

# **MAINVIEW<sup>®</sup> SRM SG-Control User Guide and Reference**

**Version 7.1**

**March 28, 2002**



Copyright 1994–2002 BMC Software, Inc., as an unpublished work. All rights reserved.

BMC Software, the BMC Software logos, and all other BMC Software product or service names are registered trademarks or trademarks of BMC Software, Inc. IBM and DB2 are registered trademarks of International Business Machines Corp. All other registered trademarks or trademarks belong to their respective companies.

THE USE AND CONTENTS OF THIS DOCUMENTATION ARE GOVERNED BY THE SOFTWARE LICENSE AGREEMENT ENCLOSED AT THE BACK OF THIS DOCUMENTATION.

## Restricted Rights Legend

U.S. GOVERNMENT RESTRICTED RIGHTS. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in FAR Section 52.227-14 Alt. III (g)(3), FAR Section 52.227-19, DFARS 252.227-7014 (b) or DFARS 227.7202, as amended from time to time. Contractor/Manufacturer is BMC Software, Inc., 2101 CityWest Blvd., Houston, TX 77042-2827, USA. Any contract notices should be sent to this address.

---

## Contacting BMC Software

You can access the BMC Software Web site at <http://www.bmc.com>. From this Web site, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

### United States and Canada

**Address** BMC Software, Inc.  
2101 CityWest Blvd.  
Houston TX 77042-2827

**Telephone** 713 918 8800 or  
800 841 2031

**Fax** 713 918 8000

### Outside United States and Canada

**Telephone** (01) 713 918 8800

**Fax** (01) 713 918 8000

---

---

## Customer Support

You can obtain technical support by using the Support page on the BMC Software Web site or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, please see “Before Contacting BMC Software.”

### Support Web Site

You can obtain technical support from BMC Software 24 hours a day, 7 days a week at <http://www.bmc.com/support.html>. From this Web site, you can

- read overviews about support services and programs that BMC Software offers
- find the most current information about BMC Software products
- search a database for problems similar to yours and possible solutions
- order or download product documentation
- report a problem or ask a question
- subscribe to receive e-mail notices when new product versions are released
- find worldwide BMC Software support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

### Support by Telephone or E-mail

In the United States and Canada, if you need technical support and do not have access to the Web, call 800 537 1813. Outside the United States and Canada, please contact your local support center for assistance. To find telephone and e-mail contact information for the BMC Software support center that services your location, refer to the Contact Customer Support section of the Support page on the BMC Software Web site at [www.bmc.com/support.html](http://www.bmc.com/support.html).

### Before Contacting BMC Software

Before you contact BMC Software, have the following information available so that Customer Support can begin working on your problem immediately:

- product information
  - product name
  - product version (release number)
  - license number and password (trial or permanent)
- operating system and environment information
  - machine type
  - operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - system hardware configuration
  - serial numbers
  - related software (database, application, and communication) including type, version, and service pack or maintenance level
- sequence of events leading to the problem
- commands and options that you used
- messages received (and the time and date that you received them)
  - product error messages
  - messages from the operating system, such as `file system full`
  - messages from related software

---

---

---

# Contents

<b>About This Book</b> .....	<b>xiii</b>
<b>Chapter 1</b>	<b>What Is SG-Control?</b>
	Overview..... 1-1
	New Terms ..... 1-2
	Product Definition..... 1-2
	SG-Control Functionality..... 1-3
	Modes of Operation ..... 1-5
	Budget Applications..... 1-6
	Reporting ..... 1-8
<b>Chapter 2</b>	<b>SGCONTRL - SG-Control Function</b>
	Overview..... 2-1
	SGCONTRL Function ..... 2-2
<b>Chapter 3</b>	<b>Planning for SG-Control Customization</b>
	Overview..... 3-2
	Tracking Space Allocation Requests ..... 3-2
	Assigning Applications to Data Sets ..... 3-3
	Monitoring and Controlling Space Utilization ..... 3-6
	Selecting a Mode ..... 3-7
	Monitor Mode ..... 3-7
	Warning and Reject Modes ..... 3-7
	Combining Modes ..... 3-8
	Assigning Budgets ..... 3-8
	Creating Applications..... 3-9
	Determining Report Requirements..... 3-10
	TSO BUDGET Command ..... 3-10
	COBOL Programs and SAS Procedures ..... 3-11
	Implementation Scenarios..... 3-11
	Scenario 1..... 3-11
	Scenario 2..... 3-19
	Scenario 3..... 3-25
	Scenario 4..... 3-28

<b>Chapter 4</b>	<b>Managing SG-Control Applications</b>	
	Overview . . . . .	4-1
	Maintaining Individual Applications . . . . .	4-3
	Maintaining Multiple Applications . . . . .	4-8
	WHERE Command . . . . .	4-12
	Viewing the Application List . . . . .	4-14
	Selecting an Individual Application . . . . .	4-15
	Displaying Data Set Information . . . . .	4-17
<b>Chapter 5</b>	<b>Batch Utility Programs</b>	
	Overview . . . . .	5-1
	Initializing and Updating the Database . . . . .	5-2
	Control Statements . . . . .	5-2
	Comments . . . . .	5-3
	Resynchronizing the Database . . . . .	5-11
	SGCMaint RESYNC Statement . . . . .	5-12
	SGCRSYNC Multitasking RESYNC . . . . .	5-16
	Diagnostic Reporting . . . . .	5-18
	SG-Control Database Reports . . . . .	5-18
	SG-Control Database Compare . . . . .	5-20
<b>Chapter 6</b>	<b>TSO Commands</b>	
	Overview . . . . .	6-1
	BUDDSN Command . . . . .	6-2
	Listing Application Codes for a Single Data Set . . . . .	6-2
	Listing Application Codes for a Group of Data Sets . . . . .	6-3
	Listing Application Codes for All Data Sets on a Volume . . . . .	6-3
	BUDGET Command . . . . .	6-4
	Command Coding Rules . . . . .	6-4
	Listing a Single Application . . . . .	6-5
	Querying Several Applications . . . . .	6-6
	Field Table for the BUDGET Command . . . . .	6-8
	Adding/Changing a Single Application . . . . .	6-9
	Mass Changes . . . . .	6-14
<b>Chapter 7</b>	<b>Reports</b>	
	Overview . . . . .	7-1
	TSO BUDGET Command as a Batch TMP . . . . .	7-2
	DFHSM Migration and Backup Report . . . . .	7-3
	SAS Reporting Procedures . . . . .	7-4
	COBOL Report Programs . . . . .	7-5
<b>Appendix A</b>	<b>SG-Control System Exits</b>	
	Overview . . . . .	A-1
	Installing Exits . . . . .	A-2
	Exit Flow . . . . .	A-3
	Security Exit . . . . .	A-4
	Application Code Build Exit . . . . .	A-6

---

Add Exit . . . . .	A-8
Check Exit. . . . .	A-9
Default Exit. . . . .	A-11
Select Exit. . . . .	A-12
Table of Sample Exits . . . . .	A-14
Application Records . . . . .	A-15
Global Work Area . . . . .	A-17
Special Considerations . . . . .	A-17

**Appendix B                    Special Functions of the TSO BUDGET Command**

**Appendix C                    SG-Control Error Resolution**

Technical Problems. . . . .	C-2
Fax Procedures . . . . .	C-2
Reserved DD Statements . . . . .	C-3
SGCTRACE DD Statement . . . . .	C-3
SGCDUMP DD Statement. . . . .	C-3

**Appendix D                    SGCDBCNV Utility**

**Glossary**

**Index**

---

---

# List of Figures

Figure 1-1	Check Processing . . . . .	1-4
Figure 1-2	Add Processing . . . . .	1-5
Figure 3-1	Scenario 1-Example JCL Using SGCMAINT for Database Initialization	3-13
Figure 3-2	Scenario 1-Example Batch TMP for Adding a Default Application .	3-15
Figure 3-3	Scenario 1-Example JCL to Resynchronize the Database with SG-Control Active . . . . .	3-17
Figure 3-4	Scenario 1-EZSRMSGC Menu . . . . .	3-18
Figure 3-5	Scenario 1-APPLTAB View . . . . .	3-19
Figure 3-6	Scenario 2-Example JCL to Update the Database Using SGCMAINT	3-21
Figure 3-7	Scenario 2-Example JCL to Resynchronize the Database Using SGCMAINT 3-23	
Figure 3-8	Scenario 2-Example Mass Change JCL Using a Batch TMP . . . . .	3-24
Figure 3-9	Scenario 2-Example Output After Mass Change . . . . .	3-24
Figure 3-10	Scenario 3-Example JCL to Update the Database Using SGCMAINT	3-26
Figure 3-11	Scenario 3-Example JCL to Resynchronize the Database Using SGC	3-27
Figure 3-12	Scenario 3-Example Output from the Budget Command . . . . .	3-28
Figure 3-13	Scenario 4-Example JCL Using SGCMAINT for Database Initialization	3-29
Figure 3-14	Scenario 4-Example Batch TMP for Adding a Default Application .	3-31
Figure 3-15	Scenario 4-Example JCL to Resynchronize the Database with SG-Control Active . . . . .	3-33
Figure 3-16	Scenario 4-Example Batch TMP to List Budget Applications . . . . .	3-34
Figure 3-17	Scenario 4-Example Output Produced from Batch TMP . . . . .	3-34
Figure 3-18	Scenario 4-Example Batch TMP to Customize Budget Applications	3-35
Figure 3-19	Scenario 4-Example Output After Customization . . . . .	3-36
Figure 4-1	EZSRMSGC Menu . . . . .	4-2
Figure 4-2	APPLDET Dialog Box . . . . .	4-3
Figure 4-3	Application Update Dialog Box . . . . .	4-4
Figure 4-4	Application Group Maintenance Update Panel . . . . .	4-8
Figure 4-5	Application Mass Update Tabular View . . . . .	4-11
Figure 4-6	APPLTAB View . . . . .	4-14
Figure 4-7	SG-Control Application Update Screen . . . . .	4-16
Figure 4-8	ACCTDET Dialog Box . . . . .	4-17
Figure 4-9	APPLDSNT View . . . . .	4-18
Figure 5-1	SGCMAINT JCL . . . . .	5-2

---

Figure 5-2	Sample Output - Example 1	5-19
Figure 5-3	Sample Output - Example 2	5-20
Figure 7-1	BUDGET TSO Command Batch TMP	7-3
Figure 7-2	Sample Output	7-3
Figure 7-3	DFHSM Migration and Backup Totals	7-4
Figure 7-4	Database Detail Report	7-7
Figure 7-5	Top Five Users of Permanent and VSAM Space	7-8
Figure 7-6	Applications over Warning Threshold Report	7-8
Figure 7-7	Summary of High-Water Marks	7-9
Figure 7-8	Application Graph	7-9
Figure 7-9	Alphabetic Listing of Applications	7-9
Figure 7-10	Database Trend Report	7-10

---

# List of Tables

Table 2-1	System Parameters Specific to SG-Control	2-2
Table 2-2	System Parameters Common to MAINVIEW SRM	2-3
Table 2-3	Rule List Parameter Reference	2-4
Table 3-1	Effect of Modes on SG-Control Processing	3-7
Table 4-1	MAINVIEW SRM Application Maintenance Options	4-2
Table 4-2	Application Maintenance Valid Field Entries	4-3
Table 4-3	Application Maintenance Input Fields	4-4
Table 4-4	Application Group Maintenance Valid Field Entries	4-9
Table 4-5	Valid Operands for the SORT Command	4-15
Table 4-6	Data Set Information Entry Panel Fields	4-18
Table 5-1	Application Code Fields	5-11
Table 6-1	Field Table for BUDGET Command	6-9
Table A-1	Sample Exits	A-14
Table A-2	Application Record Format	A-15

---

---

---

# About This Book

This book contains detailed information about MAINVIEW® Storage Resource Manager SG-Control by BMC Software and is intended for storage administrators.

