAT.LISTING,DISP=SHR
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//*****

```

Figure 31 (Part 2 of 2). Sample JCL for ACBJBAOS.

Test ACS Routines: ACBQBAIA

ACBQBAIA is called by the following SYS1.SACBCNTL members to test the ACS routines in batch:

ACBJBAC2 Translate all 4 ACS routines into an SCDS, validate the SCDS, test ACS routines, and run the NaviQuest ACS comparison utility.
ACBJBAIC Test ACS routines.

See Figure 32 for the sample JCL and parameters.

Sample JCL for Test ACS Routines

```
//*****  
//*                                                                 *  
/* SAMPLE JCL TO TEST ACS ROUTINES IN BATCH                       *  
/*                                                                 *  
/* INSTRUCTIONS BEFORE SUBMITTING:                               *  
/*                                                                 *  
/*     CHANGE JOBCARD                                           *  
/*     CHANGE PREFIX                                           *  
/*                                                                 *  
//*****  
//*****  
/*                                                                 *  
/* TEST STEP                                                     *  
/*                                                                 *  
/* SCDSNAME - NAME OF SCDS THAT CONTAINS THE TRANSLATED,       *  
/*             VALIDATED ACS ROUTINES TO BE TESTED (INPUT)     *  
/* TESTBED   - PDS CONTAINING TEST CASES THAT THE ACS ROUTINES *  
/*             SHOULD BE TESTED FOR (INPUT)                     *  
/* MEMBER    - MEMBERS TO BE TESTED IN TESTBED (INPUT)        *  
/* DC,SC,MC,SG - ROUTINES TO BE TESTED Y OR N (INPUT)         *  
/* LISTNAME  - TEST LISTING (OUTPUT)                            *  
/*                                                                 *  
//*****  
//*****  
//TESTACS EXEC ACBJBAOB,PLIB1='SYS1.DGTPLIB'  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
DEL NEW.TESTLIST  
ISPSTART CMD(ACBQBAIA +  
SCDSNAME(MYSCDS) +  
TESTBED(TESTCASE.LIBRARY) MEMBER(*) +  
LISTNAME(NEW.TESTLIST +  
DC(Y) SC(Y) MC(Y) SG(Y)) +  
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)  
/*
```

Figure 32 (Part 1 of 2). Sample JCL for ACBJBAIC.


```

//*****
//*
//* COPY TEST LISTING
//*
//* SYSUT1 - INPUT (FROM PREVIOUS STEP)
//* SYSUT2 - OUTPUT
//*
//*****
//TESTGEN EXEC PGM=IEBGENER
//SYSUT1 DD DSN=IBMUSER.NEW.TESTLIST,DISP=SHR
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*

```

Figure 32 (Part 2 of 2). Sample JCL for ACBJBAIC.

Define/Alter/Display Base Configuration: ACBQBAB1

SYS1.SACBCNTL member ACBJBAB1 calls ACBQBAB1 to define, alter, or display a new base configuration.

See Figure 33 for the sample JCL and parameters.

Use the following parameters with ACBQBAB1:

Parameters	Description
ADDGRPP	Add system group name
ADDSYSP	Add system name
BYTPTRK	Bytes per track
DEFMC	Default management class
DEFUNIT	Default unit
DELGRPP	Delete system group name
DESCR	Task description
RENGRPP	Rename system group name
RENSYSP	Rename system name
SCDS	CDS name
TRKPCYL	Tracks per cylinder

Sample JCL for Define/Alter/Display Base Configuration

```

//*****
//*
//* SAMPLE JCL TO DEFINE/ALTER/DISPLAY BASE CONFIGURATION IN BATCH
//*
//* INSTRUCTIONS BEFORE SUBMITTING:
//*
//* CHANGE JOBCARD
//* CHANGE PREFIX
//* CHANGE PARAMETERS
//*
//* PARAMETER FOLLOWING ACBQBAB1 - DEFINE OR ALTER OR DISPLAY
//*****
//* SCDS : Specify the SCDS whose base configuration you want
//* to define/alter or display.
//*
//* DESCR : Remarks about the SCDS whose base you want to
//* define or alter.
//*
//* Possible values : 1 - 120 characters.
//*

```

Figure 33 (Part 1 of 3). Sample JCL for ACBJBAB1.

```

/*  DEFMC   : Specify this DEFAULT MGMTCLS field to define or      *
/*          alter the MGMTCLS reserved for datasets which         *
/*          have not yet been assigned a MGMTCLS. Available      *
/*          only for SMS managed datasets. Controls the BKP,    *
/*          retention and migration of datasets.                 *
/*          Possible values : Valid MGMTCLS name.                *
/*  DEFUNIT : Specify this field to assign the default device    *
/*          used when non SMS datasets are allocated without     *
/*          a unit parameter,                                    *
/*  BYTPTRK : Specify BYTES/TRACK                                *
/*          Possible values : 1 - 999999                          *
/*  TRKPCYL : Specify TRACKS/CYLENDER                            *
/*          Possible values : 1 - 999999                          *
/*  ADDSYS  : Name of the system you want to add                 *
/*          Possible values : 1-8 Alphanumeric characters       *
/*          beginning with alphabetic.                           *
/*          If more than one system is being added , systems   *
/*          will be separated by ','.                             *
/*          Ex : ADDSYS(SYS1,SYS2,SYS3)                          *
/*  ADDGRP  : Name of the System group that you want to add     *
/*          Possible values : 1-8 Alphanumeric characters.     *
/*          beginning with alphabetic.                           *
/*          If more than one system group is being added, system *
/*          groups will be separated by ','.                     *
/*          Ex : ADDGRP(SGRP1,SGRP2,SGRP3)                      *
/*  DELSYS  : Name of the system you want to delete             *
/*          If more than one system is being deleted, systems  *
/*          will be separated by ','.                             *
/*          Ex : DELSYS(SYS1,SYS2,SYS3)                          *
/*  DELGRP  : Name of the System group that you want to delete *
/*          If more than one system group is being deleted,    *
/*          system groups will be separated by ','.             *
/*          Ex : DELGRP(SGRP1,SGRP2,SGRP3)                      *

```

Figure 33 (Part 2 of 3). Sample JCL for ACBJBAB1.

```

/*
/*  REN SYS  : Old and new names of the system that is being
/*            renamed, separated by ','.
/*
/*
/*            If more than one system is being renamed, systems
/*            will be separated by ','.
/*
/*
/*            Ex : REN SYS(oldname1,newname1,oldname2,newname2)
/*
/*
/*
/*  REN GRP  : Old and new names of the system group  that is being
/*            renamed, separated by ','.
/*
/*
/*            If more than one system group is being renamed,
/*            system groups ll be separated by ','.
/*
/*
/*            Ex : REN GRP(oldname1,newname1,oldname2,newname2)
/*
/*
/*
/******
/******
/******
//STEP1 EXEC ACBJBAOB,TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAB1 DEFINE/ALTER +
SCDS(TEMP.SCDS) +
DESCR() +
DEFMC() +
DEFUNIT() +
BYTPTRK() +
TRKPCYL() +
ADDSYS() +
ADDGRP() +
) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(999999)
/*
/*
//STEP3 EXEC ACBJBAOB,TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAB1 DISPLAY +
SCDS(TEMP.SCDS) +
) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(999999)
/*

```

Figure 33 (Part 3 of 3). Sample JCL for ACBJBAB1.

Define/Alter/Display Aggregate Group: ACBQBAA1

SYS1.SACBCNTL member ACBJBAA1 calls ACBQBAA1 to define, alter, or display a new aggregate group in batch.

See Figure 34 for the sample JCL and parameters.

Use the following parameters with ACBQBAA1:

Parameters	Description
ACCOUNT	Account
AGNAME	Aggregate group name
BDSNO	To browse a data set specify, number
DESCR	Task description
EDSNO	To edit a data set, specify number
INSTRDS	Instruction data set
MCNAME	Management class name
NOOFCPS	Number of copies
ODSPREF	Output data set name prefix
SCDS	CDS name
SELDS	Selection data set

Sample JCL for Define/Alter/Display Aggregate Group

```

//*****
//*
//* SAMPLE JCL TO DEFINE/ALTER/DISPLAY AGGREGATE GROUPS IN BATCH
//*
//* INSTRUCTIONS BEFORE SUBMITTING:
//*
//* CHANGE JOBCARD
//* CHANGE PREFIX
//* CHANGE PARAMETERS
//*
//* PARAMETER FOLLOWING ACBQBAA1 - DEFINE OR ALTER
//*
//* SCDS : Name of the Configuration dataset. 1-44 characters.
//*
//* AGNAME : Name of the Aggregate group to define/alter/display.*
//*
//* Possible values : 1-8 characters
//*
//* DESCR : Remarks about the Aggregate group being defined /
//* altered.
//* Possible values : 1 - 120 alphanumeric characters.
//*

```

Figure 34 (Part 1 of 3). Sample JCL for ACBJBAA1.

```

/* NOOFCPS : Specify no. of copies of the aggregate backup *
/*          output files to be created. The aggregate backup *
/*          output file consists of an instruction activity *
/*          log file, control file and data files. This is a *
/*          required field. *
/* *
/*          Possible values : 1 - 15. *
/* *
/* MCNAME : Name of the MGMTCLS containing the attribute *
/*          values for aggregate backups. *
/* *
/*          Possible values: Any valid 1 - 8 character MGMTCLS *
/*          name. *
/* *
/* ODSPREF : Output dataset name prefix. *
/* *
/*          Possible values : Any 1 - 33 character valid TSO *
/*          dataset name prefix . *
/* *
/* ACCOUNT : Specifies the character string used to provide the *
/*          accounting information parameter required for batch *
/*          jobs. *
/* *
/*          Possible values : 1 - 32 character valid JCL *
/*          accounting information. *
/* *
/* SELDS : Name of the selection dataset containing lists of *
/*          datasets to be included in the application BKP. *
/*          Could be sequential or PDS dataset. In case of PDS *
/*          Specify the member name also. *
/* *
/*          Possible values: Any 1 to 44 character valid TSO *
/*          dataset name. *
/* *
/* INSTRDS : Name of the dataset containing commands and other *
/*          instructions that is copied onto the control file *
/*          volume after the backup control file. Must be *
/*          a sequential file. *
/* *
/*          Possible values: Any 1 to 44 character valid TSO *
/*          dataset name. *
/* *
/******
/******
//STEP1 EXEC ACBJBAOB
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAA1 +
DEFINE/ALTER +
SCDS(TEST.CDS) +
AGNAME(AGGROUP1) +
DESCR() +
NOOFCPS() +
MCNAME() +

```

Figure 34 (Part 2 of 3). Sample JCL for ACBJBAA1.

```

|
| ODSPREF() +
| ACCOUNT() +
| SELDS() +
| INSTRDS() +
| ) +
| BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(999999) *
| /*
| //STEP2 EXEC ACBJBAOB,TABL2=userid.TEST.ISPTABL
| //SYSUDUMP DD SYSOUT=*
| //SYSTSIN DD *
| PROFILE PREFIX(IBMUSER)
| ISPSTART CMD(ACBQBAA1 +
| DISPLAY +
| SCDS(TEST.CDS) +
| AGNAME() +
| ) +
| BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(999999)
| //

```

Figure 34 (Part 3 of 3). Sample JCL for ACBJBAA1.

Generate Test Cases from ISMF-saved Data Set Lists: ACBQBAG3

ACBQBAG3 is called by SYS1.SACBCNTL member ACBJBAG2 to take an ISMF-saved data set table and generate test case members in a PDS library based on the attributes of the data sets in the ISMF table.

See Figure 35 for the sample JCL and parameters.

Notes:

1. Saving tables of temporary data sets might produce errors in the bulk test case generate option (11.1.1). Instead, it is recommended that the user generate test cases from the ACSTST program provided by CBIPO for temporary data sets.
2. The MULTVOL variable is always set to YES in an ISMF table if the data set has not been opened at the time the table is saved. The value is set correctly at OPEN time. This can sometime cause errors in the bulk test case generator.
3. The ACQUIRE DATA FROM VOLUME and ACQUIRE DATA IF DFHSM options under the ISMF data set selection entry panel should be set to Y before generating the list.

Sample JCL for Generating Test Cases from ISMF-saved Data Set Lists

```
//*****  
//*  
//* GENERATE TEST CASES FROM A PREVIOUSLY SAVED ISMF DATA SET LIST *  
//*  
//* TABLENM - SAVED ISMF DATA SET LIST (INPUT) *  
//* PRFX - MEMBER NAME PREFIX *  
//* TESTPDS - TEST CASE PDS (OUTPUT) *  
//* REPLACE - REPLACE CONTENTS IF DSN EXISTS *  
//* INTEST - DEBUG MODE YES/NO (KEEP IT AS NO) *  
//* *  
//*****  
//GENTC EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
ISPSTART CMD(%ACBQBAG3 TABLENM(DSNLIST) +  
INTEST(NO) PRFX(TEST) +  
TESTPDS('IBMUSER.TESTCASE.LIBRARY') REPLACE(Y)) +  
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)  
/*  
//*****
```

Figure 35. Sample JCL for ACBJBAG2.

Generate Test Cases from DCOLLECT Data: ACBQBAG1

ACBQBAG1 is called by SYS1.SACBCNTL member ACBJBAG1 to generate test cases from DCOLLECT data.

See Figure 36 for the sample JCL and parameters.

Sample JCL for Generating Test Cases from DCOLLECT Data