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

- OS/390 operating system
- job control language (JCL)
- Interactive System Productivity Facility (ISPF)
- MAINVIEW SRM operations (see the *MAINVIEW SRM User Guide and Reference*)
- MAINVIEW SRM StorageGUARD (see the *MAINVIEW SRM StorageGUARD User Guide and Reference*)

## How This Book Is Organized

This book is organized as follows. In addition, an index and glossary appear at the end of the book.

Chapter/Appendix	Description
Chapter 1, "What Is SG-Control?"	provides an overall product description for SG-Control
Chapter 2, "SGCONTRL - SG-Control Function"	provides information about the SG-Control function
Chapter 3, "Planning for SG-Control Customization"	provides information and implementation scenarios to help you plan and tailor SG-Control for your site

Chapter/Appendix	Description
Chapter 4, "Managing SG-Control Applications"	explains how to manage the SG-Control database and create reports from the StorageGUARD data collector database
Chapter 5, "Batch Utility Programs"	describes the functions of the SGCMaint and SGCRYSNC utility programs
Chapter 6, "TSO Commands"	describes two TSO commands used in SG-Control
Chapter 7, "Reports"	describes methods of monitoring budget applications on your system and producing reports
Appendix A, "SG-Control System Exits"	explains how to install exits that allow you to extend the functions of the product
Appendix B, "Special Functions of the TSO BUDGET Command"	discusses how the BUDGET command is used to change current allocation amounts or high-water marks
Appendix C, "SG-Control Error Resolution"	describes how to prepare for communication with BMC Software Customer Support regarding SG-Control
Appendix D, "SGCDBCNV Utility"	tells you how to save the SG-Control database to 5.1 format for reporting purposes

## Related Documentation

BMC Software products are supported by several types of documentation:

- online and printed books
- online Help
- release notes and other notices

In addition to this book and the online Help, you can find useful information in the publications listed in the following table. As "Online and Printed Books" on page xv explains, these publications are available on request from BMC Software.

Category	Document	Description
MAINVIEW common documents	<i>OS/390 and z/OS Installer Guide</i> <i>MAINVIEW Installation Requirements Guide</i> <i>MAINVIEW Common Customization Guide</i> <i>Using MAINVIEW</i> <i>MAINVIEW Administration Guide</i> <i>Implementing Security for MAINVIEW</i>	provide instructions for installing, configuring, using, and administering MAINVIEW
MAINVIEW SRM customization documents	<i>MAINVIEW SRM Customization Guide</i>	provides instructions for configuring and customizing MAINVIEW SRM for OS/390 including SG-Control

Category	Document	Description
core documents	<i>MAINVIEW SRM User Guide and Reference</i>	provides information common to all MAINVIEW SRM products and high-level navigation
	<i>MAINVIEW SRM Reference Summary</i>	provides information about global system parameters, fileter and rule list parameters, and functions
messages	<i>MAINVIEW SRM Messages</i>	provides hardcopy of messages that are also available online
supplemental documents	release notes, flashes, technical bulletins	provides additional information about the product

## Online and Printed Books

The books that accompany BMC Software products are available in online format and printed format. If you are a Windows or Unix user, you can view online books with Acrobat Reader from Adobe Systems. The reader is provided at no cost, as explained in “To Access Online Books.” You can also obtain additional printed books from BMC Software, as explained in “To Request Additional Printed Books.”

### To Access Online Books

Online books are formatted as Portable Document Format (PDF) files. You can view them, print them, or copy them to your computer by using Acrobat Reader 3.0 or later. You can access online books from the documentation compact disc (CD) that accompanies your product or from the World Wide Web.

In some cases, installation of Acrobat Reader and downloading the online books is an optional part of the product-installation process. For information about downloading the free reader from the Web, go to the Adobe Systems site at <http://www.adobe.com>.

To view any online book that BMC Software offers, visit the support page of the BMC Software Web site at <http://www.bmc.com/support.html>. Log on and select a product to access the related documentation. (To log on, first-time users can request a user name and password by registering at the support page or by contacting a BMC Software sales representative.)

---

## To Request Additional Printed Books

BMC Software provides printed books with your product order. To request additional books, go to <http://www.bmc.com/support.html>.

## Release Notes and Other Notices

Printed release notes accompany each BMC Software product. Release notes provide current information such as

- updates to the installation instructions
- last-minute product information

In addition, BMC Software sometimes provides updated product information between releases (in the form of a flash or a technical bulletin, for example). The latest versions of the release notes and other notices are available on the Web at <http://www.bmc.com/support.html>.

## Conventions

This section provides examples of the conventions used in this book and explains how to read ISPF panel-flow diagrams and syntax statements.

### General Conventions

This book uses the following general conventions:

Item	Example
information that you are instructed to type	Type <b>SEARCH DB</b> in the designated field.
specific (standard) keyboard key names	Press <b>Enter</b> .
field names, text on a panel	Type <b>the appropriate entry</b> in the <b>Command</b> field.
directories, file names, Web addresses	The BMC Software home page is at <b>www.bmc.com</b> .
nonspecific key names, option names	Use the HELP function key. KEEPDICTIONARY option

Item	Example
calls, commands, control statements, keywords, parameters, reserved words	Use the SEARCH command to find a particular object.  The product generates the SQL TABLE statement next.
Unix commands, command options, database names	Use the <b>sbacktrack</b> program to create a backup script.
code examples, syntax statements, system messages, screen text	//STEPLIB DD  The table <i>table_name</i> is not available.
emphasized words, new terms, variables	The instructions that you give to the software are called <i>commands</i> .  In this message, the variable <i>file_name</i> represents the file that caused the error.
single-step procedures	>> To enable incremental backups, type <b>y</b> and press <b>Enter</b> at the next prompt.

This book uses the following types of special text:

**Note:** Notes contain important information that you should consider.

**Warning!** Warnings alert you to situations that could cause problems, such as loss of data, if you do not follow instructions carefully.

**Tip:** Tips contain useful information that may improve product performance or that may make procedures easier to follow.

## Syntax Statements

Syntax statements appear in Courier. The following example shows a sample syntax statement:

```
COMMAND KEYWORD1 [KEYWORD2|KEYWORD3] KEYWORD4={YES|NO}
      file_name...
```

The following table explains conventions for syntax statements and provides examples:

Item	Example
Items in italic type represent variables that you must replace with a name or value. Use an underscore for variables with more than one word.	<code>dtsbackup <i>control_directory</i></code>
Brackets indicate a group of options. You can choose at least one of the items in the group, but none of them is required. Do not type the brackets when you enter the option. A comma means that you can choose one or more of the listed options. You must use a comma to separate the options if you choose more than one option.	<code>[<i>table_name, column_name, field</i>]</code>
Braces enclose a list of required items. You must enter at least one of the items. Do not type the braces when you enter the item.	<code>{<i>DBD_name   table_name</i>}</code>
A vertical bar means that you can choose only one of the listed items. In the example, you would choose either <i>commit</i> or <i>cancel</i> .	<code>{<i>commit   cancel</i>}</code>
An ellipsis indicates that you can repeat the previous item or items as many times as necessary.	<code><i>column_name . . .</i></code>

---

---

# Chapter 1 What Is SG-Control?

This chapter provides a brief overview of SG-Control processing.

Overview . . . . .	1-1
New Terms . . . . .	1-2
Product Definition . . . . .	1-2
SG-Control Functionality . . . . .	1-3
Modes of Operation . . . . .	1-5
Budget Applications . . . . .	1-6
Reporting . . . . .	1-8

## Overview

SG-Control provides real-time monitoring, budgeting, and control of DASD space utilization. SG-Control provides the storage administrator with the ability to tie individual data sets to SG-Control applications. This feature allows the storage administrator to observe and control how space is being used, for example, by a programmer, project, department, and so on, within the organization.

SG-Control can be used to monitor how space is being used and track trends in DASD utilization, or to help anticipate and control future DASD growth. It should be noted that SG-Control application historical reporting features are not available unless SG-Control is integrated with the StorageGUARD data collector.

## New Terms

The following SG-Control and SG-Control-related terms are introduced in this chapter and are used throughout this manual:

- **Application**—SG-Control application is a grouping of information in the SG-Control database; that should represent an entity (a function, a user, a department, and so forth) with space utilization that requires monitoring or control or that requires both monitoring and control. This information is related to an application code.
- **Budget**—Maximum amount of space (as determined by your DASD space administrator) for a given SG-Control application.
- **DADSM**—Direct Access Device Space Management is a component of MVS/DFP. The DADSM exit routines are provided by IBM® to help your installation control DASD space requests and enforce installation standards.
- **DASD space administrator**—Individual at your installation assigned to perform DASD storage administration functions.
- **Default Application Code Construction Rules**—Set of rules held in the SG-Control database that defines the number of application codes that can be associated with a space allocation request and how construction rules are built.

## Product Definition

SG-Control is a component of MAINVIEW SRM designed to run on IBM mainframes to improve productivity and save time and money by helping to minimize space-related problems, maximize the use of DASD space, and assist in the automation of DASD storage management. SG-Control accomplishes these tasks by allowing your installation's DASD space administrator to monitor and set limits on space usage.

SG-Control monitors space allocation and deallocation on DASD using the DADSM preprocessing (IGGPRES00) and postprocessing (IGGPOST0) exits. SG-Control works in conjunction with any existing DADSM exits your installation may have established already. The SG-Control uses the DADSM exits to monitor and control DASD space requests.

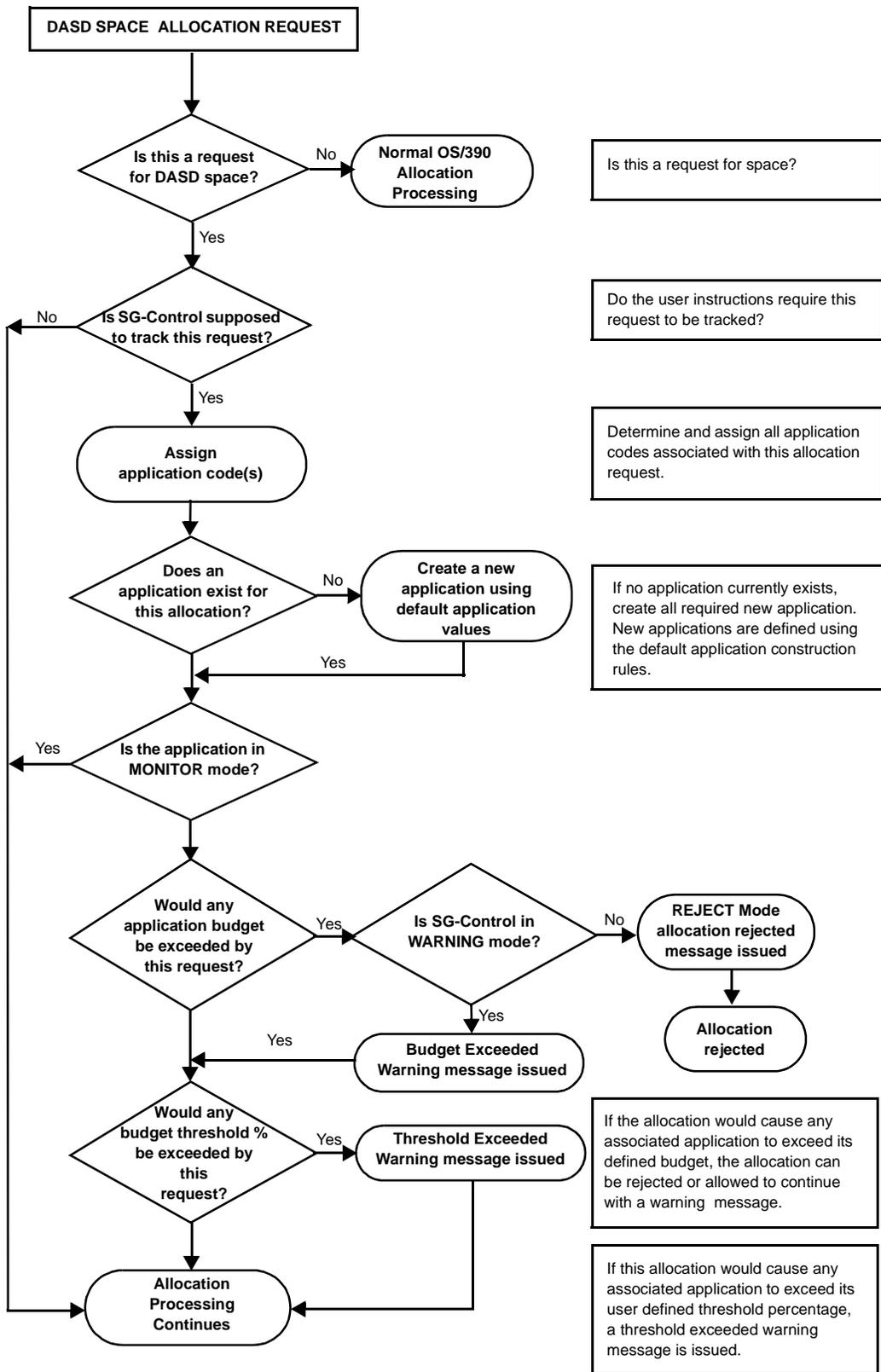
## SG-Control Functionality

The SG-Control flowcharts on the following pages illustrate how the software uses the DADSM exits to monitor and control DASD space requests.

### Check Processing

The DADSM preprocessing flowchart shown in Figure 1-1 on page 1-4 shows how SG-Control performs check processing. Check processing ensures that SG-Control tracks only the desired space request, dynamically creates applications in the SG-Control database (if necessary), generates over budget and warning threshold exceeded messages, and optionally, rejects allocation requests.

Figure 1-1 Check Processing



Is this a request for space?

Do the user instructions require this request to be tracked?

Determine and assign all application codes associated with this allocation request.

If no application currently exists, create all required new application. New applications are defined using the default application construction rules.

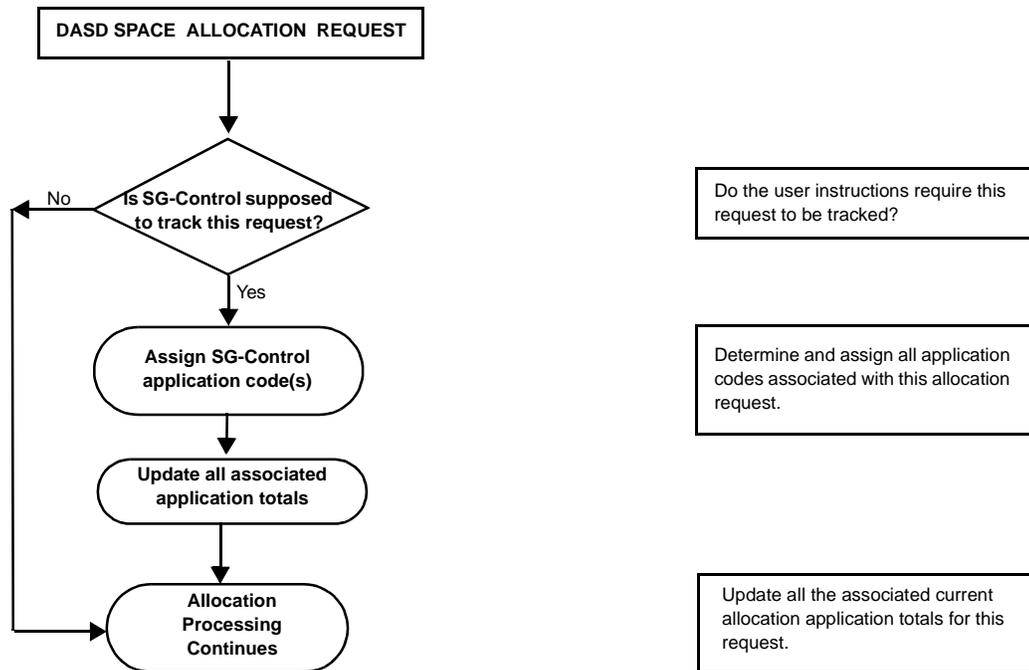
If the allocation would cause any associated application to exceed its defined budget, the allocation can be rejected or allowed to continue with a warning message.

If this allocation would cause any associated application to exceed its user defined threshold percentage, a threshold exceeded warning message is issued.

## Add Processing

The DASD postprocessing flowchart shown in the following figure shows how SG-Control performs add processing. Add processing updates the current allocation totals for the associated applications.

Figure 1-2 Add Processing



## Modes of Operation

SG-Control can be run in a combination of modes: monitor, warning, or reject.

- **Monitor** tracks space allocations and deallocations as they occur, allowing up-to-the-minute analysis of DASD space usage.
- **Warning** allows your DASD space administrator to specify (by application) how much DASD space has been assigned to the application. In other words, each application can be assigned a space budget. Separate budgets can be assigned for permanent, temporary, and VSAM space. SG-Control issues a budget exceeded warning message if a space allocation request will cause a current allocation total to exceed the budget for the application.

- **Reject** functions in the same way as warning mode, except that a space allocation request is rejected if it would cause a current allocation total to exceed the budget for the application.

SG-Control supports a combination of the above modes, allowing variations in the space administration policy for different groups of users or types of work and the phased implementation of warning and reject modes.

## Budget Applications

An application is a collection of information in the SG-Control database that is related to an application code. This information includes

- **Current allocation totals**—amount of space currently allocated on DASD for this application code
- **High-water marks**—largest amount of space used or reserved for this application code since high-water marks were last reset
- **Budgets for permanent, temporary, and VSAM space**—maximum amount of space (as determined by your DASD space administrator) allowed for this budget application

A number of user fields is also included that you can use to retain descriptive information and codes. For example, user fields can retain codes to collate information for reporting purposes.

There are two ways to create an application: dynamically or manually by using a TSO command or the MAINVIEW SRM ISPF interface. Initial information for an application can be copied from an existing application or from one of a set of default applications, or be provided by a user subroutine. For example, a default record can be used such that any dynamically created applications can have an unlimited budget, effectively implementing monitor mode for the applications. A more restrictive approach is to set up a default application with zero budgets. This could be used either to produce a warning message or to reject the allocation each time an allocation was attempted where the necessary applications had not been created manually or updated by the DASD space administrator.

### Assigning Applications to a Data Set

Numerous functions within SG-Control determine the applications associated with a data set. A data set can be associated with one to four application codes or it can be ignored by SG-Control.

Application codes can be determined from components of the data set name, data set organization, and volume serial. SG-Control provides three facilities that can be used to determine the application code:

- **Default Application Code Construction**, which can concatenate components extracted from the data set name, data set organization, or volume serial to form an application code. The default application code construction rules are specified when an SG-Control database is created. These default rules apply to all space allocation requests. Using the high-level qualifier of the data set name is a common use of this feature.
- **Filter and Rule List Processing**, which supports the use of conditional logic to construct the application code and to determine if a space allocation request should be tracked by SG-Control. This facility is especially useful when there is a need to group data sets with unlike or nonstandard names. Revised filter and rule lists can be activated at any time using the operator interface or an operator command.
- **User-Coded Subroutines**, which determine whether a space allocation request should be tracked and, if so, the associated application code.

**Note:** Support for user subroutines has been included in SG-Control for the sake of completeness. The application code build subroutine, as with the other SG-Control user subroutines, is rarely used.

## Reviewing and Updating Applications

Applications can be reviewed and updated in real time using the BUDGET TSO command, or MAINVIEW SRM ISPF interface. Access to the applications is controlled by a user-written subroutine. This feature is used to centralize or distribute the management of the applications.

Applications are accessed individually by specifying an application code or in sets by using the query feature. The query feature has several capabilities:

- A query can select all applications with a description starting with PROD (for example) or all the applications where the current allocation total for permanent space exceeds the budget amount.
- Options are provided to limit the amount of information returned for each application. A query can be specified to return only the application codes of applications over budget.
- Totals can also be calculated for all the applications that are selected by a query. A query can be specified to calculate and display the total space allocated for all applications with a description starting with PROD.

## Reporting

### **Batch**

SG-Control provides a series of COBOL and SAS reports that are used to detail application information and track DASD utilization.

By using each application's high-water marks, the trend report can identify growth patterns, improve the management of DASD space, and forecast future DASD requirements.

### **Online**

Online reporting for an individual application code is available through the MAINVIEW SRM ISPF interface. This feature offers you an up-to-the-minute picture of your DASD space utilization and availability.

---

---

# Chapter 2    **SGCONTRL - SG-Control Function**

This chapter describes the SGCONTRL function in detail.

Overview .....	2-1
SGCONTRL Function .....	2-2

## Overview

MAINVIEW SRM storage management services are divided into functions. SMFUNC $xx$  activates MAINVIEW SRM functions and controls message and tracing activity. Functions provide all the runtime services of MAINVIEW SRM. SMFUNC $xx$  points to members SMFLST $xx$  and SMRLST $xx$ , which select resources and control the operation of the functions.