```
//*****  
//*                                                                 *  
//* SAMPLE JCL TO GENERATE TEST CASES FROM DCOLLECT DATA        *  
//*                                                                 *  
//* INSTRUCTIONS BEFORE SUBMITTING:                               *  
//*                                                                 *  
//*   CHANGE JOBCARD                                             *  
//*   CHANGE PREFIX                                             *  
//*   CHANGE PARAMETERS                                          *  
//*                                                                 *  
//* THE PARAMETERS TO ACBQBAG1 ARE AS FOLLOWS:                  *  
//*                                                                 *  
//*   PARAMETER 1 IS DATA SET CONTAINING DCOLLECT DATA (INPUT) *  
//*   PARAMETER 2 IS NUMBER OF TEST CASES TO BE GENERATED      *  
//*   PARAMETER 3 IS MEMBER NAME PREFIX                          *  
//*   PARAMETER 4 IS TEST CASE PDS (OUTPUT)                     *  
//*   PARAMETER 5 IS REPLACE CONTENTS IF DSN EXISTS             *  
//*                                                                 *  
//*****  
//GENTEST EXEC  ACBJBAOB,PLIB1=SYS1.DGTPLIB  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
ISPSTART CMD(%ACBQBAG1 'IBMUSER.DCOLLECT.DATA' 10 TEST +  
'IBMUSER.TESTCASE.LIBRARY' Y) +  
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)  
/*  
//*****
```

Figure 36. Sample JCL for ACBJBAG1.

Generate Test Cases from SMF Data

SYS1.SACBCNTL member ACBJBA11 contains JCL to generate test cases from SMF type 127 type records in batch.

You will need to provide the names of the input and output data sets on the INDD and OUTDD DD statements

INDD Data set containing SMF data (input)

OUTDD Test case PDS (output)

See Figure 37 for the sample JCL and parameters.

SMF type 127 records are written by storage class exit IGDACSSC. Use a utility like IFASMFDP to unload the SMF type 127 records from the SMF log data sets. You can also use ACSTST program (from CBIPO).

Sample JCL for Generating Test Cases from SMF Data

```
//*****  
//*                                                                 *  
//* SAMPLE JCL TO GENERATE TEST CASES FROM SMF TYPE 127 RECORDS *  
//* WRITTEN BY THE STORAGE CLASS EXIT IGDACSSC; USE A STANDARD *  
//* UTILITY LIKE IFASMFDP TO UNLOAD THE TYPE 127 RECORDS FROM THE *  
//* SMF LOG DATA SETS; ALSO ACSTST PROGRAM (FROM CBIPO) SHOULD BE *  
//* AVAILABLE                                                                 *  
//*                                                                 *  
//*   INDD - DATA SET CONTAINING SMF DATA (INPUT) *  
//*   OUTDD - TEST CASE PDS (OUTPUT) *  
//*                                                                 *  
//*****  
//GENTEST EXEC PGM=ACSTST,REGION=512K,  
//           COND=(0,NE,DATA1)  
//SYSPRINT DD SYSOUT=*  
//INDD     DD DSN=IBMUSER.SMF.TYPE127.DATA,DISP=SHR  
//OUTDD    DD DSN=IBMUSER.TESTCASE.LIBRARY,DISP=(NEW,CATLG,DELETE),  
//           SPACE=(80,(1250,200,50),,ROUND),  
//           UNIT=3380,VOL=SER=M4RS05,  
//           DCB=(BLKSIZE=80,LRECL=80,RECFM=F)  
//*****
```

Figure 37. Sample JCL for ACBJBA11.

Generate Test Cases from VMA Extract Data: ACBQBAO3

ACBQBAO3 is called by SYS1.SACBCNTL member ACBJBAOW to generate test cases from records in the VMA extract file of previously created VMA data.

See Figure 38 for the sample JCL and parameters.

Sample JCL for Generating Test Cases from VMA

Note: You can also use ACSTST program (from CBIPO).

```
//*****  
//*                                                                 *  
//* SAMPLE JCL TO GENERATE TEST CASES FROM VMA EXTRACT DATA      *  
//*                                                                 *  
//* INSTRUCTIONS BEFORE SUBMITTING:                                *  
//*                                                                 *  
//*     CHANGE JOBCARD                                             *  
//*     CHANGE PREFIX                                              *  
//*     CHANGE PARAMETERS                                          *  
//*                                                                 *  
//* PARAMETERS:                                                  *  
//*                                                                 *  
//*     PARAMETER 1 - DATA SET CONTAINING VMA EXTRACT DATA      *  
//*     PARAMETER 2 - NUMBER OF TEST CASES TO BE GENERATED      *  
//*     PARAMETER 3 - MEMBER NAME PREFIX                          *  
//*     PARAMETER 4 - PROGRAM NAME TO FILTER ON                  *  
//*                                                                 *  
//*****  
//TESTGEN EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
ISPSTART CMD(%ACBQBAO3 'IBMUSER.VMA.DATA' 100 TEST) +  
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)  
/*
```

Figure 38. Sample JCL for ACBJBAOW.

ACS Test Listings Comparison: ACBQBAC1

ACBQBAC1 is called by the following SYS1.SACBCNTL members to perform a detailed comparison of the differences between the "base" and "new" ACS listing:

ACBJBAC1 Run the NaviQuest ACS Comparison utility.
ACBJBAC2 Translate all 4 ACS routines into an SCDS, validate the SCDS, test ACS routines, and run the NaviQuest ACS comparison utility.

See Figure 39 for the sample JCL and parameters.

Sample JCL for Test Listings Comparison

```

| //*****
| //*
| //* SAMPLE JCL TO COMPARE ACS TEST LISTINGS IN BATCH
| //*
| //* INSTRUCTIONS BEFORE SUBMITTING:
| //*
| //* CHANGE JOBCARD
| //* CHANGE PREFIX
| //* CHANGE PARAMETERS
| //*
| //* PARAMETERS:
| //*
| //* BASELIST - BASE ACS TEST LISTING (INPUT)
| //* NEWLIST - NEW ACS TEST LISTING (INPUT)
| //* TESTBED - TEST CASE PDS (REFERENCE INPUT)
| //* RSLTDSN - COMPARISON RESULTS DATA SET (OUTPUT)
| //* XCPTPDS - EXCEPTION TEST CASE PDS (OUTPUT)
| //* XCPSPACE - SPACE values of Except DS (Optional) 3a@WA32832*
| //* Values: (Primary Tracks,Secondary Tracks,Directory Blocks) *
| //* which are positional and optional. Defaults: (3,1,20). *
| //*
| //* NOTE: If you receive message IEC217I B14-0C on your exception *
| //* data set, you need to increase your data set size by using *
| //* the XCSPACE parameter (specially the directory blocks) *
| //*
| //*****
| //CMPRSTEP EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
| //SYSTSIN DD *
| PROFILE PREFIX(IBMUSER)
| DEL COMPARE.LISTING
| DEL TESTCASE.EXCP

```

Figure 39 (Part 1 of 2). Sample JCL for ACBJBAC1.

```

ISPSTART CMD(%ACBQBAC1 +
BASELIST(BASE.TESTLIST) +
NEWLIST(NEW.TESTLIST) +
TESTBED(TESTCASE.LIBRARY) +
RSLTDSN(COMPARE.LISTING) +
XCPTPDS(TESTCASE.EXCP) +
XCPSPACE(5,3,30)) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*****
//*
//* COPY THE COMPARISON LISTINGS DATA SET
//*
//*   SYSUT1 - INPUT (FROM PREVIOUS STEP)
//*   SYSUT2 - OUTPUT
//*
//*****
//REPGEN   EXEC PGM=IEBGENER
//SYSUT1   DD DISP=SHR,DSN=IBMUSER.COMPARE.LISTING
//SYSUT2   DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSPRINT DD SYSOUT=*
//

```

Figure 39 (Part 2 of 2). Sample JCL for ACBJBAC1.

Enhanced ACS Test Listing: ACBQBAX1

ACBQBAX1 is called by SYS1.SACBCNTL member ACBJBAX1 to generate a detailed cross-reference listing report from the original ACS listing. The cross-reference listing report is used to help determine where there are logic errors in an ACS routine. The report can include the data set name, unit type, data set size, expiration date, job name, and program name for each exception test case.

See Figure 40 for the sample JCL and parameters.

Use the following parameters with the ACBQBAX1:

Parameters	Description
DSNINFO(Y N)	Includes the data set name in the cross-reference
UNITINFO(Y N)	Includes the unit name in the cross-reference
SIZEINFO(Y N)	Includes size information (in K) in the cross-reference
EXPTINFO(Y N)	Includes expiration date information in the cross-reference
JOBINFO(Y N)	Includes jobname information in the cross-reference
PGMINFO(Y N)	Includes program name information in the cross-reference

Sample JCL for Generating an Enhanced ACS Test Listing

```
//*****
//* SAMPLE JCL TO GENERATE ENHANCED ACS TEST LISTING IN BATCH      *
//*****
//*****
//*                                                                    *
//* STEP TO GENERATE ENHANCED ACS TEST LISTING                      *
//*                                                                    *
//* PARAMETERS:                                                     *
//*                                                                    *
//*     ACSDSN   - ACS TEST LISTING (INPUT)                          *
//*     XREFDSN  - ENHANCED ACS TEST LISTING (OUTPUT)                *
//*     OUTPUT SELECTION CRITERIA: (DATA COMES FROM TEST CASE PDS) *
//*     DSNINFO  - DSN      - DATA SET NAME                        *
//*     UNITINFO - UNIT    - UNIT ON WHICH THE DATA SET RESIDES   *
//*     SIZEINFO - SIZE    - SIZE OF DATA SET                      *
//*     EXPTINFO - EXPDT   - EXPIRY DATE OF THE DATA SET          *
//*     JOBINFO  - JOB     - JOB WHICH ALLOCATED THE DATA SET     *
//*     PGMINFO  - PGM    - PROGRAM WHICH ALLOCATED THE DATA SET *
//*                                                                    *
//*****
```

Figure 40 (Part 1 of 2). Sample JCL for ACBJBAX1.

```

//GENREP EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
DEL RSLT1
%ACBQBAX1 +
ACSDSN(BASE.LISTING) +
XREFDSN(RSLT1) +
DSNINFO(Y) +
UNITINFO(Y) +
SIZEINFO(Y) +
EXPTINFO(Y) +
JOBINFO(Y) +
PGMINFO(Y)
/*
//*****
//*
//* COPY ENHANCED ACS TEST LISTING
//*
//*   SYSUT1 - INPUT (FROM PREVIOUS STEP)
//*   SYSUT2 - OUTPUT
//*
//*****
//REPGEN EXEC PGM=IEBGENER
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=IBMUSER.ENHANCED.LISTING
//SYSUT2 DD SYSOUT=*
//*

```

Figure 40 (Part 2 of 2). Sample JCL for ACBJBAX1.

Update Test Cases with Expected Results: ACBQBAU1

ACBQBAU1 is called by SYS1.SACBCNTL member ACBJBAU2 to update the testbed library with the expected results for the new data set type that has been placed in the *new* ACS listing.

See Figure 41 for the sample JCL and parameters.

Note: ACBQBAU1 is I/O intensive, as it must update each test case member in the PDS with the new expected results.

Sample JCL to Update Test Cases

```
//*****  
//*                                                                 *  
//* SAMPLE JCL TO UPDATE TEST CASES WITH TEST RESULTS IN BATCH   *  
//*                                                                 *  
//*   INSTRUCTIONS BEFORE SUBMITTING:                             *  
//*                                                                 *  
//*       CHANGE JOBCARD                                          *  
//*       CHANGE PREFIX                                           *  
//*       CHANGE PARAMETERS                                       *  
//*                                                                 *  
//*   PARAMETERS:                                                *  
//*                                                                 *  
//*       NEWLIST  - NEW ACS TEST LISTING                          *  
//*       TESTBED  - TEST CASE PDS                                 *  
//*       XCPTDSN  - EXCEPTION TEST CASE PDS                       *  
//*       RSLTDSN  - COMPARISON RESULTS DATA SET                  *  
//*                                                                 *  
//*****  
//UPDTE EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
ISPSTART CMD(%ACBQBAU1 +  
NEWLIST(NEW.TESTLIST) +  
XCPTPDS(TESTCASE.EXCP) +  
TESTBED(TESTCASE.LIBRARY) +  
RSLTDSN(COMPARE.LISTING)) +  
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)  
/*
```

Figure 41. Sample JCL for ACBJBAU2.

Generate Report from ISMF-saved Data Set List: ACBQBAR1

ACBQBAR1 EXEC is called by SYS1.SACBCNTL member ACBJBAOD to generate a flat file from an ISMF-saved data set table, listing the fields of your choice, in the order you specify.

See Figure 42 on page 108 for the sample JCL and parameters.

Use the following parameters on the SYSIN DDNAME statement when you run the ACBQBAR1 EXEC. Each parameter must be on a separate line.

Parameters	Description
%NOTUSED	Prints the amount of space <i>not</i> used for the data set.
ALLOCS ALLOCSPC	Prints the data set's allocated space.
ALLOCUSED	Prints the amount of used space for the data set.
BACKUP LASTBKUP	Prints the date of the last backup for the data set.
BLKSIZE CISIZE	Prints the BLKSIZE or CISIZE of the data set.
BLKUNUSED	Prints the number of unused blocks for the data set.
CHANGE CHGIND	Is the change indicator set for the data set?
CREATE CREATEDT	Prints the data set's create date.
DATACLAS DC	Prints the data class for the data set, if there is one.
DEVTYPE DEVICE	Prints the device type that the data set resides on.
DSNAME DSN	Prints the data set name.
DSNLENGTH=nn	Limits print of data set name to nn characters. Defaults to 44 characters.
DSORG	Prints the data set's organization.
ENTRYTYPE ENTRY	Prints the entry type of data set.
ENVIRONMENT ENVIR	Prints the data set's environment (managed or unmanaged).
EXPIRE EXPIREDT	Prints the data set's expiration date.
EXTNUM	Prints the data set's extension number.
LASTREF LASTREFDT	Prints the last reference date of the data set.
LRECL	Prints the LRECL of the data set.
MGMTCLAS MC	Prints the management class for the data set, if there is one.
MULTVOL	Is the data set a part of a multivolume data set?
NUMEXT EXTNUM	Prints the number of extents for the data set.
OPTIMAL	Prints the optimal block size of the data set.
OWNER	Prints the owner of the data set, if there is one.
PAGELength=nn	Sets page length for the report (default is 60).
PDSE	Is the data set a PDSE?
REBLOCK REBLK	Is the data set reblockable?
RECFM RECFMT	Prints the record format of the data set.

Parameters	Description
SECALLOC ALLOCSEC	Prints the secondary allocation amount for the data set.
STORCLAS SC	Prints the storage class for the data set, if there is one.
TITLE=xxx	Prints a title for the report. Title does <i>not</i> need to be placed in parentheses or quotation marks, and cannot expand more than one record in length.
TOTALS	Prints totals for the data set space allocations. If this parameter is not specified, totals does not print.
UNIT ALLOCUNIT	Prints the allocation unit (TRKS, CYLS, and for forth) for the data set.
VOLSER	Prints the data set's volume serial number.

Sample JCL for Generating a Report from ISMF-saved Data Set List

```

//*****
//*
//* SAMPLE JCL TO GENERATE DATA SET REPORT FROM A PREVIOUSLY SAVED *
//* ISMF DATA SET LIST *
//* *
//* INSTRUCTIONS BEFORE SUBMITTING: *
//* *
//* CHANGE JOBCARD *
//* CHANGE PREFIX *
//* CHANGE PARAMETERS *
//* *
//*****
//*****
//* *
//* DELETE STEP TO DELETE THE REPORT DATA SET IF IT EXISTS ALREADY *
//* *
//*****
//DELREP EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE IBMUSER.DATASET.REPORT
/*
//*****
//* *
//* DATA SET REPORT GENERATION STEP *
//* *
//* PARAMETER FOLLOWING ACBQBAR1 - SAVED ISMF LIST (INPUT) *
//* ISPFIL - DATA SET REPORT (OUTPUT) *
//* SYSIN - KEY WORDS SPECIFYING DATA IN THE REPORT *
//* *
//*****
//GENREP EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//ISPFIL DD DSN=IBMUSER.DATASET.REPORT,DISP=(NEW,CATLG),
// BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA
//SYSTSIN DD *

```

Figure 42 (Part 1 of 2). Sample JCL for ACBJBAOD.