Functions are defined in SET statements. You can change parameters

- by editing the member directly
- by using the MAINVIEW SRM function SET commands
- through the Functions option on the Parmlib Members pop-up menu of the EZSRM Menu (see “Chapter 6, How to Define and Activate Functions” in the *MAINVIEW SRM User Guide and Reference*).

## SGCONTRL Function

SG-Control operates with only one function, SGCONTRL. This chapter describes the parameters used with the SGCONTRL function.

The SGCONTRL function monitors space at allocation and deallocation using the DADSM exits (preprocessing and postprocessing) and any existing DADSM exits you may have established. Only user-defined space requests are tracked. Individual budgetary applications (logical groups) have preestablished space budgets (thresholds) and are updated dynamically. Space requests can be approved or denied at allocation.

Applications are multi-tiered (up to four levels) for increased reporting. A single data set can be part of different applications. For example, reports are available by user, project, department, and company. Application classification are based on DFSMS Storage Group, Management Class, Storage Class, Data Class, data set name, data set organization, application code level, volume, qualifier, or user-defined fields.

SGCONTRL reduces the required user input by using a dynamic application generator. Space usage is tracked dynamically and recorded in the SG-Control database to provide utilization totals and high-water marks. DASD is tracked for permanent, temporary, and VSAM data sets. Separate totals are also maintained for DFHSM migrate and backup data sets. High-water marks based on actual usage provide accurate trending and forecasting data.

### System Parameters

The system parameters that affect SGCONTRL are described in Table 2-1 and Table 2-2.

**Table 2-1 System Parameters Specific to SG-Control**

Parameter	Purpose
SGC_ADDEXIT	Defines the name of SG-Control Add Exit.
SGC_CHKEXIT	Defines the name of SG-Control Check Exit.
SGC_DEFEXIT	Defines the name of SG-Control Default Exit.
SGC_KEYEXIT	Defines the name of SG-Control Application Code Build Exit.
SGC_SECEXIT	Defines the name of the SG-Control Security Exit.
SGC_SELEXIT	Defines the name of SG-Control Select Exit.
SGC_STOGRP	Specifies whether to retrieve SMS storage group information. This parameter should be set to YES only if SMS storage group information is required for FLIST or RLIST processing.

**Table 2-1 System Parameters Specific to SG-Control**

Parameter	Purpose
SGC_STORCLS	Specifies whether to retrieve class information. This parameter may be set to YES only if SMS class information is required for FLIST or RLIST processing.
SGCDSN	Defines the data set name of the SG-Control database. The DDNAME SGCDB is dynamically allocated to this name with startup of SVSGC when the SGCDB DD is not present in the SVOS start up JCL.
SGCSCALE	Specifies the default scaling value for budget space amounts in SG-Control.

**Table 2-2 System Parameters Common to MAINVIEW SRM**

Parameter	Purpose
AOO_SYBSYS	Specifies the AutoOPERATOR subsystems that are to receive events.
BBI3_SSID	Specifies the CAS subsystem name to which the SVOS PAS should connect.
DIAG	Suffix of parameter member SMDIAGxx.
DIAGMSDD	Established WTO message tracing.
DISPLAY	Display functions list in the ISPF interface.
DUMPDD	Provides SYS1.DUMPxx dump if MAINVIEW SRM abends.
EVNT	Specifies the suffix you assign to the name of the SMEVNTxx event definition member.
FORPLEXNAME	Specifies one or more user-defined systems that can be included or excluded in a sysplex environment.
FORSMFID	Specifies SMF records that can be included or excluded in a sysplex environment.
FORSYSID	Specifies user-defined system IDs that can be included or excluded in a sysplex environment.
FUNC	Suffix of parameter member SMFUNCxx.
IGNOREDD	Suppresses MAINVIEW SRM activity for jobstep.
MODTRCDD	Sets module entry/exit tracing.
MSGID	Specifies the inclusion of the MAINVIEW SRM message identifier in the message text.
MSGLVL	Level of messages to be generated.
PASSWORD	Specifies the password(s) supplied by BMC Software.
POOL	Suffix of parameter member SMPPOOLxx.
SMFID	Record number for MAINVIEW SRM SMF records.
SYSLIB	Specifies a default data set to be allocated at SVOS startup
SYSLIB2	Specifies a default data set to be allocated at SVOS startup
SYSLIB3	Specifies a default data set to be allocated at SVOS startup.

**Table 2-2 System Parameters Common to MAINVIEW SRM**

Parameter	Purpose
TRACEDD	Traces MAINVIEW SRM activity for jobstep.
VAR	Suffix of parameter member SMVARSxx
WTORDC	Specifies the message descriptor coed(s).
WTORC	Specifies the routing code to be assigned to the message text. For more information, see the WTO macro ROUTCDE= parameter in the <i>MVS Supervisor Services and Macro Instructions</i> .

**Rule List Parameter Reference**

The following table presents syntax and descriptions for SG-Control rule list parameters (in member SMFLSTxx).

**Table 2-3 Rule List Parameter Reference**

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
NEWAPPL=xxxxxxxxx	Specifies the value of an application code that is used to override the default application code. The NEWAPPL parameter replaces the NEWACCT parameter, which also remains valid.
SCAN=EXIT	Specifies not to budget space for any level associated with a data set; this parameter is unique to SG-Control

**Parameter Explanations****EVENTID=**

**Purpose:** Specifies the identifier assigned to a user event in SMEVNTxx. This parameter will cause an event to be generated from this function.

**Note:** If EVENTID= is used on an FLST SET statement with MODE=INACT, the event will still be issued.

**Syntax:** EVENTID=xxxxx

where xxxxx is the 5-character string specified on the EVNTID parameter in SMEVNTxx.

**Default:** None

**NEWAPPL=**

**Purpose:** Specifies the value of an application code that is used to override the default application code. The NEWAPPL parameter replaces the NEWACCT parameter, which also remains valid.

**Syntax:** NEWAPPL=xxxxxxx

where xxxxxx is an alphanumeric code 1 to 50 characters long

When NEWAPPL=NONE is specified, no processing is done for the data set for the level(s) specified.

**Default:** None

### SCAN=

**Purpose:** Specifies not to budget space for any level associated with a data set for SG-Control

**Syntax:** SCAN=EXIT

### Filter List Parameters Supported

The following are the only filter list parameters that are available for the SGCONTRL function.

DATACLAS	DSNAME	DSNTYPE	DSORG
DSTYPE	HSM	HSMDSN	JOB
JOBACCT	JOBCLASS	JOBTYPE	LEVEL
LLQ	MGMTCLAS	OLDACCT	PGMRNAME
POOL	QUALL	QUALn	RACFGRP
RACFUID	SGC_FUNC	SMS	STORCLAS
STORGRP	SYSID	TEMPDSN	USRCn
VOLSER			

### Usage Notes

The NEWAPPL statement is used to override the default application code.

**Note:** SG-Control skips space budgeting on the current level if the new application code is blank. The new application code cannot exceed 50 bytes or the value specified for MAXKEY when the SG-Control database was created, whichever is less.

The syntax for defining a SET NEWAPPL= statement is

```
SET NEWAPPL='text', (keyword, displ, len, NOBLANK | BLANK), HIT
```

SG-Control can substitute values into your new application code symbolically so you can build codes with information from various sources. This feature is enabled by using the NEWAPPL statement operands.

**'text'**

**Function:** Contains the text of the replacement application code. The text can be from 1 to 50 characters in length. The text must be enclosed in single quotations if it contains embedded spaces or commas.

Data can be substituted into text dynamically by specifying a *hit* character in the text. The default hit character is ?. SG-Control scans the text for a question mark (?). Any question mark encountered is replaced with information in the next positional operand. Scanning for symbolic substitution continues until all question marks have been replaced. If the new text requires a question mark, the hit character can be modified with the HIT operand in the NEWAPPL statement. In the following example, the volume name is substituted into the new application code, concatenated to the literal SYSTEMS.

```
SET NEWAPPL=' SYSTEMS?' ,VOLSER
```

To substitute a substring of the operand following the hit character, enclose the substitution text and two more operands that specify the substring displacement and length in parentheses. The following example uses the first 3 bytes of the volume name.

```
SET NEWAPPL=' SYSTEMS?' , (VOLSER, 1, 3)
```

The new application code above is 10 characters long: the 7-character literal SYSTEMS followed by the first 3 characters of the VOLSER. For example, if the VOLSER is STR001, the NEWAPPL is SYSTEMSSTR.

To substitute variable information without literals, type one question mark for each substitution desired. In this example, the first-level qualifier will be suffixed by the last qualifier.

```
SET NEWAPPL=' ??' ,QUAL1 ,QUALL
```

NEWAPPL can be any combination of literals and variable information, but it cannot exceed the length of the application code as defined with the MAXKEY parameter during initialization.

**Default:** None

**Format:** Any valid text string from 1 to 50 characters in length.

*(keyword,displ,length,NOBLANK | BLANK)...*

**Function:** A positional operand that specifies the dynamic substitution data used to build the replacement application code for the SG-Control application dynamically.

You must specify as many of these positional operands as there are hit characters within the text. The keyword operand identifies the data to be inserted.

If you specify only the keyword, the entire string that it represents is used. You must use parentheses if you want a substring of the keyword. The displacement (displ) defaults to the first character and length defaults to the length associated with the keyword.

The following keywords are valid: QUAL1, QUAL2, QUAL3, QUAL4, QUAL5, QUAL6, QUAL7, QUAL8, QUALL, DSNAME, VOLSER, DSORG, USRC1, USRC2, TEMPDSN, LEVEL, OLDACCT, DATACLAS, STORCLAS, MGMTCLAS, STORGRP, SMS.

If you specify NOBLANK at the end of the operand, trailing blanks are removed from the string during substitution. The default (BLANK) does *not* remove blanks from the end of a string during substitution. Since these operands are positional, you must code a comma for those you omit.

In the following example, the first- and second-level qualifiers are substituted into the application. Any blanks are removed.

```
SET NEWAPPL='??', (QUAL1, , , NOBLANK), (QUAL2, , , NOBLANK)
```



---

---

# Chapter 3 Planning for SG-Control Customization

This chapter provides information and implementation scenarios to help you plan and tailor SG-Control for your site.

Overview . . . . .	3-2
Tracking Space Allocation Requests . . . . .	3-2
Assigning Applications to Data Sets . . . . .	3-3
Monitoring and Controlling Space Utilization . . . . .	3-6
Selecting a Mode . . . . .	3-7
Monitor Mode . . . . .	3-7
Warning and Reject Modes . . . . .	3-7
Combining Modes . . . . .	3-8
Assigning Budgets . . . . .	3-8
Creating Applications . . . . .	3-9
Determining Report Requirements . . . . .	3-10
TSO BUDGET Command . . . . .	3-10
COBOL Programs and SAS Procedures . . . . .	3-11
Implementation Scenarios . . . . .	3-11
Scenario 1 . . . . .	3-11
Scenario 2 . . . . .	3-19
Scenario 3 . . . . .	3-25
Scenario 4 . . . . .	3-28

## Overview

Customizing SG-Control requires careful planning and an understanding of the features and capabilities of the product. The major customization issues are as follows:

- Which space allocation requests do I want to track?
- How do I assign budget applications to data sets?
- What kind of space utilization do I want to monitor or control (permanent, temporary, HSM, VSAM)?
- What mode (monitor/warning/reject) should I use and how do I assign budgets?
- How do I create applications?
- What are my reporting requirements?

**Tip:** In addition to these customization issues, backup and recovery of the SG-Control database should be considered during the planning and customization of SG-Control. Use the RESYNC function of the SGCMAINT utility to accomplish these tasks. Additional information about SGCMAINT and details about backup and recovery procedures are available in “Resynchronizing the Database” on page 5-11.

## Tracking Space Allocation Requests

SG-Control has the ability, using filter list and rule list parameters, to track or ignore space allocation requests. SG-Control also provides a number of optional user routines that can be used to determine whether space allocation requests should be tracked. The tracking function can be used to implement SG-Control in steps or to restrict SG-Control to a set of jobs or users (for testing purposes).

### Tracking with Filter and Rule List Parameters

Filter list and rule list parameters can be used to evaluate the following items and to conditionally track or ignore a space allocation request:

- Data set name
- Data set attributes
- Volume serial

- Attributes of the job or jobstep requesting the space (see the note below)

**Note:** It is not recommended that the job or jobstep requesting the space be used. If the attributes of the job or jobstep that created the data set do not match the attributes of the job or jobstep that releases spaces or extends or deletes the data set, inconsistent results may be produced.

When SG-Control intercepts an allocation request, the system assigns the data set from one to four application codes based on the default application code construction rules. The filter list and rule list parameters have the ability to replace the control code or bypass the data set.

### **Tracking with Optional User Routines**

SG-Control also provides a number of optional user routines that can be used to determine whether space allocation requests should be tracked. These routines have been added for completeness and for users who prefer to code assembler subroutines rather than a selection language. Refer to the documentation on Select, Application Code Build, Check, and Add Exits in the Appendix A, “SG-Control System Exits” for more information.

**Note:** BMC Software recommends the use of the filter list and rule list parameters for maximum flexibility.

## **Assigning Applications to Data Sets**

The SG-Control database maintains information on the following items for each application:

- Budgets
- Current allocation totals
- High-water marks
- User fields
- System options

Applications are identified uniquely and accessed using an application code. A significant part of planning the implementation of SG-Control is defining the individual application codes. To define individual application codes

- identify what the application codes represent.

- determine the method of assigning one or more application codes to a space allocation request. An application code should represent an entity (a function, a user, a department, and so forth) with space utilization that requires monitoring or control or that requires both monitoring and control.

**Note:** The difficulty involved in assigning an application code to a space allocation request is relative to the implementation and adherence to data set naming standards. For example, if all data sets used by production jobs start with a high-level qualifier of PROD and the second qualifier identifies the application, it is very simple to assign application codes to monitor the space utilization for production and each application.

If you can determine the owner of a data set based on the data set name, the data set organization, and volume serial, the same logic can be used to assign application codes in SG-Control.

Application codes can be constructed by using the default application code construction rules, filter list and rule list parameters, and Application Code Build Exit. A fictitious company is used in the following demonstration of this three-step process.

Up to four application codes can be assigned to a request. Each application code is assigned a level, and at least one level must be specified. This feature can be used to implement a multi-level application structure. For example, the first level, Level 1, could be used to construct an application code that represents a division, Level 2, a department, Level 3, a project, and Level 4, a user.

### **Define Default Application Code Construction Rules**

The default application code construction rules are defined when a database is created and the rules are stored as part of the system options in the database. These rules allow you to construct application codes by concatenating up to four components extracted from the data set name, data set organization, and volume serial.

ABC International Corporation wants to monitor space utilization by division, by department within a division, and by user. The installation naming standard is as follows.

*AABB . CCCC . xxxxxxxx . xxxxxxxx*

where

*AA* identifies the division  
*BB* identifies the department

CCCCC identifies the user (user IDs are unique)

The following application codes are constructed using this naming standard:

- The Level 1 application code is constructed from the first two positions of the high-level qualifier.
- The Level 2 application code is constructed from the entire high-level qualifier.
- The Level 3 application code is constructed from the entire second-level qualifier.

The default application code construction rules are used to assign application codes to *all* space allocation requests.

**Note:** If an installation does not need instant access to space allocation totals at multiple levels and does not plan to implement warning or reject mode at multiple levels, it may be more efficient to use a single level and to specify user fields in the application record to collate information for reporting and inquiries.

### Specify Filter and Rule List Parameters

Next, the filter list and rule list parameters are invoked and pass the level number and application code. The filter list and rule list parameters conditionally can skip processing for an allocation request or a level, or they can replace the application code.

ABC International Corporation has decided to implement SG-Control in phases. Initially, the product will monitor space usage for selected departments in the Widget Manufacturing Division. Usage will be tracked for the planning and marketing departments. Usage will also be tracked for individual users in the planning and marketing departments.

The filter list and rule list parameter logic is as follows.

- Does the high-level qualifier start with WG (for Widget Manufacturing)? If not, ignore this space allocation request.
- Do the third and fourth positions of the high-level qualifier contain either PL (for planning) or MK (for marketing)? If not, ignore the level.

At a later date, the filter list and rule list parameters can be modified to include more departments and divisions. These changes to the filter list and rule list parameters are implemented in realtime; SG-Control does not have to be recycled.

The greatest value of the filter list and rule list parameters are demonstrated in a situation where there are multiple naming standards or no standards at all. If an installation can define a set of logical steps that can be used to identify the owner of a data set, this logic can be used in the filter list and rule list parameters to assign application codes.

The filter list and rule list parameters also can be used to replace an application code built by the default application code construction rules. This feature is used to consolidate a number of application codes into a single application or to replace the cryptic components of an application code with more meaningful text.

ABC International decided that an application code of WGMK was too cryptic.

A simple addition to the filter list and rule list parameters changes WGMK to WIDGET MKTG.

### **Pass Control to the Application Code Build Exit**

After the filter list and rule list parameters have processed an application code, control can be passed to the Application Code Build Exit. This exit has been added for completeness. It may also be used by users who prefer to code Assembler subroutines rather than use a selection language.

## **Monitoring and Controlling Space Utilization**

SG-Control can maintain separate allocation totals for the following types of space:

- Permanent
- Temporary
- VSAM
- DFHSM

Temporary, VSAM, and HSM space allocations can be ignored or combined into the permanent allocation totals. Allocation totals are maintained in terms of bytes.

SG-Control identifies temporary data sets as data sets with system-generated names. Named data sets with a disposition of (new, pass) or (new, delete) are not considered temporary because a subsequent jobstep or CLOSE could be used to change the final disposition to CATLG or KEEP. SG-Control could not detect this change to the final disposition.

## Selecting a Mode

SG-Control functions in the following modes:

- Monitor
- Warning
- Reject
- A combination of the above

### Monitor Mode

It is recommended that initially you implement your system in monitor mode and specify a budget for all applications running in monitor mode so that information on DASD space utilization is gathered. Budgets are used by the MAINVIEW SRM ISPF Interface as the basis for percentage usage calculations and graphic and tabular displays. In monitor mode, the budget is used by the system only for MAINVIEW SRM ISPF Interface calculation reporting purposes. The information gathered in the reports helps to determine how space is being used in an installation and if there is a need for warning or reject mode.

### Warning and Reject Modes

Applications set to warning or reject mode may be assigned a budget for allocation. In warning mode, when the limits of the budget are approached or exceeded, a warning message is written to the SYSLOG and joblog and sent to the TSO user (if the request was from a TSO session). In reject mode, a warning is also issued when the limits of the budget are approached. Additionally, if a space allocation request will cause an allocation total to exceed the budget, the request is rejected and reject messages are issued.

The following table shows the effect of modes on SG-Control processing.

**Table 3-1 Effect of Modes on SG-Control Processing**

Current Application Mode	MONITOR	WARNING	REJECT
Warning threshold percentage exceeded. Warning message?	N	Y	Y
Over Budget Warning Message is set.	N	Y	N
Allocation Rejected?	N	N	Y

## Combining Modes

Modes can be implemented globally by specifying them to default as system options. The system option is stored in the SG-Control database and may be overridden by setting modes as application options when an application is set up. Specifying modes as application options allows variations in the space administration for different groups of users or types of work, and it allows for the phased implementation of warning and reject modes. Full implementation of the SG-Control modes requires the use of both system default and application modes.

## Assigning Budgets

Budgets are changed using the BUDGET TSO command or the MAINVIEW SRM ISPF Interface.

- The BUDGET TSO command is used to change individual applications or to change groups of applications.
- Individual applications are selected by using their application codes.
- Applications to be changed in groups are selected by using a query statement. A query statement can be used to select applications based on the value in a field or by comparing two fields in the application record. For example, a BUDGET TSO command could be coded to change the mode and permanent allocation budget of all applications to 500KB where user field 1 contains the literal TEST ACCT.
- Individual applications can be changed interactively, or a number of applications can be changed quickly and efficiently by running the budget commands under a batch TMP.

---

# Creating Applications

Applications can be created automatically or manually.

## Automatically

Applications can be created automatically as needed. If SG-Control constructs a new application code during the processing of a space allocation request, a new application is created automatically. The SG-Control resynchronization function (RESYNC) also supports this feature. RESYNC is used to ensure that a database reflects DASD utilization accurately and provides a method of populating a new database with application information.

New applications created automatically can be customized using the TSO BUDGET command.

## Manually

Applications can be created manually using the TSO BUDGET command. The values used to initialize the budgets, user fields, application options, and so forth, are obtained by one of the following methods:

- Values can be extracted from a default or model application. Typically, an installation establishes a single default application with an application code of CONTROL. Values are extracted from this application whenever a new application is created. Model applications can be created using the BUDGET TSO command. An active application also can be used as a model.
- Values can be provided by a user subroutine. The Default Exit allows an installation to determine programmatically the application code of the model application that should be used when creating an application automatically. The application code of the model application can be assigned based on the new application code data set name, data set organization, or volume serial. This subroutine also can be used to replace individual fields in the new application.
- Values can be specified using the TSO BUDGET command. The TSO BUDGET command also supports a FROM parameter that can be used to provide the application code of a model application. This facility allows the DASD Space Administrator to choose a model from a set of model applications when creating new applications manually.

**Note:** BMC Software recommends adopting a unique application code naming convention to help identify model applications.

## Determining Report Requirements

StorageGUARD provides a number of facilities that can be used to report space utilization information:

- The MAINVIEW SRM interface can be used to display up-to-the-minute space utilization and historical space information. This is the recommended method for displaying application information and updating applications. See the *MAINVIEW SRM User Guide and Reference* for more information on using the MAINVIEW SRM interface to view and control SG-Control budgets. The BUDGET TSO command can be used to select and display up-to-the-minute space utilization information.
- A number of COBOL programs and SAS procedures are provided to print a variety of reports.