```

PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAR1 DSNLIST) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//SYSIN DD *
TITLE=LIST OF DATASETS FOR HIGHLEVEL QUALIFIER IBMUSER
DSN
VOLSER
LASTREF
EXTNUM
TOTALS
/*
//*****
//*
//* COPY THE DATA SET REPORT
//*
//*   SYSUT1 - INPUT (FROM PREVIOUS STEP)
//*   SYSUT2 - OUTPUT
//*
//*****
//STEP2 EXEC PGM=IEBGENER
//SYSUT1 DD DSN=IBMUSER.DATASET.REPORT,DISP=SHR
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//*

```

Figure 42 (Part 2 of 2). Sample JCL for ACBJBAOD.

Generate Report from ISMF-saved DASD Volume List: ACBQVAR1

ACBQVAR1 is called by SYS1.SACBCNTL member ACBJBAOF to generate a flat file from an ISMF-saved DASD volume table and lists the fields of your choice, in the order you specify.

See Figure 43 on page 111 for the sample JCL and parameters.

Use the following parameters on the SYSIN DDNAME statement when you run the ACBQVAR1 EXEC in batch. Each parameter must be on a separate line.

Parameters	Description
%FREE	Prints the % of free space on the volume.
ALLOCSPC ALLOCSP	Prints the amount of allocated space on the volume.
CACHEFW	Does the volume have Cache Fast Write enabled for it?
DASDFW	Does the volume have DASD FAST WRITE enabled for it?
DEVICE DEVICETYPE	Prints the device type of the volume.
DEVNUM ADDRESS	Prints the device number of the volume.
DUPLEX DUPLEXST	Is the volume part of a dual-copy pair?
FRAG FRAGINDX	Prints the fragmentation index of the volume.
FREEDSCB DSCBFREE	Prints the number of free data set control blocks in the VTOC of the volume.
FREESPC FREESP	Prints the volumes free space in KB.
FREEVIRS VIRSFREE	Prints the number of free VIRs in the indexed VTOC on the volume.
FREEEXT EXTFREE	Prints the number of free extents on the volume.
INDXSTAT STATINDX	Prints the status of the indexed VTOC on the volume.
LRGEXT EXTLRG	Prints the size of the largest extent available on the volume.
OTHER OTHERDEV	If duplex, what is the secondary volume?
PAGELength=nn	Sets page length for the report (default is 60).
PHYSTAT STATPHYS	Prints the physical status of the volume.
RDCACHE RDSTAT	Does the volume have READ CACHE enabled for it?
SHARE SHRDASD	Is the DASD volume genned as SHARED?
STORGRP SG	Prints the storage group that the volume belongs to.
SUBSYS SUBSYSID	Prints the controller's subsystem identifier.
TITLE=xxx	Prints a title for the report. Title does <i>not</i> need to be placed in parentheses or quotation marks, and cannot expand more than one record in length.
TOTALS	Prints totals for the volume. If this parameter is not specified, totals does not print.
USE USEATTR	Prints the volumes use attribute (public and so on).
VOLSER	Prints the volume serial number.

Sample JCL for Generating a Report from ISMF-saved DASD Volume List:

```
//*****  
//* *  
//* SAMPLE JCL TO GENERATE DASD VOLUME REPORT FROM A PREVIOUSLY *  
//* SAVED ISMF DASD VOLUME LIST *  
//* *  
//* INSTRUCTIONS BEFORE SUBMITTING: *  
//* *  
//* CHANGE JOBCARD *  
//* CHANGE PREFIX *  
//* CHANGE PARAMETERS *  
//* *  
//*****  
//*****  
//* *  
//* DELETE STEP TO DELETE THE REPORT IF IT EXISTS ALREADY *  
//* *  
//*****  
//DELREP EXEC PGM=IDCAMS  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
DELETE IBMUSER.DASDVOL.REPORT  
/*  
//*****  
//* *  
//* DASD VOLUME REPORT GENERATION STEP *  
//* *  
//* PARAMETER FOLLOWING ACBQVAR1 - ISMF SAVED LIST (INPUT) *  
//* ISPFIL - DASD VOLUME REPORT (OUTPUT) *  
//* SYSIN - KEY WORDS TO SPECIFY THE DATA IN THE REPORT *  
//* *  
//*****  
//GENREP EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL  
//ISPFIL DD DSN=IBMUSER.DASDVOL.REPORT,DISP=(NEW,CATLG),  
// BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
ISPSTART CMD(ACBQVAR1 DASDLIST) +  
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)  
/*  
//SYSIN DD *  
VOLSER  
INDXSTAT  
FRAG  
USEATTR  
SG  
TITLE=STATUS OF VOLUMES THAT ARE SMS MANAGED  
/*
```

Figure 43 (Part 1 of 2). Sample JCL for ACBJBAOF.

```

//*****
//*
//* COPY THE DASD VOLUME REPORT
//*
//*      SYSUT1 - INPUT (FROM PREVIOUS STEP)
//*      SYSUT2 - OUTPUT
//*
//*****
//REPGEN EXEC PGM=IEBGENER
//SYSUT1 DD DSN=IBMUSER.DASDVOL.REPORT,DISP=SHR
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//*
```

Figure 43 (Part 2 of 2). Sample JCL for ACBJBAOF.

Generate Report from ISMF-Saved Tape List: ACBQBAR4

ACBQBAR4 is called by SYS1.SACBCNTL member ACBJBAOT to generate a flat file from an ISMF-saved tape table and lists the fields of your choice, in the order you specify.

See Figure 44 on page 114 for the sample JCL and parameters.

Use the following parameters on the SYSIN DDNAME statement when you run the ACBQBAR4 EXEC in batch. Each parameter must be on a separate line.

Parameters	Description
CKPTVOL CKPT	Prints whether this is a checkpoint volume.
COMPTYPE TYPECOMP	Prints the compaction type.
CRTDTVOL VOLCRTDT	Prints the volume's creation date.
EXPDTVOL VOLEXPDT	Prints the volume's expiration date.
LSTENTDT LASTEJECT	Prints the last eject/entry date.
LSTMNTDT LASTMOUNT	Prints the volume's last mount date.
LSTWRTDT LASTWRITE	Prints the volume's last write date.
LIBNAME	Prints the volume's library name.
MEDIA MEDIATYPE	Prints the volume's media type.
OWNER OWNERINFO	Prints the volume's owner information.
PAGELength=nn	Sets the page length for the report (default is 60)
RECTECH TECHREC	Prints the volume's recording technology.
SHELF SHELFLOC	Prints the volume's shelf location.
SPCLATTR ATTRSPCL	Prints the volume's special attributes.
STORGRP SGNAME	Prints the storage group that the tape belongs to.
TITLE=nn	Prints a title for the report. Title does <i>not</i> need to be placed in parentheses or quotation marks and cannot expand more than one record in length.
USEATTR ATTRUSE	Prints the volume use attributes.
VOLERROR ERRORVOL	Prints the volume's error information.
VOLLOC LOCVOL	Prints the volume's location.
VOLSER	Prints the volume serial number.
WRTPROT PROTWRT	Is the volume write protected?

Sample JCL for Generating a Report from ISMF-saved Tape List

```
//*****  
//*  
//* SAMPLE JCL TO GENERATE TAPE VOLUME REPORT FROM A PREVIOUSLY *  
//* SAVED ISMF MOUNTABLE TAPE VOLUME LIST *  
//* *  
//* INSTRUCTIONS BEFORE SUBMITTING: *  
//* *  
//* CHANGE JOBCARD *  
//* CHANGE PREFIX *  
//* CHANGE PARAMETERS *  
//* *  
//*****  
//*****  
//* *  
//* DELETE STEP, TO DELETE THE REPORT DATA SET IF IT EXISTS ALREADY *  
//* *  
//*****  
//DELREP EXEC PGM=IDCAMS  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
DELETE IBMUSER.TAPEVOL.REPORT  
/*  
//*****  
//* *  
//* REPORT GENERATION STEP *  
//* *  
//* PARAMETER FOLLOWING ACBQBAR4 - ISMF SAVED LIST NAME (INPUT) *  
//* ISPFIL - TAPE VOLUME REPORT (OUTPUT) *  
//* SYSIN - KEYWORDS SPECIFYING THE DATA IN THE REPORT *  
//* *  
//*****  
//GENREP EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL  
//ISPFIL DD DSN=IBMUSER.TAPEVOL.REPORT,DISP=(NEW,CATLG),  
// BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
ISPSTART CMD(ACBQBAR4 TAPELIST) +  
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)  
/*  
//SYSIN DD *  
VOLSER  
LASTMOUNT  
LASTEJECT  
USEATTR  
STORGRP  
LIBNAME  
TITLE=STATUS OF TAPES AS ON 06/01/96  
/*
```

Figure 44 (Part 1 of 2). Sample JCL for ACBJBAOT.


```

//*****
//*
//* COPY THE TAPE VOLUME REPORT
//*
//*   SYSUT1 - INPUT (FROM PREVIOUS STEP)
//*   SYSUT2 - OUTPUT
//*
//*****
//REPGEN EXEC PGM=IEBGENER
//SYSUT1 DD DSN=IBMUSER.TAPEVOL.REPORT,DISP=SHR
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//*****

```

Figure 44 (Part 2 of 2). Sample JCL for ACBJBAOT.

Generate Data Set Report from DCOLLECT Data: ACBQBAR7

ACBQBAR7 is called by SYS1.SACBCNTL member ACBJBARD to generate a flat file from DCOLLECT data taken from data set records and lists the fields of your choice, in the order you specify.

See Figure 45 on page 117 for the sample JCL and parameters.

Use the following parameters on the SYSIN DDNAME statement when you run the ACBQBAR7 EXEC in batch. Each parameter must be on a separate line.

Parameters	Description
ALLOCSP ALLOCSPC	Prints the data sets allocated space.
BACKUP BACKUPDT	Prints the date of the last backup for data set.
BLKUNUSED	Prints the number of unused blocks for the data set.
CHANGE CHGIND	Is the change indicator on for data set?
CREATE CREATEDT	Prints the data set's creation date.
DATACLAS DC	Prints the data class for the data set, if there is one.
DSNAME DSN	Prints the data set name.
DSNLENGTH=nn	Limits print of data set name to nn characters. Defaults to 44 characters.
DSORG	Prints the DSORG for data set.
ENTRYTYPE	Prints the data set's entry type.
EXPIRE EXPIREDT	Prints the data set's expiration date.
LASTREF LASTREFDT	Prints the data set's last reference date.
LRECL	Prints the data set's record length.
MGMTCLS	Prints the management class for the data set, if there is one.
MULTVOL	Is the data set multivolume?
NUMEXT EXTNUM	Prints the number of extents for data set.
PAGELength=nn	Sets the page length for the report (default is 60).
PDSE	Is the data set a PDSE?
REBLOCK REBLK	Is the data set reblockable?
RECFM RECFMT	Prints the data set's record format.
SMS MANAGED	Is the data set SMS-managed?
STORCLAS SC	Prints the storage class for data set, if there is one.
STORGRP SG	Prints the storage group for data set, if there is one.
TITLE=xxx	Prints a title for the report. Title does <i>not</i> need to be placed in parentheses or quotation marks, and cannot expand more than one record in length.
TOTALS	Specifies whether you want DSN space totals printed for this DCOLLECT data.
USED%	Prints the percentage of used space for the data set.
VOLSEQ	Prints the volume sequence number for data set.

Parameters	Description
VOLSER	Prints the volume serial of the data set.
VVRCHK VVR	If the data set is SMS-managed, does it have a valid VVR or NVR?

Sample JCL for Generating Data Set Report from DCOLLECT Data:

```

//*****
//*
//* SAMPLE JCL TO GENERATE DATA SET REPORT FROM DCOLLECT DATA
//*
//* INSTRUCTIONS BEFORE SUBMITTING:
//*
//* CHANGE JOBCARD
//* CHANGE PREFIX
//* CHANGE PARAMETERS
//*
//*****
//*****
//*
//* DELETE STEP - TO DELETE THE REPORT DATA SET IF IT EXISTS ALREADY
//*
//*****
//DELREP EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE IBMUSER.DATASET.REPORT
/*
//*****
//*
//* REPORT GENERATION STEP
//*
//* DCOLIN - DCOLLECT DATA (INPUT)
//* ISPFIL - DATA SET REPORT (OUTPUT)
//* SYSIN - KEY WORDS TO SPECIFY THE DATA IN THE OUTPUT
//*
//* CHOOSE FROM THE FOLLOWING PARAMETERS FOR DATA TO BE LISTED
//* THE ORDER CHOSEN DETERMINES THE ORDER IN THE REPORT
//* THE TOTAL REPORT WIDTH MAY NOT EXCEED 133
//*
//* PARAMETERS:
//*

```

Figure 45 (Part 1 of 3). Sample JCL for ACBJBARD.