### TSO BUDGET Command

The TSO BUDGET command selects applications by specifying an application code or by using a query.

For example, a TSO BUDGET command can be used to display the total space utilization of all applications with an application code that starts with ACCTNG to show how much space the accounting department is currently using. Or, it could be used to display the application codes used by the accounting department.

Queries can select applications by comparing a named field with a value or by comparing the values in two fields in an application. For example, a query could select all the applications that currently have more than 10,000KB allocated. Or, the current allocation totals for each application can be compared to the budgeted totals for each application to select applications that are over budget.

A number of options can then used to specify how the selected information should be displayed. These options can display totals, all of the application information, or just the application codes of the selected applications.

The TSO BUDGET command also can be formatted and run as a batch TMP.

## COBOL Programs and SAS Procedures

SG-Control includes a number of sample COBOL report programs and SAS procedures. Since the reporting requirements vary by each installation, these programs and procedures are provided as models. Sample reports include the following information:

- Alphabetic listing of applications
- Largest user of DASD space
- Applications over warning level
- Summary of high-water marks
- Application Space Utilization Graph

A series of trend report programs and SAS procedures can be used to monitor the changes in DASD space utilization over a period of time.

## Implementation Scenarios

The following sections present four different implementation scenarios. These scenarios are designed to help the SG-Control Administrator and the DASD Space Administrator identify and analyze company requirements and then successfully implement SG-Control.

Each scenario starts with a discussion of the requirements and continues with a detailed description of each step needed to implement the product based on certain requirements.

Use these scenarios as a guide as you plan and implement SG-Control.

### Scenario 1

ABC International has decided to implement SG-Control in phases. Initially, the product will be used to monitor the space utilized by each department in the Widget Manufacturing Division. Fortunately, ABC International's adoption and adherence to a data set naming standard makes the identification of Widget's data sets an easy task.

The ABC International data set naming standard is

*AABB . XXXX . XXXX . XXXX*

here

*AA* identifies the division (Widget Manufacturing Division)  
*BB* identifies the department within the division

Widget Manufacturing is identified by a division code of WG. The department codes used by Widget Manufacturing are as follows:

DV	Development
MK	Marketing
PL	Planning
PR	Production
RS	Research

### **Decisions**

#### ABC International Implementation Decisions

- Allocation requests will be tracked for the Widget Manufacturing Division only.
- Five SG-Control applications will be created to maintain individual totals for each department. Additional applications will be provided to allow for growth.
- To ensure that the application codes used to access the application information are unique, the application codes will be constructed from both the division and department code.
- Individual space allocation totals will be maintained for permanent, temporary, and VSAM space. Allocation totals will not be maintained for DFHSM space.
- SG-Control will be implemented in monitor mode. A default application will be created with the budgets for permanent, temporary, and VSAM space. These budgets will be used by the MAINVIEW SRM ISPF interface for percentage utilization calculations and reporting.
- The SG-Control database detail report will be run weekly and the MAINVIEW SRM ISPF interface will be used as needed to display up-to-the-minute space allocation information.

## Instructions

### Step-by-Step Instructions for Implementing ABC International Decisions

#### Step 1 Initialize the database.

The database is initialized using SGCMAINT, the SG-Control utility program. This multi-function program is used to initialize, copy, resynchronize, report, and compare the contents of SG-Control databases. The functions of this program are controlled by a series of control statements. The initialization (or INIT) control statements are used to allocate the database and define the system options and default application code construction rules.

Following is JCL that is used to initialize the ABC International database:

**Figure 3-1 Scenario 1-Example JCL Using SGCMAINT for Database Initialization**

---

```
//EXAMPLE JOB
//*
//*      INITIALIZE THE DATABASE
//*
//INIT    EXECPGM=SGCMAINT
//STEPLIB DD  DISP=SHR,DSN=XXXXXXXX.BBLINK
//SYSPRINT DD  SYSOUT=*
//SYSUT1  DD  UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN   DD  *
INIT FILE(NEW)
INIT DBNAME(XXXXXXXX.SPACEDB)
INIT RECORDS(100) RECVOLS(10)
INIT VOLUME(<DBVOL>)
INIT LEVEL(1) FIELD1(QUAL1,1,4) RECALC
INIT MODE(MONITOR)
INIT MAXKEY(7)
/*
```

---

#### Control Statement Explanations

- INIT FILE(NEW)  
Indicates a new database is to be created.
- INIT DBNAME (xxxxxxxx.SPACEDB)  
INIT RECORDS(100) RECVOLS(10)  
INIT VOLUME(<DBVOL>)

These three statements provide the data set name, capacity, and location of the new database. The RECVOLS parameter controls the number of volume totals that can be maintained for each record. Maintaining volume totals gives SG-Control the ability to resynchronize a single volume and maintain accurate application totals. The number specified for RECVOLS should equal the maximum number of volumes that will contain data sets associated with any one application. Multiple records will be used for an application if the data sets associated with the application are on more than the value in RECVOLS. ABC International has also included an allowance for growth.

The size of records in the SG-Control database is calculated based on the RECVOLS specification. The VOLUME parameter specifies the location of the database. The database should be located on a low contention volume, ideally with a cached controller. The database should not be placed on a volume with the JES2 check point, page/swap data set, or on any volume that will be held with reserves regularly.

- INIT LEVEL(1) FIELD1(QUAL1,1,4) RECALC

Defines the default application code construction rules in the system options. In this example, ABC International will be using one level; that is, only one application code will be assigned to each space allocation request. The application code will be constructed from the first four characters of the first qualifier of the data set name.

- INIT MODE(MONITOR)

Sets the mode system option.

- INIT MAXKEY(7)

The MAXKEY parameter defines the length of the longest application code used in the database.

After SGCMAINT has allocated and formatted the new database successfully, the next step is to review global parameter options before adding a default application.

## **Step 2 Review global parameter options.**

Before defining application options, review the global options coded in SMMSYSxx global system members. To access the global system members, select the System choice from the Parmlib Members pop-up menu. Review SGC\_ options related to application definitions.

## **Step 3 Add a default application.**

The TSO BUDGET command can be used to add, list, query, and change applications in the database. The command can be executed from a TSO session or a batch TMP. The following figure shows how the default application is added to the database using a batch TMP.

**Figure 3-2 Scenario 1-Example Batch TMP for Adding a Default Application**

---

```
//*
//*      ADD THE DEFAULT APPLICATION TO THE DATABASE
//*
//ADDDEF  EXEC PGM=IKJEFT01,DYNAMNBR=10
//STEPLIB DD  DISP=SHR,DSN=<APFLIB>
//SYSPRINT DD  SYSOUT=*
//SYSTEM  DD  SYSOUT=*
//SYSIN   DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN DD  *
        PROFILE NOPREFIX
        ALLOC FILE(SGCDB) DSN(XXXXXXXX.SPACEDB) SHR
        BUDGET 'CONTROL' ADD NAME('SG-CONTROL DEFAULT')+
        MAXPERM(100000) MAXTEMP(100000)+
        MAXVSAM(100000)
/*
```

---

Since SG-Control has not been started, the database must be allocated. The TSO BUDGET command is then used to add a default application to the database. An application code of CONTROL identifies the application as being the default application. The value specified for the name and budgets for permanent, temporary, and VSAM space (MAXPERM, MAXTEMP, and MAXVSAM) is included in the default application.

**Step 4 Build the SG-Control filter list and rule list members.**

FLST and RLST parameters are used to select which DASD space allocation requests should be tracked and to replace application codes. In this example, the filter list and rule list members have been coded to track only space allocation requests for Widget Manufacturing. All of Widget Manufacturing data set names start with the division code WG. All other space allocation requests, including those issued by DFHSM, are ignored.

**Note:** FLST and RLST parameters are coded in filter list and rule list members accessed through the SGCONTRL function of the Functions choice on the EZSRM menu. Filter list and rule list members may also be accessed through the FLST/RLST choice on the Parmlib Members pop-up menu.

The following example shows FLST parameters:

---

```
SET MODE=ACT          TRACK REQUESTS FOR WIDGETS
INC QUAL1=WG/         BYPASS REMAINING REQUESTS
SET MODE=INACT
INC QUAL1=/
```

---

The following example shows RLST parameters:

---

```
SET NEWAPPL='?',OLDAPPL  USE THE DEFAULT APPLICATION
INC QUAL1=/
```

---

Refresh the filter list and rule list members to ensure the parameters are syntactically correct.

#### **Step 5 Start SG-Control.**

SVOS is the started task that activates SG-Control dynamically. If you are unfamiliar with the capabilities and functions of SVOS, it is important that you review operator services in the *MAINVIEW SRM User Guide and Reference*.

To activate the new database:

##### **5.A** Shutdown SG-Control by issuing the following command:

```
P SVSGC
```

##### **5.B** Restart SG-Control using the following command:

```
S SVSGC
```

#### **Step 6 Resynchronize the database.**

The RESYNC function of the SGCMAINT program can be used to populate the database with application information quickly and efficiently. Refer to “Resynchronizing the Database” on page 5-11 for detailed instructions.

The database can be resynchronized with SG-Control active or inactive. If SG-Control is inactive when the RESYNC is run, the application information may be accurate only for a short period of time. Any DASD space allocation requests completed after the RESYNC while SG-Control was inactive are not reflected in the database.

The database also can be resynchronized while SG-Control is active. This approach ensures that SG-Control will load and maintain accurate application information. The RESYNC function can be run against all mounted volumes (default) or against selected volumes.

Widget Manufacturing permanent and VSAM data sets are pooled in a series of volumes whose volume serials start with WIDG. Temporary data sets are allocated on volume serial SCR150.

ABC International decided to RESYNC the database with SG-Control active.

The RESYNC was restricted, using VOLUME statements, to volumes that contained Widget Manufacturing data sets. Use the following JCL to resynchronize the database with SG-Control active.

**Figure 3-3 Scenario 1-Example JCL to Resynchronize the Database with SG-Control Active**

---

```
//*
//*          RESYNCHRONIZE THE DATABASE
//*
//RESYNC    EXEC PGM=SGCMAINT
//STEPLIB   DD  DISP=SHR,DSN=XXXXXXXX.BBLINK
//SYSPRINT  DD  SYSOUT=*
//SYSUT1    DD  UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN     DD  *
RESYNC ACTIVEDB
VOLUME INCLUDE(WIDG*)
VOLUME INCLUDE(SCR150)
//*
```

---

### **Step 7 Review and modify budget applications.**

Once the database has been populated with application information, the MAINVIEW SRM ISPF interface or the TSO BUDGET command can be used to review and modify applications.

ABC International decides to use the MAINVIEW SRM ISPF interface to review the permanent allocation totals and update the name field with the descriptive name of each application.

The simplest way to review and manipulate application information in the manner required by ABC International is to select the SG-Control Application choice from the EZSRM menu. The EZSRMSGC menu is displayed as shown in Figure 3-4 on page 3-18.

**Figure 3-4 Scenario 1-EZSRMSGC Menu**

```

14MAY2001 09:24:29 ----- INFORMATION DISPLAY -----
COMMAND ==>
CURR WIN ==> 1          ALT WIN ==>
>W1 =EZSRMSGC=====SJSH=====*=====14MAY2001==09:24:29===MVSMSGC=D===1
EZSRMSGC Menu

SG-Control          +-----+          SRM Administration
                    |         |          |
. Application Maint  | Place cursor on | > Parmlib Members
                    | menu item and   | . Functions
. Application Mass Upd | press ENTER   | . SRM Component Status
. Application List    +-----+ . MVSMSGC View List
. Data Set Information . MVSMSGC Batch Reports
                    . Return...

```

**Note:** Online help is available from each of the MAINVIEW SRM panels. Press the PF1 key to obtain help text.

Select the Application List and press **Enter**. The APPLTAB view is displayed as shown in Figure 3-5 on page 3-19. This view lists all the application codes in the database, and it displays permanent and temporary allocated, high-water mark, and budget figures for each record. See the Chapter 4, “Managing SG-Control Applications” for complete instructions on using the APPLTAB view.

Figure 3-5 Scenario 1-APPLTAB View

```

30MAY2001 15:19:48 ----- INFORMATION DISPLAY -----
COMMAND ==>                                     SCROLL ==> PAGE
CURR WIN ==> 1           ALT WIN ==>
>W1 =APPLTAB=====SJSH=====*****30MAY2001==15:19:25====MVSMSGC=D=4204
C ApplicationName      Appl Permanent Permanent Permanent Temporary Tempora
- -----            Stat Budget   Allocated H-W-M      Budget   Allocat
ACCO8                  DEL    976.6Mi   113328    113328  512000000
ACCO9                  DEL    976.6Mi   113328    113328  512000000
ACCO10                 DEL    976.6Mi   113328    113328  512000000
ACCO11                 DEL    976.6Mi   113328    113328  512000000
ACCO12                 DEL    976.6Mi   113328    113328  512000000
ACCO13                 DEL    976.6Mi   113328    113328  512000000
ACCO14                 DEL    976.6Mi   113328    113328  512000000
ACCO15                 DEL    976.6Mi   113328    113328  512000000
ACCO16                 DEL    976.6Mi   113328    113328  512000000
ACFUSER                DEL    976.6Mi   283320    283320  512000000
ACFWK1                 ACTV   976.6Mi   9746208   10709496 512000000
ACFWK2                 ACTV   976.6Mi  255044664 255951288 512000000
ACFWK3                 ACTV   976.6Mi    2.6Gi    2.6Gi  512000000
ACFWRK                 ACTV   976.6Mi  1699920   1699920  512000000
ADSM                   DEL    976.6Mi  59667192  59667192 512000000
AES                    DEL    976.6Mi  35471664  35471664 512000000

```

## Scenario 2

ABC International decides to use SG-Control to control the space used by the development department of the recently acquired Grommet Manufacturing Division. ABC International currently is using SG-Control to monitor the space used by the Widget Manufacturing Division.

Grommet's data set naming standard is

```
GROM.D####.xxxxxxxx
```

where

*GROM* identifies the data set as belonging to Grommet Manufacturing Division

*D* identifies the Development Department

*####* is a project number identifying the project.

The naming standards for ABC International and Grommet Manufacturing conflict. Eventually, Grommet Manufacturing data sets will be renamed to conform to the ABC International standard.

## Decisions

### ABC International Implementation Decisions for Grommet Manufacturing

- All allocation requests will be tracked for the Development Department.
- SG-Control applications will be created to maintain individual totals for each project. The application code will be constructed from the division code (GR) and the project number. The capacity of the SG-Control database will be expanded to 350 records to support the additional entries.
- Allocation totals will be maintained for permanent, temporary, and VSAM space. Allocation totals will not be maintained for DFHSM migrate and backup data sets.
- SG-Control will be modified to run with a combination of modes. Applications for Widget Manufacturing will be run in monitor mode and applications for Grommet Manufacturing will run in reject mode with a warning threshold set to 80%.
- The RESYNC function will be used to populate the database with the additional Grommet application information.
- The SG-Control Alphabetic Listing of Applications and Applications over Warning Threshold reports will be run weekly and the TSO BUDGET command will be used as needed to display up-to-the-minute space allocation information. The SG-Control Trend report will be run monthly to identify changes in DASD utilization.

## Instructions

### Step-by-Step Instructions for Implementing ABC International's Decisions

#### **Step 1** Update the database.

The database is updated using SGCMaint, the SG-Control utility program.

The following JCL is used to create an updated copy of the database.

**Figure 3-6 Scenario 2-Example JCL to Update the Database Using SGCMaint**

---

```
//Example JOB
//*
//*          UPDATE THE Database
//*
//INIT      EXEC PGM=SGCMaint
//STEPLIB   DD  DISP=SHR,DSN=XXXXXXXX.BBLINK
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  *
INIT FILE(UPDATE)
INIT FROMDB(XXXXXXXX.SPACEDB)
INIT DBNAME(XXXXXXXX.NEWDB)
INIT RECORDS(350)
INIT VOLUME(<DBVOL>)
/*
```

---

The following statement indicates that an updated version of the database is to be created. Database options and application information are copied from the existing database:

```
INIT FILE(UPDATE)
```

The following statement defines the name of the existing database:

```
INIT FROMDB(XXXXXXXX.SPACEDB)
```

The following three statements provide the data set name, physical size, and location of the updated database.

---

```
INIT DBNAME(XXXXXXXX.NEWDB)
INIT RECORDS(350)
INIT VOLUME(<DBVOL>)
```

---

## **Step 2 Modify the SG-Control filter list and rule list members.**

FLST and RLST parameters are used to select which DASD space allocation requests should be tracked and to replace application codes. In the following examples, the original members have been modified to also track space allocation requests for Grommet Manufacturing's Development Department. Application codes are constructed from the division code (GR) and the project number.

The following is an example of FLST parameters:

---

```
SET MODE=ACT
INC QUAL1=WG/                                TRACK REQUESTS FOR WIDGET
INC DSNAME=GROM.D/                          TRACK REQUESTS FOR GROMMET
DEVELOPMENT
```

---

The following is an example of RLST parameters:

---

```
SET NEWAPPL='?', OLDAPPL          USE DEFAULT APPLICATION
INC QUAL1=WG/
SET NEWAPPL='GR?', (QUAL, 2, 4)  BUILD NEW APPLICATION CODES
FOR GROMMET                      INC DSNAME=GROM.D/
DEVELOPMENT
```

---

Refresh the filter list and rule list members to ensure the parameters are syntactically correct.

### **Step 3 Start SG-Control.**

SVOS is the started task that activates SG-Control dynamically. If you are unfamiliar with the capabilities and functions of SVOS, it is important that you review operator services in the *MAINVIEW SRM User Guide and Reference*.

To activate the new database:

#### **3.A Shutdown SG-Control using the following command:**

```
P SVSGC
```

#### **3.B Restart SG-Control using the following command:**

```
S SVSGC
```

### **Step 4 Resynchronize the database.**

The RESYNC function of the SGCMAINT program can be used to add the Grommet development application information to the database quickly and efficiently.

ABC International decided to RESYNC the database with SG-Control active.

The RESYNC can be restricted (using VOLUME statements) to volumes that contain Grommet Manufacturing data sets.

**Figure 3-7 Scenario 2-Example JCL to Resynchronize the Database Using SGCMaint**

---

```
//*
//*      RESYNCHRONIZE THE DATABASE
//*
//RESYNC   EXEC  PGM=SGCMaint
//STEPLIB DD   DISP=SHR,DSN=XXXXXXXX.BBLINK
//SYSPRINT DD   SYSOUT=*
//SGCTrace DD   DUMMY
//SYSIN    DD   *
RESYNC  ACTIVEDEB
VOLUME INCLUDE(GROM*)
/*
```

---

**Step 5 Review and modify budget applications.**

All of Grommet's applications in the database were created using the default application. The names and budgets need to be tailored to meet user needs.

The TSO BUDGET command can be used to add, list, query, and change applications in the database. The command can be executed from a TSO session or a batch TMP.

ABC International decided to use a batch TMP to make mass changes to Grommet's applications.

Initially, all applications will have GROMMET in the name field, a permanent space budget of 120,000KB, a VSAM space budget of 60,000KB, a warning threshold of 80%, and an application mode of REJECT. Temporary space is not limited.

Application maintenance dialog boxes and views will be used later to review and tailor each application.

**Figure 3-8 Scenario 2-Example Mass Change JCL Using a Batch TMP**


---

```

//*
//*      MASS CHANGE BUDGET APPLICATIONS
//*
//UPDACCT  EXEC PGM=IKJEFT01
//STEPLIB  DD   DISP=SHR,DSN=XXXXXXXXX.BBLINK
//SYSPRINT DD   SYSOUT=*
//SYSTEM   DD   SYSOUT=*
//SYSIN    DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
        PROFILE NOPREFIX NOMSGID
        BUDGET (ACCT,EQ,'GR*') CHANGE NAME(GROMMET) +
        MAXPERM(120000) MAXVSAM(60000) WARN(80) +
        MAXTEMP(999999999) REJECT
/

```

---

The following figure shows output produced from the mass change JCL.

**Figure 3-9 Scenario 2-Example Output After Mass Change**


---

```

PROFILE NOPREFIX NOMSGID
READY
BUDGET GR*
GR1573
DATE-CREATED=2000/05/05 LAST-USED=2000/05/09 WARN=80% INDEX=7 REJECT
NAME='GROMMET'
  TYPE=PERM  CUR=34219.00k  MAX= 120000.68k  HWM=34219.11k
  TYPE=TEMP  CUR= 0.00k    MAX=99999999.36k  HWM= 0.00k
  TYPE=VSAM  CUR= 94.67k   MAX= 60000.47k   HWM= 94.43k
  TYPE=HSM   CUR= 0.00k    HWM= 0.00k
GR1578
DATE-CREATED=2000/05/05 LAST-USED=2000/05/09 WARN=80% INDEX=8 REJECT
NAME = 'GROMMET'
  TYPE=PERM  CUR=86219.79k  MAX= 120000.68k  HWM=86219.12k
  TYPE=TEMP  CUR= 0.00k    MAX=99999999.36k  HWM= 0.00k
  TYPE=VSAM  CUR= 1494.54k  MAX= 60000.47k   HWM= 1494.09k
  TYPE=HSM   CUR= 0.00k    HWM= 0.04k
GR2132
DATE-CREATED=2000/05/05 LAST-USED=2000/05/09 WARN=80% INDEX=9 REJECT
NAME = 'GROMMET'
  TYPE=PERM  CUR=869719.22k  MAX= 120000.68k  HWM=869719.46k
  TYPE=TEMP  CUR= 0.00k    MAX=99999999.12k  HWM= 0.00k
  TYPE=VSAM  CUR= 68794.09k  MAX= 60000.34k   HWM= 68794.98k
  TYPE=HSM   CUR= 0.00k    HWM= 0.00k

```

---

## Scenario 3

ABC International now decides to use SG-Control to monitor the amount of space used on DASD by DFHSM for each record.