```

/**      ALLOCSP | ALLOCSPC - ALLOCATED SPACE *
/**      BACKUP | LASTBKUP - DATA SET'S LAST BACKUP DATE *
/**      BLKUNUSED - BLOCKS UNUSED *
/**      CHANGE | CHGIND - CHANGE INDICATOR FOR DATA SET *
/**      CREATE | CREATEDT - DATA SET CREATION DATE *
/**      DATACLAS | DC - DATACLAS OF DATA SET *
/**      DSNNAME | DSN - DATASETNAME *
/**      DSNLENGTH - LIMIT DATASET NAME TO THIS NUMBER *
/**      - OF CHARACTERS *
/**      DSORG - DATA SET ORGANIZATION *
/**      ENTRYTYPE - ENTRY TYPE OF DATA SET *
/**      EXPIRE | EXPIREDT - DATA SET EXPIRATION DATE *
/**      LASTREF | LASTREFDT - DATA SET LAST REFERENCE DATE *
/**      LRECL - RECORD LENGTH *
/**      MGMTCLAS | MC - MANAGEMENT CLASS FOR DATA SET *
/**      MULTVOL - IS THE DATA SET MULTI-VOLUME? *
/**      NUMEXT | EXTNUM - NUMBER OF EXTENTS *
/**      PAGELENGTH - NUMBER OF LINES PER PAGE (DEF IS 60) *
/**      PDSE - IS THE DATA SET A PDSE? *
/**      REBLOCK | REBLK - IS THE DATA SET REBLOCKABLE? *
/**      RECFM | RECFMT - RECORD FORMAT *
/**      SMS | MANAGED - IS THE DATA SET MANAGED? *
/**      STORCLAS | SC - STORAGE CLASS FOR DATA SET *
/**      STORGRP | SG - STORAGE GROUP FOR DATA SET *
/**      TITLE=XXXXX - TITLE FOR REPORT *
/**      TOTALS - PRINT DSN SPACE TOTALS *
/**      USED% - USED SPACE % *
/**      VOLSEQ - VOLUME SEQUENCE NUMBER *
/**      VOLSER - VOLUME SERIAL *
/**      VVRCHK | VVR - IF SMS, IS THERE A VVR OR NVR? *
/** *
/*******
//GENREP EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//DCOLIN DD DSN=IBMUSER.DCOLLECT.DATA,DISP=SHR
//ISPFIL DD DSN=IBMUSER.DATASET.REPORT,DISP=(NEW,CATLG),
// BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAR7) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//SYSIN DD *
DSN
VOLSER
ALLOCSP
RECFM
BLKUNUSED
TITLE=DATA SET REPORT FROM DCOLLECT DATA - 06/01/96
TOTALS
BLKSIZE
EXTNUM
DSORG
STORCLAS
/*

```

Figure 45 (Part 2 of 3). Sample JCL for ACBJBARD.

```

//*****
//*
//* COPY THE DATA SET REPORT
//*
//* SYSUT1 - INPUT (FROM PREVIOUS STEP)
//* SYSUT2 - OUTPUT
//*
//*****
//REPGEN EXEC PGM=IEBGENER
//SYSUT1 DD DSN=IBMUSER.DATASET.REPORT,DISP=SHR
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//*****

```

Figure 45 (Part 3 of 3). Sample JCL for ACBJBARD.

Generate DASD Volume Report from DCOLLECT Data: ACBQBAR6

ACBQBAR6 is called by SYS1.SACBCNTL member ACBJBAXV to generate a flat file from DCOLLECT data from DASD volume records, and lists the fields of your choice, in the order you specify.

See Figure 46 on page 121 for the sample JCL and parameters.

Use the following parameters on the SYSIN DDNAME statement when you run the ACBQBAR6 EXEC in batch. Each parameter must be on a separate line.

Parameters	Description
%FREE FREE%	Prints the percentage of free space on the volume.
DEVCAP CAPACITY	Prints the devices capacity.
DEVNUM ADDRESS	Prints the device number of the volume.
DEVTYPE	Prints the device type of the volume.
FRAGINDX FRAG	Prints the volumes fragmentation index.
FREEDSCB DSCBFREE	Prints the number of free DSCBs in the VTOC.
FREEKB KBFREE	Prints the number of bytes (in KB) on the volume.
FREEVIR VIRFREE	Prints the number of free VIRs in the indexed VTOC on the volume.
FREEEXT EXTFREE	Prints the number of free extents on the volume.
INDEX INDSTAT	Prints the indexed VTOC status.
LRGEXT EXTLRG	Prints the volume's largest available extent, in number of bytes.
PAGELENGTH=nn	Sets page length for report (default is 60).
SHRDASD SHARED	Prints whether the volume is shared.
SPACALLOC ALLOCSPC	Prints the amount of allocated space on the volume (in KB).
STORGRP	Prints the storage group that the volume belongs to.
TITLE=xxx	Prints a title for the report. Title does <i>not</i> need to be placed in parentheses or quotation marks, and cannot expand more than one record in length.
USEATTR ATTRIB	Prints the use attribute (PUBLIC, STORAGE, and so on) of the volume.
VOLSER VOL	Prints the volume serial.
VOLSTAT SMSSTAT	Prints the SMS status (initial, convert, and so on) of the volume.

Sample JCL for Generating a DASD Volume Report from DCOLLECT Data:

```

//*****
//*
//* SAMPLE JCL TO GENERATE DASD VOLUME REPORT FROM DCOLLECT DATA *
//*
//* INSTRUCTIONS BEFORE SUBMITTING: *
//*
//* CHANGE JOBCARD *
//* CHANGE PREFIX *
//*
//* CHOOSE FROM THE FOLLOWING PARAMETERS FOR DATA TO BE LISTED. *
//* THE ORDER CHOSEN DETERMINES THE ORDER IN THE REPORT. *
//* THE TOTAL REPORT WIDTH MAY NOT EXCEED 133. *
//*
//* %FREE | FREE% - % FREESPACE ON VOLUME. *
//* DEVCAP | CAPACITY - DEVICE CAPACITY *
//* DEVNUM | ADDRESS - DEVICE NUMBER OF VOLUME. *
//* DEVTYPE - DEVICE TYPE OF VOLUME. *
//* FRAGINDX | FRAG - FRAGMENTATION INDEX OF VOLUME. *
//* FREEDSCB | DSCBFREE - # OF FREE DSCBS IN VTOC. *
//* FREEVIR | VIRFREE - # OF FREE VIRS IN INDEXED VTOC. *
//* FREEEXT | EXTFREE - NUMBER OF FREE EXTENTS ON VOLUME. *
//* INDEX | INDXSTAT - STATUS OF INDEXED VTOC ON VOLUME. *
//* KBFREE | FREEKB - FREESPACE IN KB BYTES. *
//* LRGEXT | EXTLRG - SIZE OF LARGEST EXTENT ON VOLUME. *
//* PAGELENGTH - NUMBER OF LINES PER PAGE (DEF IS 60)*
//* - OF CHARACTERS. *
//* SHRDASD|SHARED - IS THE VOLUME A SHARED DASD VOLUME? *
//* SPACALLOC|ALLOCSPC - ALLOCATED SPACE ON VOLUME. *
//* STORGRP - VOLUMES STORAGE GROUP. *
//* TITLE=XXX - TITLE FOR REPORT *
//* USEATTR | ATTRIB - VOLUME USE ATTRIBUTE (PUBLIC,ETC.) *
//* VOLSER | VOL - VOLUME SERIAL *
//* VOLSTAT | SMSSTAT - VOLUMES SMS STATUS (CONVERT, ETC.) *
//*
//*****
//*****
//*
//* DELETE STEP TO DELETE THE REPORT DATA SET IF EXISTS ALREADY *
//*
//*****
//DELREP EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE IBMUSER.DASDVOL.REPORT
/*

```

Figure 46 (Part 1 of 2). Sample JCL for ACBJBAXV.

```

//*****
//*
//* REPORT GENERATION STEP
//*
//*   DCOLIN - DCOLLECT DATA (INPUT)
//*   ISPFIL - DASD VOLUME REPORT (OUTPUT)
//*   SYSIN  - KEYWORDS SPECIFYING THE INFO IN THE REPORT
//*
//*****
//GENREP EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//DCOLIN DD DSN=IBMUSER.DCOLLECT.DATA,DISP=SHR
//ISPFIL DD DSN=IBMUSER.DASDVOL.REPORT,DISP=(NEW,CATLG),
//   BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAR6) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//SYSIN DD *
VOLSER
FRAG
DEVTYPE
TITLE=DASD VOLUME REPORT FROM DCOLLECT FOR MVS2 SYSTEM 06/01/96
ADDRESS
INDEX
FREE%
FREEDSCB
STORGRP
/*
//*****
//*
//* COPY DASD VOLUME REPORT
//*
//*   SYSUT1 - INPUT (FROM PREVIOUS STEP)
//*   SYSUT2 - OUTPUT
//*
//*****
//REPGEN EXEC PGM=IEBGENER
//SYSUT1 DD DSN=IBMUSER.DASDVOL.REPORT,DISP=(SHR,DELETE)
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//*****

```

Figure 46 (Part 2 of 2). Sample JCL for ACBJBAXV.

Generate SMS Configuration Reports from DCOLLECT Data: ACBQBAR8

ACBQBAR8 is called by SYS1.SACBCNTL member ACBJBAR2 to generate a flat file from DCOLLECT data and lists the records that you select, from DCOLLECT SMS configuration records.

See Figure 47 for the sample JCL and parameters.

Use the following parameters on the SYSIN DD statement ACBQBAR8:

Parameters	Description
RECORD=AG	Formats aggregate group record information records.
RECORD=AI	Prints accounting information from the ACS.
RECORD=BC	Formats base configuration record information records.
RECORD=CN	Prints cache names from the base configuration.
RECORD=DC	Formats data class configuration records.
RECORD=DR	Formats OAM drive record information records.
RECORD=LB	Formats OAM library record information records.
RECORD=MC	Formats management class configuration records.
RECORD=SC	Formats storage class configuration records.
RECORD=SG	Formats storage group configuration records.
RECORD=VL	Formats storage group volume record information records.

Sample JCL for Generating SMS Configuration Reports from DCOLLECT Data

```

//*****
//*
//* SAMPLE JCL TO GENERATE SMS CONFIGURATION REPORT FROM
//* DCOLLECT DATA
//*
//* INSTRUCTIONS BEFORE SUBMITTING:
//*   CHANGE JOBCARD
//*   CHANGE PREFIX
//*   CHANGE PARAMETERS
//*
//*****
//*****
//*
//* DELETE STEP - TODELETE THE REPORT DATA SET IF IT EXISTS ALREADY
//*
//*****
//DELRPT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *

```

Figure 47 (Part 1 of 2). Sample JCL for ACBJBAR2.

```

DELETE IBMUSER.SMSCONF.REPORT
/*
//*****
//*
//*   REPORT GENERATION STEP
//*
//*   DCOLIN - DCOLLECT DATA (INPUT)
//*   ISPFIL - SMS CONFIGURATION REPORT (OUTPUT)
//*   SYSIN   - PARAMETERS THAT SPECIFY THE OUTPUTS NEEDED
//*
//*   PARAMETERS:
//*
//*   RECORD=DC - DATA CLASSES
//*   RECORD=SC - STORAGE CLASSES
//*   RECORD=MC - MANAGEMENT CLASSES
//*   RECORD=SG - STORAGE GROUPS
//*   RECORD=BC - BASE CONFIGURATION
//*   RECORD=VL - STORAGE GROUP VOLUMES
//*   RECORD=AG - AGGREGATE GROUPS
//*   RECORD=LB - OAM LIBRARIES
//*   RECORD=DR - OAM DRIVES
//*   RECORD=AI - ACS ACCOUNTING
//*
//*****
//REPGEN EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//DCOLIN DD DSN=IBMUSER.DCOLLECT.DATA,DISP=SHR
//ISPFIL DD DSN=IBMUSER.SMSCONF.REPORT,DISP=(NEW,CATLG),
//          BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAR8) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//SYSIN DD *
RECORD=DC
RECORD=MC
RECORD=SC
RECORD=SG
/*
//*****
//*
//*   COPY SMS CONFIGURATION REPORT
//*
//*   SYSUT1 - INPUT (FROM PREVIOUS STEP)
//*   SYSUT2 - OUTPUT
//*
//*****
//REPGEN EXEC PGM=IEBGENER
//SYSUT1 DD DISP=SHR,DSN=IBMUSER.SMSCONF.REPORT
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//*****

```

Figure 47 (Part 2 of 2). Sample JCL for ACBJBAR2.

Generate Model Commands from ISMF–saved List: ACBQBAM1

SYS1.SACBCNTL member ACBJBAM1 calls ACBQBAM1 to create model commands from a saved ISMF list.

See Figure 48 for the sample JCL and parameters.

Sample JCL for Generating Model Commands from ISMF–saved List

```
//*****  
//*  
//* SAMPLE JCL TO GENERATE MODEL COMMANDS FROM A SAVED ISMF DATA *  
//* SET LIST IN BATCH *  
//* *  
//* INSTRUCTIONS BEFORE SUBMITTING: *  
//* *  
//* CHANGE JOBCARD *  
//* CHANGE PREFIX *  
//* CHANGE PARAMETERS *  
//* *  
//* PARAMETERS: *  
//* *  
//* PARAMETER FOLLOWING ACBQBAM1 - SAVED ISMF LIST NAME *  
//* DATA SET OR DASD VOLUME LIST *  
//* REST OF THE PARAMETERS - MODEL COMMAND FOR GENERATION *  
//* *  
//*****  
//GENCMDS EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
ISPSTART CMD(%ACBQBAM1 DSNLIST +  
DD DSN=/,VOL=SER=@ +  
) +  
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)  
/*  
//*
```

Figure 48. Sample JCL for ACBJBAM1.

Generate Model Commands from DCOLLECT Data: ACBQBAM2

SYS1.SACBCNTL member ACBJBAM2 calls ACBQBAM2 to create model commands from DCOLLECT data.

See Figure 49 for the sample JCL and parameters.

Sample JCL for Generating Model Commands from DCOLLECT Data

```
//*****  
//*                                                                 *  
//* SAMPLE JCL TO GENERATE MODEL COMMANDS FROM DCOLLECT DATA   *  
//*                                                                 *  
//* INSTRUCTIONS BEFORE SUBMITTING:                               *  
//*                                                                 *  
//*   CHANGE JOBCARD                                             *  
//*   CHANGE PREFIX                                             *  
//*   CHANGE PARAMETERS                                         *  
//*   CHANGE DCOLIN DD                                          *  
//*                                                                 *  
//* PARAMETERS:                                                 *  
//*                                                                 *  
//*   DCOLIN - DCOLLECT DATA (INPUT)                           *  
//*   PARAMETERS FOLLOWING ACBQBAM2 - MODEL COMMAND FOR GENERATION *  
//*                                                                 *  
//*****  
//STEP1 EXEC ACBJBAOB  
//DCOLIN DD DSN=IBMUSER.DCOLLECT.DATA,DISP=SHR  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
ISPSTART CMD(%ACBQBAM2 DSN=/,VOL=SER=@) +  
BDISPMAX(999999999) BREDIMAX(3) BATSCRW(132) BATSCRD(27)  
/*  
//*
```

Figure 49. Sample JCL for ACBJBAM2.

Chapter 4. Creating DFSMS Online Reports

With NaviQuest you can create a report of the contents of an SMS configuration, or the planned changes to a configuration, in the form of a sequential data set. The report can be printed using your choice of tool or utility.

The following reports can be generated:

- Customized data set reports from ISMF saved data set list
- Customized volume reports from ISMF saved DASD volume list
- Customized volume reports from ISMF saved tape volume list
- Customized data set reports from DCOLLECT data
- Customized volume reports from DCOLLECT data
- SMS configuration reports from DCOLLECT data

Note: These reports can also be generated in batch. Chapter 3, “Performing Storage Administration Tasks in Batch” on page 25, provides the CLISTS or EXECs required to run these reports in batch.

Before you can generate any of these reports, you must already have created *either* DCOLLECT data or an ISMF table. After creating your base data, you can select the fields that you want included in the report and the order you want them presented.

For example, if you want the data set, percentage used, and expired date fields printed in that particular order, you would enter a value of 1 in the data set field, a value of 2 in the percentage used field, and a value of 3 in the expired date field. You may number as many or as few fields as you need. To generate SMS reports, choose option 5 from the NaviQuest Primary Option Menu.

```
Panel Help
-----
ACBSMDP0 ENHANCED ACS MANAGEMENT - NAVIQUEST PRIMARY OPTION MENU
Enter Selection or Command ==> _____

Select one of the following options and press Enter:

  1 Test Case Generation
  2 ACS Test Listings Comparison
  3 Enhanced ACS Test Listing
  4 Test Case Update with Test Results
  5 SMS Report Generation
  6 Model Commands Generation
  7 Batch Testing/Configuration Management
  X Exit

Use HELP Command for Help; Use END Command or X to Exit.
```

Figure 50. NaviQuest V1R5 Primary Option Menu, ACBSMDP0

Note: For more information on generating saved lists from ISMF tables, refer to *DFSMS/MVS V1R5 Using ISMF*.

Creating Data Set Reports from Saved ISMF Lists

To generate data set reports, use the “Data Set Report From Saved ISMF List Entry Panel” panel (option 11.5.1). This panel requires you to set up the parameters of the data sets and to define which fields you want in your report.

1. From the “SMS Report Generation Selection Menu” panel (option 11.5), choose “Data Set Report from Saved ISMF List” (option 1).

```
Panel Help
-----
ACBSFLR1          SMS REPORT GENERATION SELECTION MENU
Enter Selection or Command ==> _____

Select one of the following options and press Enter:

  1 Data Set Report from Saved ISMF List
  2 DASD Volume Report from Saved ISMF List
  3 Tape Volume Report from Saved ISMF List
  4 Data Set Report from DCOLLECT Data (D records)
  5 DASD Volume Report from DCOLLECT Data (V records)
  6 SMS Configuration Report from DCOLLECT Data (DC, SC, MC, SG, BC,
    VL, AG, DR, LB, CN or AI records)

Use HELP Command for Help; Use END Command to Exit.
```

Figure 51. SMS Report Generation Selection Menu, ACBSFLR1

After choosing option 1, a panel appears from which you can choose from a selection of items that you want to appear in your report.

2. Fill in the top portion of the next screen (Data Set Report From Saved ISMF List Entry Panel) with the name of the data set that you want to hold the output report, whether you want to replace the existing data in the output data set, the name of the previously saved table, number of lines per page you want in the report, and whether you want totals for the allocated and used space attributes.

Note: The maximum allowed size for reports is 133 columns and for data set names is 44 bytes. If you select the data set name to appear in the report, you can code the Max Length of the DSN Print field to free some of the 44 bytes.

Coding the minimum number of 11 frees up 33 extra bytes of report columns; however, only the first 11 bytes of the data set name will be included in the report.

3. Fill in as many fields as you want in the order you want them printed for your report.

Note: Use PF7 (up) and PF8 (down) to scroll backward or forward.

For example, the filled in fields of the following sample panel indicate that the user wants a report that prints the data set name, the amount of allocated space, the amount of allocated space actually used, block size, and the optimal size.

```

Panel Help
-----
ACBDRPR5      DATA SET REPORT FROM SAVED ISMF LIST ENTRY PANEL
Command ==> _____

To generate report, specify the following information and press Enter:
Saved ISMF List . . . . . _____ (Data Set List)
Data Set to Hold Report
====>
  Replace Contents if DSN Exists . . _ (Y or N)  Page Length . . _
  Max Length of DSN Print . . . . . _          Totals . . _ (Y or N)

Specify fields in numeric order (max width of report is 133 characters):
                                     Length          More:  +
                                     Length
DATA SET NAME . . . . . 1_ (45)  SEC ALLOC . . . . . _ (9)
ALLOC SPACE . . . . . 2_ (8)    DS ORG . . . . . _ (6)
ALLOC USED . . . . . 3_ (8)    REC FMT . . . . . _ (7)
% NOT USED . . . . . _ (8)     RECORD LENGTH . . . . . _ (6)
COMPRESSED FORMAT . . . . . _ (7)  BLK SZ CI SIZE . . . . . 4_ (8)
% USER DATA REDUCT . . . . . _ (8)  OPTIMAL SIZE . . . . . 5_ (9)

Use DOWN Command to Scroll Forward; Use UP Command to Scroll Backward;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 52. Data Set Report from Saved ISMF List Entry Panel, ACBDRPR5

4. After you have filled in all the information on the first panel, press PF8 to scroll forward. Fill in the remaining required information. To submit the report for processing, press the Enter key. This creates a saved data set that can be browsed.

For information about generating data set name reports in batch, see “Generate Report from ISMF-saved Data Set List: ACBQBAR1” on page 107.

Creating Volume Reports from Saved ISMF Lists

To generate DASD volume reports, use “DASD Volume Report from Saved ISMF List.” (option 11.5.2).

1. From the “SMS Report Generation Selection Menu” panel (option 11.5), choose “DASD Volume Report from Saved ISMF List.” (option 2).
2. Fill in the data set name, table member name, page length, and totals.

Indicate whether you want to replace the specified data set if it already exists. If N is specified and the data set name already exists, an error message will be returned. If Y is specified, the data set will be deleted, a new data set with the same name will be allocated, and the report will be written to this data set.

3. Select as many fields as you want in the report, using numbers to indicate the order in which you want them printed.

For example, the filled in fields of the following sample panel indicate that the user wants a report that prints the volume serial number, the amount of free space, the fragmentation index, and the device number, in that order.

```

Panel Help
-----
ACBDRPR7   DASD VOLUME REPORT FROM SAVED ISMF LIST ENTRY PANEL
Command ==> _____

To generate report, specify the following information and press Enter:
Saved ISMF List . . . . . 1_ (7) (DASD Volume List)
Data Set to Hold Report
====> _____
Replace Contents if DSN Exists . . _ (Y or N) Page Length . . _ (9)
Totals . . _ (Y or N)

Specify fields in numeric order (max width of report is 133 characters):
More: +
Length
VOLUME SERIAL . . . . . 1_ (7)  FREE VIRS . . . . . _ (9)
FREE SPACE . . . . . 2_ (8)  DEVICE TYPE . . . . . _ (8)
% FREE . . . . . _ (6)  DEV NUMBER . . . . . 4_ (7)
ALLOC SPACE . . . . . _ (8)  SHR DASD . . . . . _ (8)
FRAG INDEX . . . . . 3_ (8)  USE ATTR . . . . . _ (8)
LARGEST EXTENT . . . . . _ (8)  RD CACHE STATUS . . . . . _ (9)

Use DOWN Command to Scroll Forward; Use UP Command to Scroll Backward;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 53. DASD Volume Report from Saved ISMF List Entry Panel, ACBDRPR7

4. After you have filled in all the information, press the Enter key. This creates a saved data set that can be browsed.

For information about generating volume reports in batch, see “Generate Report from ISMF-saved DASD Volume List: ACBQVAR1” on page 110.

Creating Customized Tape Reports from Saved ISMF Lists

To generate tape reports, use option 11.5.3:

1. From the “SMS Report Generation Selection Menu” panel (option 11.5), choose “Tape Volume Report from Saved ISMF List” (option 3).
2. Fill in the data set name, table member name, and page length.

Indicate whether you want to replace the specified data set if it already exists. If N is specified and the data set name already exists, an error message will be returned. If Y is specified, the data set will be deleted, a new data set with the same name will be allocated, and the report will be written to this data set.

3. Select as many fields as you want in the report, using numbers to indicate the order in which you want them printed.

For example, the filled in fields of the following sample panel indicate that the user wants a report that prints the volume serial number, the use attribute, the volume error status, the media type, and the shelf location.


```

Panel Help
-----
ACBDFLR3 TAPE VOLUME REPORT FROM SAVED ISMF LIST ENTRY PANEL
Command ==> _____

To generate report, specify the following information and press Enter:
Saved ISMF List . . . . . _____ (Tape Volume List)
Data Set to Hold Report
====> _____
Replace Contents if DSN Exists . . _ (Y or N) Page Length . . _

Specify fields in numeric order (max width of report is 133 characters):
More: +
Length Length
VOLUME SERIAL . . . . . 1_ (7) LAST WRITTEN . . . . . _ (11)
USE ATTR . . . . . 2_ (8) LAST MOUNT . . . . . _ (11)
VOL ERROR ST . . . . . 3_ (19) LAST ENTER/EJECT . . . . . _ (11)
CKPT VOLUME . . . . . _ (5) VOLUME EXPDT . . . . . _ (11)
LIBRARY NAME . . . . . _ (9) VOL CREATE DATE . . . . . _ (11)
STORGRP NAME . . . . . _ (9) WRITE PROTECT . . . . . _ (8)
MEDIA TYPE . . . . . 4_ (7) VOLUME LOCATION . . . . . 5_ (8)

Use DOWN Command to Scroll Forward; Use UP Command to Scroll Backward;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 54. Tape Volume Report from Saved ISMF List Entry Panel, ACBDFLR3

4. After you have filled in all the information, press the Enter key. This creates a saved data set that can be browsed.

For information about generating tape reports in batch, see “Generate Report from ISMF-Saved Tape List: ACBQBAR4” on page 113.

Data Set Report from DCOLLECT Data

To generate data set reports from DCOLLECT data, use option 11.5.4. Select the fields that are to be included in the report with a number to indicate the order they are to be printed.

The filled in fields of the following sample panel indicate that the user wants a report that prints the data set name, the amount of allocated space, the percentage used, the data set organization, and the volume serial number, in that order.

```

Panel Help
-----
ACBDRPR1 DATA SET REPORT FROM DCOLLECT DATA ENTRY PANEL
Command ==> _____

To generate report, specify the following information and press Enter:
DCOLLECT Data . . _____
Data Set to Hold Report
====> _____
Replace Contents if DSN Exists . . _ (Y or N) Page Length . . _
Max Length of DSN Print . . . . . _ Totals . . _ (Y or N)

Specify fields in numeric order (max width of report is 133 characters):
More: +
Length Length
DATA SET NAME . . . . . 1_ (45) CHG IND . . . . . _ (7)
ALLOC SPACE . . . . . 2_ (10) DS ORG . . . . . 4_ (6)
% USED . . . . . 3_ (6) REC FMT . . . . . _ (7)
BLOCK UNUSED . . . . . _ (8) RECORD LENGTH . . . . . _ (6)
BLK SZ CI SIZE . . . . . _ (8) SMS-MANAGED . . . . . _ (5)
EXPIRE DATE . . . . . _ (11) VOLUME SERIAL . . . . . 5_ (7)

Use DOWN Command to Scroll Forward; Use UP Command to Scroll Backward;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 55. Data Set Report from DCOLLECT Data Entry Panel, ACBDRPR1

Note: Use PF7 (up) to scroll backward and PF8 (down) to scroll forward. Press the Enter key to generate the report.

DASD Volume Report from DCOLLECT Data

To generate volume reports from DCOLLECT data, use option 11.5.5. Select the fields that are to be included in the report with a number to indicate the order they are to be printed.

For example, the filled in fields of the following sample panel indicate that the user wants a report that prints the volume serial number, the index status, the use attribute, and the number of free extents.

```

Panel Help
-----
ACBDRPR2   DASD VOLUME REPORT FROM DCOLLECT DATA ENTRY PANEL
Command ==> _____

To generate report, specify the following information and press Enter:
DCOLLECT Data . . . _____
Data Set to Hold Report
====> _____
Replace Contents if DSN Exists . . _ (Y or N) Page Length . . . _
Totals . . . . . _ (Y or N)

Specify fields in numeric order (max width of report is 133 characters):
More: +
Length Length
VOLUME SERIAL . . . . . 1_ (7) LARGEST EXTENT . . . . . _ (8)
INDEX STATUS . . . . . 2_ (9) FREE EXTENTS . . . . . 4_ (8)
USE ATTRIBUTE . . . . . 3_ (8) FREE DSCBS . . . . . _ (9)
SHARED DASD . . . . . _ (8) FREE VIRS . . . . . _ (8)
% FREESPACE . . . . . _ (6) DEVICE TYPE . . . . . _ (8)
FREESPACE (KB) . . . . . _ (8) DEV NUMBER . . . . . _ (7)

Use DOWN Command to Scroll Forward; Use UP Command to Scroll Backward;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 56. DASD Volume Report from DCOLLECT Data Entry Panel, ACBDRPR2

SMS Configuration Report from DCOLLECT Data

To generate SMS configuration reports from DCOLLECT data, use option 11.5.6. Select the fields that are to be included in the report with either Yes or No (Y or N).

- Y** include this record in the report
- N** do not include this record in the report

```

Panel Help
-----
ACBDRPR3 DFSMS CONFIGURATION REPORT FROM DCOLLECT DATA ENTRY PANEL
Command ==> _____

To generate report, specify the following information and press Enter:
DCOLLECT Data . . . _____
Data Set to Hold Report
====> _____
Replace Contents if DSN Exists . . _ (Y or N) Page Length . . _ __

Specify information to be included in the report: (Y or N)

Data Classes (DC) . . . . _      Storage Group Volumes (VL) . . _
Storage Classes (SC) . . _      OAM Drives (DR) . . . . . _
Management Classes (MC) . . _   OAM Libraries (LB) . . . . . _
Storage Groups (SG) . . . _     Cache Names (CN) . . . . . _
Base Configuration (BC) . . _   ACS Accounting (AI) . . . . . _
Aggregate Groups (AG) . . _

Use HELP Command for Help; Use END Command to Exit.

```

Figure 57. SMS Configuration Report from DCOLLECT Data Entry Panel, ACBDRPR3

For information about generating DCOLLECT reports in batch, see “Generate Data Set Report from DCOLLECT Data: ACBQBAR7” on page 116, 52, and “Generate SMS Configuration Reports from DCOLLECT Data: ACBQBAR8” on page 123.

Chapter 5. Additional Storage Administration Functions

The following are additional storage administration functions:

- QSAVE and QRETRIEV ISMF commands

The QSAVE and QRETRIEV ISMF commands let you save a “query” of frequently used parameters under ISMF. You can then retrieve the parameters by their query names. The QSAVE and QRETRIEV ISMF commands can be used while running the ISMF data set list or volume list options, interactively or in batch.

- Model Command Generator (ACBQFLM1 EXEC for saved ISMF lists and ACBQADM2 EXEC for DCOLLECT data)

The Model Command Generator option creates a “model” command against each entry in a data set saved ISMF list, a saved ISMF volume list, or from DCOLLECT data. NaviQuest does the symbolic substitution.

- COPYFILT macro

Using the COPYFILT macro you can create synchronized filter lists (FILTLISTs) that can be applied across all ACS routines. Create a FILTLIST member in your ACS routine source data set. Make all filter list updates there. Then call the COPYFILT macro from the command line to have the changes replicated across all of the ACS routines.

QSAVE and QRETRIEV Commands: Saving and Retrieving ISMF Selection Criteria

The QSAVE function saves all the variables of an ISMF query used for a particular run. To use the same variables in a later run, call QRETRIEV from the “Data Set Selection Entry Panel” or the “Volume Selection Entry Panel.” The QRETRIEV function retrieves all the values used in the first run and reuses them for a batch or an interactive session.

How to Use the QSAVE and QRETRIEV Commands

When a query is saved, a new entry is created in an ISPF table. ISMFQDSN is the table name for data set queries. ISMFQVOL is the table name for volume queries. If the tables do not exist when the first query is saved, the table is created into the library pointed to by DDNAME ISPTABL.

Duplicate query names within the ISMFQDSN and ISMFQVOL tables are not allowed. An error message indicating that there is a duplicate query name will be received if an attempt is made to save a query that already exists. An attempt to retrieve a query that does not exist, will receive an error message indicating that no query has been found.

To use either command, perform the following steps:

1. Create and then save a query.

The user has two options for creating a query:

- Create the query while running ISMF in batch.

Specify the QSAVE(*name*) parameter in the JCL for the batch job. Where *name* is the query name to be saved.

- Generate the QUERY online.

To do this, fill in all desired parameters for the query from either ISMF option 1 (data set list option) or option 2 (volume list option). Then return to the first page of the selection panels.

To save the query, enter the query name on the field marked either QUERY NAME TO SAVE or RETRIEVE. Go to the top of the panel and enter QSAVE. You will receive messages indicating that the query name has been successfully saved.

The following sample panel, "Data Set Selection Entry Panel," is the first panel with fields filled in.

```

Panel Defaults Utilities Scroll Help
-----
DGTDDDS1          DATA SET SELECTION ENTRY PANEL          Page 1 of 5
Command ==>>> _____

For a Data Set List, Select Source of Generated List ==>> _ (1 OR 2)

 1 Generate from a Saved List          Query Name to
   List Name . . . _____          Save or Retrieve . . . _____

 2 Generate a new list from criteria below
   Data Set Name . . . _____

Specify Source of the new list . . . _ (1 - VTOC, 2 - Catalog)
 1 Generate list from VTOC
   Volume Serial Number . . . _____ (fully or partially specified)
 2 Generate list from Catalog
   Catalog Name . . . _____
   Catalog Password . . . . . _____ (if password protected)
   Volume Serial Number . . . _____ (fully or partially specified)
   Acquire Data from Volume . . . . . _ (Y or N)
   Acquire Data if DFSMSHsm Migrated . . _ (Y or N)
Use ENTER to Perform Selection; Use DOWN Command to View Next Selection Panel;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 58. Data Set Selection Entry Panel, DGTDDDS1

2. After the query is saved, retrieve the query name by specifying the QUERY(*name*) parameter in a batch job or by filling in the QUERY NAME TO SAVE OR RETRIEVE field on the data set or volume option panel of ISMF. Then issue the QRETRIEV command at the top of the panel.

Creating Model Commands

You can create a user-specified model command to build a set of commands for each data set or volume in the DCOLLECT data or saved ISMF list. The ACBQFLM1 EXEC uses a saved ISMF list and the ACBQADM2 EXEC uses DCOLLECT data.

How to Create Model Commands

To create DFSMSdss commands that delete selected data sets:

1. Use ISMF to build an ISMF table of the data sets you want deleted.
2. Use the model command facility to build your DSS commands.
3. Specify both the ISMF table and the DSS model command you want applied to all the data sets in the list.

To change the management class assigned to a group of data sets of a given high level qualifier whose user requirements have changed (high level qualifier ABC to management class MCSTAND) :

1. Use ISMF to create a list of data sets that meet the criteria you want.
2. Tailor the list to remove any data sets that you do not want processed by the model command function.
3. Save the ISMF list into a saved table.

Example of Saved Data Set List:

```
ABC.BACKUP.LAB.REVIEW
ABC.BACKUP.LAB.SCRIPT
ABC.BACKUP.LAB.JCL
```

Note: In steps 1 through 3 of this example, an ISMF-saved list is used as input to generate model commands. The remaining steps (4 through 8) apply to either ISMF-saved lists or DCOLLECT records as model command input.

4. Choose option 11.6. The ISMF “ Model Commands Generation Selection Menu” panel appears. See Figure 59.

```
Panel Help
-----
ACBSUTM3      MODEL COMMANDS GENERATION SELECTION MENU
Enter Selection or Command ==> _____

Select the input data to be used and press Enter:

  1 Saved ISMF List
  2 DCOLLECT Data

Use HELP Command for Help; Use END Command to Exit.
```

Figure 59. Model Commands Generation Selection Menu, ACBSUTM3

5. Select either option 1 or 2 (11.6.1 or 11.6.2).
6. Fill in the appropriate fields:
 - If you selected option 1 (ACBQFLM1 EXEC), enter the following information in the “ Model Commands Generation from Saved ISMF List Entry Panel”:
 - In the first field, the name of the table just saved
 - In the second field,

ALTER / MANAGEMENTCLAS(newmgmtclas)

Example of Model Command:

ALTER / MGMTCLAS(MCSTAND)

```

Panel  Help
-----
ACBDUTM1  MODEL COMMANDS GENERATION FROM SAVED ISMF LIST ENTRY PANEL
Command ==> _____

To generate model commands, specify the following information and press Enter:

  Saved ISMF List . . . _____ (Data Set List or DASD Volume List)

  Model Command for Generation
  ==> _____

Note: Available substitutions in the Model Command for Generation field
      are '/' for Data Set Name and '@' for Volume Serial.

Use HELP Command for Help; Use END Command to Exit.

```

Figure 60. Model Commands Generation from Saved ISMF List Entry Panel, ACBDUTM1

- You can also generate model commands with DCOLLECT data as input, instead of using saved lists. To do this, select option 2 (11.6.2) (ACBQADM2 EXEC) and enter the following information in the “Model Commands Generation from DCOLLECT Data Entry Panel”:

- In the first field, the name of the DCOLLECT data set
- In the second field,

ALTER / MANAGEMENTCLAS(newmgmtclas)

```

Panel  Help
-----
ACBDUTM2  MODEL COMMANDS GENERATION FROM DCOLLECT DATA ENTRY PANEL
Command ==> _____

To generate model commands, specify the following information and press Enter:

  Data Set Containing DCOLLECT Data
  ==> _____

  Model Command for Generation
  ==> _____

Note: Available substitutions in the Model Command for Generation field
      are '/' for Data Set Name and '@' for Volume Serial.

Use HELP Command for Help; Use END Command to Exit.

```

Figure 61. Model Commands Generation from DCOLLECT Data Entry Panel, ACBDUTM2

The model command function uses the same conventions as ISPF option 3.4. That is, a slash (/) may be located in the command wherever the user wants the data set name substituted. An “at” sign (@) may be used in the command wherever the user wants the volume serial number, which a data set resides on substituted. However, only two slashes and one @ may be used in each model command.

Attention: There is no syntax checking performed for the command entered by the user. It is the user's responsibility to correctly enter the command to be created against the entries in the table or DCOLLECT data.

Example Output from Command Generation:

```
ALTER ABC.BACKUP.LAB.REVIEW MGMTCLAS(MCSTAND)
ALTER ABC.BACKUP.LAB.SCRIPT MGMTCLAS(MCSTAND)
ALTER ABC.BACKUP.LAB.JCL MGMTCLAS(MCSTAND)
```

This output will be created in a sequential data set with the name USERID.Mxx.MODELCMD, where xx varies from 1 to 99.

7. Add JCL to the commands to complete the job. When the command generation completes, you are placed in ISPF "edit."

Example Job:

```
//ALTER JOB ----
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        ALTER ABC.BACK.LAB.REVIEW MGMTCLAS(MCSTAND)
        ALTER ABC.BACK.LAB.SCRIPT MGMTCLAS(MCSTAND)
        ALTER ABC.BACK.LAB.JCL MGMTCLAS(MCSTAND)
/*
```

Note: You can also use the SYSIN DD statement to reference the sequential data set that contains the output commands instead of the SYSIN statement followed by commands as shown above.

8. Submit the job for execution.

The model command facility can also be used with volume lists from ISPF tables or DCOLLECT data. The following example using volume lists creates a job to DEFrag all volumes in an ISMF volume list:

Example of Saved List:

```
SCR001 -----
SCR002 -----
SCR003 -----
SCR004 -----
SDV000 -----
SDV001 -----
SHR200 -----
```

Example of Model Command:

```
DEFrag DYNAM (@)
```

Example Output Command Generation:

```
DEFrag DYNAM(SCR001)
DEFrag DYNAM(SCR002)
DEFrag DYNAM(SCR003)
DEFrag DYNAM(SCR004)
DEFrag DYNAM(SDV000)
DEFrag DYNAM(SDV001)
DEFrag DYNAM(SHR200)
```


COPYFILT macro to place the consolidated filter list. The COPYFILT process copies the FILTLISTS from the FILTLIST member and places them between these two lines of special character strings.

How to Use the COPYFILT MACRO after Initial Setup

To use the COPYFILT macro *after* you have completed the initial setup, perform the following steps:

1. Change the FILTLIST member that is used by all four ACS routines.

To do this, edit the member containing the FILTLISTS. At the command line, save the member containing the FILTLISTS with the SAVE command.

2. Enter COPYFILT at the command line.

Note: The user must be in edit mode (that is, ISPF option 2; you cannot use ISPF option 3.4) for a member of the ACS source data set for this facility to work correctly.

The panel below appears, and the user is then prompted for the following information:

- Member names of the data class, storage class, management class, and storage group ACS routines inside the data set
- The member name containing those FILTLISTS
- A brief description, for change control, of the change made to the FILTLISTS

3. Press the Enter key. The four routines inside the data set are updated automatically, and the storage administrator receives a message that the update has taken place. If an error occurs, error messages indicate what the problem is.

For example, the fields in the following panel are filled in with typical information:

```
Panel Help
-----
ACBDFI01          COPY FILTLISTS ENTRY PANEL
Command ==>>>

To copy FILTLISTS, specify the following information and press Enter:

Member Containing FILTLISTS . . . . _____

Data Class ACS Routine Name . . . . DATACLAS
Storage Class ACS Routine Name . . . . STORCLAS
Management Class ACS Routine Name . . . . MGMTCLAS
Storage Group ACS Routine Name . . . . STORGRP_

Change Log Entry . . Migrating DB2 tables_____

Use HELP Command for Help; Use END Command to Exit.
```

Figure 62. Copy FILTLISTS Entry Panel, ACBDFI01

Glossary of Terms

The following terms are defined as they are used in the DFSMS/MVS Library. If you do not find the term you are looking for, see the IBM Software Glossary:

<http://www.networking.ibm.com/nsg/nsgmain.htm>

This glossary is an ever-evolving document that defines technical terms used in the documentation for many IBM software products.

A

ACS. Automatic class selection.

aggregate group. A collection of related data sets and control information that have been pooled to meet a defined backup or recovery strategy.

allocate. The process of assigning space on a device to hold a new data set. The allocation process is executed from ISPF/PDF, TSO, or through JCL statements.

attribute. Attributes refer to the different types of information ISMF can gather about your data sets, volumes, or SMS classes. Attributes correspond to the data column headings displayed on ISMF lists.

automatic class selection. A mechanism for assigning SMS classes and storage groups.

automatic class selection (ACS) routine. A procedural set of ACS language statements. Based on a set of input variables, the ACS language statements generate the name of a predefined SMS class, or a list of names of predefined storage groups, for a data set.

availability. For a storage subsystem, the degree to which a data set or object can be accessed when requested by a user.

B

base configuration. The part of an SMS configuration that contains general storage management attributes, such as the default management class, default unit, and default device geometry. It also identifies the systems or system groups that an SMS configuration manages.

base line test set. A special case test set that includes all test cases for data sets not planned to be SMS-managed. This test set has nulls (' ') as the expected values.

C

catalog. A data set that contains extensive information required to locate other data sets, to allocate and deallocate storage space, to verify the access authority of a program or operator, and to accumulate data set usage statistics.

CBIPO. Custom built installation process offering.

class. One of the following: data class, management class, storage class.

cluster. A data component and an index component in a VSAM key-sequenced data set; or a data component alone in a VSAM entry-sequenced data set.

coded character set identifier (CCSID) description. ISMF will display the first 17 characters of the default LOCALNAME element defined for the coded character set identifier (CCSID) in the character data representation architecture (CDRA) repository.

compress. (1) To reduce the amount of storage required for a given data set by having the system replace identical words, phrases, or data patterns with a shorter token associated with the word, phrase, or data pattern. (2) To reclaim the unused and unavailable space in a partitioned data set that results from deleting or modifying members by moving all unused space to the end of the data set.

construct. One of the following: data class, storage class, management class, storage group, aggregate group, base configuration.

construct application. An application to allow storage administrators to specify storage management policies for data in their computer complex. Some examples of construct applications are: management class application, storage class application, and data class application.

CPU. Central processing unit.

D

DASD volume. A DASD space identified by a common label and accessed by a set of related addresses. See also *volume, primary storage, migration level 1, migration level 2.*

data class (DC). A collection of allocation and space attributes, defined by the storage administrator, that are used to create a data set.

data classification. The process of determining data types and identifying different detailed data subtypes that require unique data services from DFSMS.

data entry panel. Any ISMF functional panel that prompts you for information that you must supply to ISMF to do a task. ISMF primes mandatory fields with default values or previously specified values.

Data Facility Product (DFP or DFSMSdfp). An IBM licensed program used to manage programs, devices, and data in an MVS operating environment.

data set. In DFSMS/MVS, the major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access. In non-OS/390 UNIX environments, the terms *data set* and *file* are generally equivalent and sometimes are used interchangeably. See also *file*.

data subtype. Groupings of data sets from a data type that requires identical DFSMS services.

data type. A major grouping of data that most installations have, for example, test, system, production, or temporary data.

DCOLLECT. An IDCAMS command for data collection that produces information on data sets, volumes, or configurations.

DCOLLECT. Data collection utility.

DFP. Data Facility Product.

DFSMS. Data Facility Storage Management Subsystem

DFSMS environment. An environment that helps automate and centralize the management of storage. This is achieved through a combination of hardware, software, and policies. In the DFSMS environment for MVS, the function is provided by DFSORT, RACF, and the combination of DFSMS/MVS and MVS.

DFSMS/MVS. An IBM System/390 licensed program that provides storage, data, and device management functions. When combined with MVS/ESA SP Version 5 it composes the base MVS/ESA operating environment. DFSMS/MVS consists of DFSMSdfp, DFSMSdss, DFSMShsm, and DFSMSrmm.

DFSMSdfp. A DFSMS/MVS functional component or base element of OS/390, that provides functions for storage management, data management, program management, device management, and distributed data access.

DFSMSdss. A DFSMS/MVS functional component or base element of OS/390, used to copy, move, dump, and restore data sets and volumes.

DFSMShsm. A DFSMS/MVS functional component or base element of OS/390, used for backing up and recovering data, and managing space on volumes in the storage hierarchy.

DFSMSrmm. A DFSMS/MVS functional component or base element of OS/390, that manages removable media.

dictionary. A table that associates words, phrases, or data patterns to shorter tokens. The tokens replace the associated words, phrases, or data patterns when a data set is compressed.

Distributed Data Management (DDM) attributes. DDM attributes are additional attributes to those required for local MVS data access. Some examples of associated DDM attributes are file size, hidden file, and lock options.

dummy storage group. A type of storage group that contains the serial numbers of volumes no longer connected to a system. Dummy storage groups allow existing JCL to function without having to be changed. See also *storage group*.

E

Enhanced Data Recording Format. In DFSMShsm, a method of compressing and encoding data during migration or backup to reduce storage space.

error. A state caused when the results are different from the saved expected results during regression testing. See also *exception*.

exception. A state caused when a NaviQuest ACS comparison report identifies a test case whose results differ from the saved expected results during either initial or regression testing. In a regression test case, the exception is an error. In an initial test case, the exception may or may not be an error. See also *error*.

expected results. The values assigned by the ACS routines for data class, storage class, management class, and storage group for a specific data subtype.

F

filter. To select data sets based on specified criteria. These criteria consist of fully or partially qualified data set characteristics. In ISMF you use the attributes of the list entries as the filtering criteria, and the relational operators to indicate the range of values to be displayed.

functional panel. Any ISMF application panel that is not a help panel. It can be a menu panel, a data entry panel, a list panel, a display panel, or a confirmation panel.

G

generic unit name. A name assigned to a class of devices with the same geometry (such as 3390). Contrast with *esoteric unit name*.

H

help panel. An ISMF panel that provides a snapshot description of task-oriented information relevant to the functional panel from which Help was invoked.

I

IDCAMS. The program name for access method services.

initial test. First-time test of a single data subtype. The initial test is done prior to converting the data to SMS and may include regression testing for other data subtypes that have already been tested successfully.

implementation phase. The cycle where data having one or more data subtypes with the same data type is converted to SMS management.

interactive. Pertaining to an application in which each entry calls forth a response from the system or program, usually in the form of a conversational dialog with screens or menus.

Interactive Storage Management Facility (ISMF). The interactive interface of DFSMS/MVS that allows users and storage administrators access to the storage management functions.

Interactive System Productivity Facility (ISPF). An IBM licensed program used to develop, test, and run application programs interactively. ISPF is the interactive interface for all storage management functions.

ISMF. Interactive Storage Management Facility.

ISPF. Interactive System Productivity Facility.

L

line operator. The term *line operator* refers to one of the two command entry modes, line versus list. The line operator is entered against a single entry in a list.

list application. An ISMF application which can list and manage Lists saved from various other ISMF applications.

list command. A list command refers to the list command entry mode. The command is entered on the command line and processed against all the entries in the list.

list panel. A list panel is a tabular display of data set names, volume serial numbers, management class names, data class names, or storage class names and their storage attributes.

logical. With respect to data, the attributes that describe the data and its usage, as opposed to the physical location of the data.

M

management class (MC). A collection of management attributes, defined by the storage administrator, used to control the release of allocated but unused space; to control the retention, migration, and back up of data sets; to control the retention and back up of aggregate groups, and to control the retention, back up, and class transition of objects.

master catalog. A catalog that points to user catalogs. See *catalog*.

migration. The process of moving unused data to lower cost storage in order to make space for high-availability data. If you wish to use the data set, it must be recalled. See also *migration level 1* and *migration level 2*

migration level 1. DFSMSHsm-owned DASD volumes that contain data sets migrated from primary storage volumes. The data can be compressed. See also *storage hierarchy*. Contrast with *primary storage* and *migration level 2*.

migration level 2. DFSMSHsm-owned tape or DASD volumes that contain data sets migrated from primary storage volumes or from migration level 1 volumes. The data can be compressed. See also *storage hierarchy*. Contrast with *primary storage* and *migration level 1*.

mountable optical volume. A portion of data stored on an optical disk.

MVS/ESA. An MVS operating system environment that supports ESA/390.

MVS/ESA SP. An IBM licensed program used to control the MVS operating system. MVS/ESA SP together with DFSMS/MVS compose the base MVS/ESA operating environment. See also *OS/390*.

N

NEVR. A non-VSAM data set volume record.

O

object. A named byte stream having no specific format or record orientation.

object access method (OAM). An access method that provides storage, retrieval, and storage hierarchy management for objects and provides storage and retrieval management for tape volumes contained in system-managed libraries.

object backup storage group. A type of storage group that contains optical or tape volumes used for backup copies of objects. See also *storage group*.

object code only (OCO). Licensed programs for which source materials are not made available to licensees.

object storage group. A type of storage group that contains objects on DASD, tape, or optical volumes. See also *storage group*.

object storage hierarchy. A three-level hierarchy consisting of objects stored in DB2 table spaces on DASD, on optical volumes that reside in a library, and on optical volumes that reside on a shelf. See also *storage hierarchy*.

optical volume. Storage space on an optical disk, identified by a volume label contained in object directory tables, which contains objects and available free space. An optical volume is one side of a double-sided optical disk. See also *volume*.

OS/390. OS/390 is a network computing-ready, integrated operating system consisting of more than 50 base elements and integrated optional features delivered as a configured, tested system. See also *MVS/ESA SP*.

P

panel. A predefined screen of information presented on a display terminal. Within ISMF, there are two types of panels: functional panels and help panels.

partitioned data set (PDS). A data set on direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

partitioned data set extended (PDSE). A system-managed data set that contains an indexed directory and members that are similar to the directory and members of partitioned data sets. A PDSE can be used instead of a partitioned data set

performance. (1) A measurement of the amount of work a product can produce with a given amount of resources. (2) In a DFSMS environment, a measurement of effective data processing speed with respect to objectives set by the storage administrator. Performance is largely determined by throughput, response time, and system availability.

phase test set. The group of subtype test sets that define data subtypes to be converted to SMS management in a single phase.

pool storage group. A type of storage group that contains system-managed DASD volumes. Pool storage groups allow groups of volumes to be managed as a single entity. See also *storage group*.

primary space allocation. Amount of space requested by a user for a data set when it is created. Contrast with *secondary space allocation*.

primary storage. A DASD volume available to users for data allocation. The volumes in primary storage are called primary volumes. See also *storage hierarchy*. Contrast with *migration level 1* and *migration level 2*.

priming. When ISMF displays a data entry panel already filled with the values that were in effect the last time you used the program, ISMF is said to prime the panel.

profile. Profile, or ISMF profile, refers to the set of program features provided by ISMF to run. Some of the profile features you may modify are the user mode, the mode used to recover from abends and log errors, and the mode used to run JCL job statements.

Program Development Facility (PDF). An IBM licensed program used with ISPF to provide a work environment for the development of programs and the use of functions such as Edit and Browse.

Q

qualifier. A modifier that makes a name unique. A data set name is made up of one or more qualifiers.

R

regression test. A test of data subtypes that have already been tested successfully and whose test cases (all in the subtype test set) have been updated with the expected results for the data subtype. A regression test always includes one initial test.

relational operator. A logical operator used to qualify a subgroup of data sets from a larger list. In ISMF, relational operators are used to filter data set lists.

Resource Access Control Facility (RACF). An IBM-licensed program or a base element of OS/390, that provides for access control by identifying and verifying the users to the system, authorizing access to protected resources, logging the detected unauthorized attempts to enter the system, and logging the detected accesses to protected resources.

Resource Measurement Facility (RMF). An IBM licensed program or optional element of OS/390, that measures selected areas of system activity and presents the data collected in the format of printed reports, system management facilities (SMF) records, or display reports. Use RMF to evaluate system performance and identify reasons for performance problems.

results. The values ACS routines assign for data class, storage class, management class, and storage group for a specific test case. These results may or may not be correct.

S

saved expected results. The expected results that have been saved in the text case after the successful initial test of the subtype test set. Saved expected results are used by the ACS comparison function during regression testing to make sure that each test case continues to be assigned the expected results.

secondary space allocation. Amount of additional space requested by the user for a data set when primary space is full. Contrast with *primary space allocation*.

selection entry panel. A type of data entry panel that prompts you for information about the objects (data sets, volumes, or SMS classes) you wish to include in a list. The more specific you are in your requirements,

the fewer entries will match your requirements and be included in the list.

SMF. System management facility.

SMS. Storage Management Subsystem.

SMS configuration. A configuration base, Storage Management Subsystem class, group, library, and drive definitions, and ACS routines that the Storage Management Subsystem uses to manage storage. See also *base configuration* and *source control data set*.

source control data set (SCDS). A VSAM linear data set containing an SMS configuration. The SMS configuration in an SCDS can be changed and validated using ISMF. See also *active control data set* and *communications data set*.

storage administrator. A person in the data processing center who is responsible for defining, implementing, and maintaining storage management policies.

storage class (SC). A collection of storage attributes that identify performance goals and availability requirements, defined by the storage administrator, used to select a device that can meet those goals and requirements.

storage group. A collection of storage volumes and attributes, defined by the storage administrator. The collections can be a group of DASD volumes or tape volumes, or a group of DASD, optical, or tape volumes treated as a single object storage hierarchy. See also *VIO storage group*, *pool storage group*, *tape storage group*, *object storage group*, *object backup storage group*, and *dummy storage group*.

storage hierarchy. An arrangement of storage devices with different speeds and capacities. The levels of the storage hierarchy include main storage (memory, DASD cache), primary storage (DASD containing uncompressed data), migration level 1 (DASD containing data in a space-saving format), and migration level 2 (tape cartridges containing data in a space-saving format). See also *primary storage*, *migration level 1*, *migration level 2*, and *object storage hierarchy*.

storage management. The activities of data set allocation, placement, monitoring, migration, backup, recall, recovery, and deletion. These can be done either manually or by using automated processes. The Storage Management Subsystem automates these processes for you, while optimizing storage resources. See also *Storage Management Subsystem*.

Storage Management Subsystem (SMS). A DFSMS/MVS facility used to automate and centralize the management of storage. Using SMS, a storage

administrator describes data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the system through data class, storage class, management class, storage group, and ACS routine definitions.

stripe. In DFSMS/MVS, the portion of a striped data set that resides on one volume. The records in that portion are not always logically consecutive. The system distributes records among the stripes such that the volumes can be read from or written to simultaneously to gain better performance. Whether it is striped is not apparent to the application program.

striped data set. In DFSMS/MVS, an extended-format data set consisting of two or more stripes. SMS determines the number of stripes to use based on the value of the SUSTAINED DATA RATE in the storage class. Striped data sets can take advantage of the sequential data striping access technique. See *striping* and *stripe*.

striping. A software implementation of a disk array that distributes data sets across multiple volumes to improve performance.

subtype prefix. A unique 1–4 character prefix associated with each data subtype, used to relate the data classification data subtype to the subtype test set. Also used to group all test cases together for a single subtype test set and in the member name for each test case in the subtype test set.

subtype test set. A group of test cases for data sets associated with the same data subtype from a subtype test set. All test cases within one subtype test set have the same expected results.

sysplex. Describes a collection of systems in a multisystem environment supported by the cross-system coupling facility.

system group. All systems that are part of the same sysplex and are running the Storage Management Subsystem with the same configuration, minus any systems in the sysplex that are explicitly defined in the SMS configuration.

system-managed storage. Storage managed by the Storage Management Subsystem. SMS attempts to deliver required services for availability, performance, space, and security to applications. See also *DFSMS environment*.

system programmer. A programmer who plans, generates, maintains, extends, and controls the use of an operating system and applications with the aim of improving overall productivity of an installation.

T

tape storage group. A type of storage group that contains system-managed private tape volumes. The tape storage group definition specifies the system-managed tape libraries that can contain tape volumes. See also *storage group*.

tape volume. A tape volume is the recording space on a single tape cartridge. See also *volume*.

temporary data set. An uncataloged data set whose name begins with & or &&, that is normally used only for the duration of a job or interactive session. Contrast with *permanent data set*.

test bed library. A PDS that contains a member that defines every test case.

test case. The definition of parameters associated with a single data set.

test PDS. A partitioned data set containing a member that defines every test case.

TSO/E. Time Sharing Option/Extended.

V

validate. To check the completeness and consistency of an individual ACS routine or an entire SMS configuration.

virtual input/output (VIO) storage group. A type of storage group that allocates data sets to paging storage, which simulates a DASD volume. VIO storage groups do not contain any actual DASD volumes. See also *storage group*.

volume. The storage space on DASD, tape, or optical devices, which is identified by a volume label. See also *DASD volume*, *optical volume*, and *tape volume*.

volume mount analyzer (VMA). A program that helps you analyze your current tape environment. With tape mount management, you can identify data sets that can be redirected to the DASD buffer for management using SMS facilities.

Index

A

- ACBJBAOB JCL 29
- ACBQADM2 136
- ACBQBAA1 95
- ACBQBAB1 92
- ACBQBAC1 102
- ACBQBAD1 52
- ACBQBAG1 99
- ACBQBAG3 98
- ACBQBAI2 33
- ACBQBAI4 38
- ACBQBAI6 42
- ACBQBAI9 82
- ACBQBAIA 90
- ACBQBAJ1 44
- ACBQBAJ2 72
- ACBQBAJ3 76
- ACBQBAJ7 66
- ACBQBAJ8 79
- ACBQBAJA 65
- ACBQBAJB 69
- ACBQBAO1 84
- ACBQBAO2 88
- ACBQBAO3 101
- ACBQBAR1 107
- ACBQBAR4 113
- ACBQBAR6 120
- ACBQBAR7 115
- ACBQBAR8 123
- ACBQBAS1 61
- ACBQBAU1 106
- ACBQBAX1 104
- ACBQFLM1 136
- ACBQVAR1 110
- ACS routine changes 14
- ACS routine translation option 84
- ACS test case comparison option 102
- activating DFSMS configuration 19
- aggregate group
 - alter 95
 - define 95
 - display 95
- automated testing, overview 7

B

- base configuration
 - alter 92
 - define 92
 - display 92

- base line 23
- base line test set 3
- batch mode, running ISMF 25
- batch option, examples 25
- batch testing and configuration management 25
- batch, JCL for 26
- bulk test case from ACBQBAO3 option 101
- bulk test case from DCOLLECT option 99
- bulk test case option 98

C

- capabilities, filtering 4
- changes
 - ACS routine 14
 - construct 14
- classification, data 5
- CLIST ACBQBAG3 98
- CLISTS 31
- collecting data for input 9
- collecting, data 4
- commands
 - QRETRIEV 135
 - QSAVE 135
- comparing regression testing 15
- comparison function 21
- construct changes 14
- converting to SMS management 20
- COPYFILT macro 140
- creating
 - customized tape reports 130
 - data set name reports 128
 - DCOLLECT reports 131
 - model commands 136
 - SMS reports 127
 - volume reports 129

D

- data class
 - alter 52
 - define 52
 - display 52
- data classification 3, 5
- data DCOLLECT, input 5
- data saved lists, input 4
- data set list 33
- data set name reports, creating 128
- data SMF, input 5
- data subtype 3
- data testing, example 20

- data type 3
- data VMA, input 5
- data, collecting 4
- DCOLLECT reports, creating 131
- determining errors 17
- DFSMS configuration, activating 19
- diagram, tree 5
- dummy storage group
 - alter 65
 - define 65

E

- enhancements, miscellaneous 135
- errors
 - batch option 25
 - data testing 20
 - definition 3
 - determining 17
- exceptions 21
- EXEC and CLIST table 31
- EXECs
 - ACBQADM2 136
 - ACBQBAA1 95
 - ACBQBAB1 92
 - ACBQBAC1 102
 - ACBQBAD1 52
 - ACBQBAG1 99
 - ACBQBAl2 33
 - ACBQBAl4 38
 - ACBQBAl6 42
 - ACBQBAl9 82
 - ACBQBAIA 90
 - ACBQBAJ1 44
 - ACBQBAJ2 72
 - ACBQBAJ3 76
 - ACBQBAJ7 66
 - ACBQBAJ8 79
 - ACBQBAJA 65
 - ACBQBAJB 69
 - ACBQBAO1 84
 - ACBQBAO2 88
 - ACBQBAO3 101
 - ACBQBAR1 107
 - ACBQBAR4 113
 - ACBQBAR6 120
 - ACBQBAR7 115
 - ACBQBAR8 123
 - ACBQBAS1 61
 - ACBQBAU1 106
 - ACBQBAX1 104
 - ACBQFLM1 136
 - ACBQVAR1 110
 - DCOLLECT 5
 - expected results 18
 - ISMF variables 135

- EXECs (*continued*)
 - prefix 4
 - saved lists test 4
 - SMF 5
 - test set 4
 - testing next 19
 - VMA 5
- expected results 3
- expected results, saving 18

F

- FILTLISTs, updating 14
- flowchart, testing example 21

G

- generating model commands 135
- generating test cases 9

I

- implementation phase 3
- initial test 3
- ISMF
 - options table 31
 - primary option menu 2
 - running in batch 25
 - test case option 90
 - variables
 - retrieving 135
 - saving 135
- ISPSTART batch parameters 26

J

- JCL for batch 26

L

- library, test bed 4

M

- macro, COPYFILT 140
- management class
 - alter 44
 - define 44
 - display 44
- menu, ISMF primary option 2
- method considerations 5
- method of testing, NaviQuest 21
- miscellaneous enhancements 135
- model commands, creating 136
- mountable tape list 42

N

- NaviQuest overview 1
- NaviQuest primary option menu 2
- NaviQuest, how to use to test 8
- NaviQuest, method of testing 21
- NaviQuest, testing procedure 8
- new terminology 3
- null results 23
- nulls 21

O

- object backup storage group
 - alter 66
 - define 66
- object storage group
 - alter 69
 - define 69
- option 1, data set list 33
- option 11, primary option menu 2
- option 11.1.1, test saved ISMF list 98
- option 11.1.2, test DCOLLECT data 99
- option 11.1.4, test VMA extract data 101
- option 11.2, compare ACS listings 102
- option 11.3, test enhanced ACS 104
- option 11.4, update test case results 106
- option 11.5.1, ISMF-saved data set report 107
- option 11.5.2, ISMF-saved volume report 110
- option 11.5.3, ISMF-saved tape volume report 113
- option 11.5.4, data set report from DCOLLECT 116
- option 11.5.5, DASD volume report from DCOLLECT 120
- option 11.5.6, SMS configuration reports 123
- option 2.1, volume list 38
- option 2.3, mountable table 42
- option 3.2, display management class 44
- option 3.3, define management class 44
- option 3.4, alter management class 44
- option 4.2, display data class 52
- option 4.3, define data class 52
- option 4.4, alter data class 52
- option 5.2, display storage class 61
- option 5.3, define storage class 61
- option 5.4, alter storage class 61
- option 6.2, define dummy storage group 65
- option 6.2, define object storage group 69
- option 6.2, define object backup storage group 66
- option 6.2, define pool storage group 72
- option 6.2, define tape storage group 76
- option 6.2, define VIO storage group 79
- option 6.3, alter dummy storage group 65
- option 6.3, alter object storage group 69
- option 6.3, alter object backup storage group 66
- option 6.3, alter pool storage group 72

- option 6.3, alter tape storage group 76
- option 6.3, alter VIO storage group 79
- option 6.4, storage group add/delete 82
- option 7.2, translate 84
- option 7.3, SCDS validation 88
- option 7.4.3, test ACS routines 90
- option 8.1, display base configuration 92
- option 8.2, define base configuration 92
- option 8.3, alter base configuration 92
- option 9.1, display aggregate group 95
- option 9.2, define aggregate group 95
- option 9.3, alter aggregate group 95
- overview automated testing 7
- overview, NaviQuest 1

P

- phase test set 3
- pool storage group
 - alter 72
 - define 72
- preface vii
- prefix, subtype 4
- primary option menu, ISMF 2
- primary option menu, NaviQuest 2
- publications, referenced viii

Q

- QRETRIEV command 135
- QSAVE command 135

R

- referenced publications viii
- regression test 3
- regression testing, comparing 15
- reports, creating 129
- results, expected 3
- retrieving ISMF variables 135
- REXX EXECs 31
- running ISMF in batch 25
- running test cases 14

S

- sample JCL library member list 27
- SCDS validation 88
- scenario, testing 20
- setup, testing environment 7
- SMS management, converting to 20
- SMS reports, creating 127
- storage class
 - alter 61
 - define 61
 - display 61

storage group add/delete option 82
 syntax
 ACBQBAA1 95
 ACBQBAB1 92
 ACBQBAC1 102
 ACBQBAD1 52
 ACBQBAG1 99
 ACBQBAG3 98
 ACBQBAGI2 33
 ACBQBAGI4 38
 ACBQBAGI6 42
 ACBQBAGI9 82
 ACBQBAJ1 44
 ACBQBAJ2 72
 ACBQBAJ3 76
 ACBQBAJ7 66
 ACBQBAJ8 79
 ACBQBAJA 65
 ACBQBAJB 69
 ACBQBAO1 84
 ACBQBAO2 88
 ACBQBAO3 101
 ACBQBAR1 107
 ACBQBAR4 113
 ACBQBAR6 120
 ACBQBAR7 116
 ACBQBAR8 123
 ACBQBAS1 61
 ACBQBAU1 106
 ACBQBAX1 104
 ACBQVAR1 110
 generating 9
 running 14
 SYS1.SACBCNTL 27
 system temporary data sets 21
 systemp 21

T

table, EXEC and CLIST 31
 tape storage group
 alter 76
 define 76
 terminology, new 3
 test ACS routines 90
 test bed library 4
 test case, definition 4
 test data restrictions 5
 test results, validating 17
 test set, subtype 4
 testbed library attributes 8
 testing
 ACS routines 7, 14
 environment setup 7
 FILTLISTs 14
 importance of 1

testing (*continued*)
 initial 3
 next subtype 19
 option 11.6 106
 phase set 3
 scenario 20
 test results 17
 trademarks v
 translate 84
 translating ACS routines 14
 tree diagram 5

V

VIO storage group
 alter 79
 define 79
 volume list 38

Readers' Comments — We'd Like to Hear from You

DFSMS/MVS Version 1 Release 5

NaviQuest User's Guide

Publication No. SC26-7194-01

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

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

Name

Address

Company or Organization

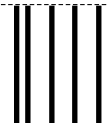
Phone No.



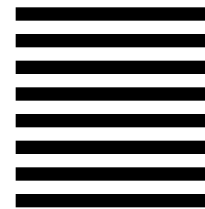
Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
RCF Processing Department
G26/050
5600 Cottle Road
SAN JOSE, CA 95193-0001



Fold and Tape

Please do not staple

Fold and Tape



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC26-7194-01



Spine information:



DFSMS/MVS Version 1 Release 5

NaviQuest User's Guide