The implementation will be simple since the application construction is limited to the first and second qualifiers of the data set name.

**Note:** This method of tracking DFHSM usage by application is restricted to using the first two qualifiers of the data set name to build the application code.

The prefix used for the DFHSM migrate and backup data sets is HSM.

### Decisions

ABC International Implementation Decisions for Adding DFHSM Tracking

- Allocation totals for DFHSM will be kept separately within the application.
- The RESYNC function will be used to add the DFHSM totals to the database.
- The TSO BUDGET command will be used as needed to display DFHSM usage by application.

### Instructions

Step-by-Step Instructions for Implementing ABC International's Decisions

#### Step 1 Update the Database

The database is updated using SGCMAINT, the SG-Control utility program. The following JCL is used to create an updated copy of the database.

**Figure 3-10 Scenario 3-Example JCL to Update the Database Using SGCMaint**

---

```
//Example JOB
//*
//*   UPDATE THE DATABASE
//*
//UPDDB   EXEC   PGM=SGCMaint
//STEPLIB DD     DISP=SHR,DSN=xxxxxxxx.BBLINK
//SYSPRINT DD    SYSOUT=*
//SYSIN   DD     *
INIT FILE(UPDATE)
INIT FROMDB(XXXXXXXX.SPACEDB)
INIT DBNAME(XXXXXXXX.NEWDB)
INIT VOLUME(<DBVOL>)
INIT HSMTRACK(YES)           <-----HSM tracking is
INIT HBACKUPPREFIX(HSM)     <-----implemented with
INIT HMIGRATEPREFIX(HSM)    <-----these three parameters
/*
```

---

The following three statements provide the data set name, physical size, and location of the updated database.

---

```
INIT FILE(UPDATE)
INIT FROMDB(XXXXXXXX.SPACEDB)
INIT DBNAME(XXXXXXXX.NEWDB)
INIT VOLUME(<DBVOL>)
```

---

The following statement enables DFHSM tracking. During SG-Control processing, QUAL1 and QUAL2 parameters are replaced automatically with values from the original data set name instead of the QUAL1 and QUAL2 parameters of the DFHSM data set.

```
INIT HSMTRACK(YES)
```

The following statement tells the SG-Control system that HSM is the prefix used for DFHSM backup data sets:

```
INIT HBACKUPPREFIX(HSM)
```

The following statement tells the SG-Control system that HSM is the prefix used for DFHSM migrate data sets:

```
INIT HMIGRATEPREFIX(HSM)
```

**Step 2 Start SG-Control.**

SVOS is the started task that activates SG-Control dynamically. If you are unfamiliar with the capabilities and functions of SVOS, it is important that you review operator services in the *MAINVIEW SRM User Guide and Reference*.

To activate the new database:

**2.A** Shutdown SG-Control using the following command:

```
P SVSGC
```

**2.B** Restart SG-Control using the following command:

```
S SVSGC
```

**Step 3 Resynchronize the database.**

By restricting the RESYNC function to the DFHSM volsers, the database can be updated with the DFHSM totals.

**Figure 3-11 Scenario 3-Example JCL to Resynchronize the Database Using SGC**

---

```

MAINT
//*
//*          RESYNCHRONIZE THE DATABASE
//*
//RESYNC    EXEC    PGM=SGCMAINT
//STEPLIB  DD      DISP=SHR,DSN=XXXXXXXX.BBLINK
//SYSPRINT DD      SYSOUT=*
//SYSIN    DD      *
RESYNC ACTIVEDB
VOLUME INCLUDE(HSM*)
/*

```

---

**Step 4 Review allocation totals with TSO BUDGET command.**

On the TSO command line, type **TSO BUDGET GR2132** and the following example output is produced:

**Figure 3-12 Scenario 3-Example Output from the Budget Command**


---

SGC3203	DATE-CREATED=2000/01/14	LAST-USED=2000/01/21	WARN=0%	INDEX=8
SGC3205	TYPE=PERM	CUR=149536.00K	MAX=0.00K	HWM=149978.69K
SGC3205	TYPE=TEMP	CUR= 0.00K	MAX=0.00K	HWM= 0.00K
SGC3205	TYPE=VSAM	CUR= 58268.74K	MAX=0.00K	HWM= 58268.74K
SGC3205	TYPE=HSM	CUR=244972.20K		HWM=244972.20K

---

## Scenario 4

XYZ Financial Services decides to implement SG-Control company-wide. The product will be used to monitor and control the space used by the production systems and by the development staff.

The XYZ Financial Services data set naming standard is

*AAAAA.BBBCCC.xxxxxxxxxx*

where

*AAAAA* identifies the owner of the data set. The owner can be PROD for production, TEST for product testing, or a TSO user ID.

*BBB* identifies the application. The application can be ILA for Installment Loans, DDA for Demand Deposit, and TDA for time deposits.

*CCC* identifies the version number.

### Decisions

#### XYZ Financial Services Implementation Decisions

- All allocation requests will be tracked.
- SG-Control applications will be created to maintain individual totals for production, product testing, systems programming, and each TSO user. Initially, 800 applications will be needed. Additional records will be provided to allow for expansion.
- Allocation totals will be combined for permanent and VSAM space. Temporary and DFHSM allocations will not be tracked.
- SG-Control will be implemented with a combination of modes. The production application (PROD) and the product testing application (TEST) will run in monitor mode. The applications for each version of an application will run in warning mode. The applications for TSO users will run in reject mode with a warning threshold set to 80%.

- The RESYNC function will be used to populate the database with application information.
- The SG-Control Alphabetic Listing of Applications and Applications over Warning Threshold reports will be run weekly and the TSO BUDGET command will be used as needed to display up-to-the-minute space allocation information. The MAINVIEW SRM ISPF interface will be used to identify changes in DASD utilization.

### Instructions

Step-by-Step Instructions for Implementing XYZ Financial Services Decisions

#### Step 1 Initialize the database.

The database is initialized using SGCMAINT, the SG-Control utility program. The following JCL is used to initialize XYZ Financial Services' database.

**Figure 3-13 Scenario 4-Example JCL Using SGCMAINT for Database Initialization**

---

```
//Example JOB
//*
//*      INITIALIZE THE DATABASE
//*
//INIT      EXEC  PGM=SGCMAINT
//STEPLIB   DD    DISP=SHR,DSN=XXXXXXXX.BBLINK
//SYSPRINT  DD    SYSOUT=*
//SYSIN     DD    *
INIT FILE(NEW)
INIT DBNAME(XXXXXXXX.SPACEDB)
INIT RECORDS(1000) RECVOLS(50)
INIT VOLUME(<DBVOL>)
INIT LEVEL(1) FIELD1(QUAL1) RECALC
INIT LEVEL(2) FIELD1(QUAL2) RECALC
INIT MODE(MONITOR)
INIT MAXKEY(8)
/
```

---

The following statement indicates a new database is to be created:

```
INIT FILE(NEW)
```

The following three statements provide the data set name, the capacity, and the location of the new database. The RECVOLS parameter controls the number of volume totals that can be maintained for each record. The number specified for RECVOLS should equal the maximum number of volumes that will contain data associated with any one application. XYZ Financial Services has included an allowance for growth.

---

```
INIT DBNAME(XXXXXXXXX.SPACEDB)
INIT RECORDS(1000) RECVOLS(50)
INIT VOLUME(<DBVOL>)
```

---

The following two statements define the default application code construction rules in the system options. XYZ Financial Services will use two levels; that is, up to two application codes can be associated with each space allocation request. The first level will be constructed from the first qualifier of the data set name and the second level will be constructed from the second qualifier of the data set name.

---

```
INIT LEVEL(1) FIELD1(QUAL1) RECALC
INIT LEVEL(2) FIELD1(QUAL2) RECALC
```

---

The following statement sets the mode system option:

```
INIT MODE(MONITOR)
```

In the following statement, the MAXKEY parameter defines the length of the longest application code used in the database:

```
INIT MAXKEY(8)
```

After SGCMAINT has allocated and formatted the new database successfully, the next step is to add a default application.

## **Step 2 Add a default application.**

The TSO BUDGET command can be used to add, list, query, and change applications in the database. The command can be executed from a TSO session or a batch TMP. The following example shows how the default application is added to the database using a batch TMP:

**Figure 3-14 Scenario 4-Example Batch TMP for Adding a Default Application**

---

```

// *
// *      ADD THE DEFAULT APPLICATION TO THE DATABASE
// *
// ADDDEF      EXEC   PGM=IKJEFT01 , DYNAMNBR=10
// STEPLIB    DD      DISP=SHR , DSN=XXXXXXXX.BBLINK
// SYSPRINT    DD      SYSOUT= *
// SYSTEM     DD      SYSOUT= *
// SYSIN      DD      SYSOUT= *
// SYSTSPRT   DD      SYSOUT= *
// SYSTSIN    DD      *
PROFILE NOPREFIX
ALLOC FILE(SGCDB) DSN(XXXXXXXX.SPACEDB) SHR
BUDGET 'CONTROL' ADD MAXPERM(200000) NOTEMP +
PVSAM NAME('TSO USER') NOHSM
/ *

```

---

Since SG-Control has not been started, the database must be allocated. The TSO BUDGET command is then used to add a default application to the database. An application code of CONTROL identifies the application as being the default application. The values specified for the name, application options, and budget are included in the default application. SG-Control is being implemented in a combination of modes. The permanent allocation budget amount has been set to 200000 to provide a figure that can be used by the MAINVIEW SRM ISPF interface for percentage utilization calculations and reports. The NOTEMP and NOHSM options direct SG-Control to ignore all temporary space and DFHSM allocations and the PVSAM option directs SG-Control to include the VSAM space allocation totals with the permanent space allocation totals.

Individual application information will be customized later in the implementation process.

**Step 3 Build the SG-Control filter list and rule list members.**

FLST and RLST parameters are used to select which DASD space allocation requests should be tracked and to replace the application codes. In this example, system data sets are assigned an application code of SYSTEMS, all allocation requests will be tracked at level 1, and only allocations with a second-level qualifier starting with ILA, DDA, or TDA will be tracked at Level 2.

The following example shows FLST parameters:

SET MODE=ACT INC LEVEL=1 INC LEVEL=2,QUAL2=(ILA,TDA,DDA/)	INCLUDE ALL LEVEL1 TRACK LEVEL 2 REQUESTS FOR ILA, TDA, AND DDA
---	--

The following example shows RLST parameters:

SET NEWAPPL='SYSTEMS' INC LEVEL=1,QUAL1=SYS/ SET NEWAPPL='?',OLDAPPL INC LEVEL=1 INC LEVEL=2,QUAL2=(ILA,TDA,DDA/)	ASSIGN SYSTEM DATA SET TO APPLICATION SYSTEMS USE DEFAULT APPLICATION CODES FOR REMAINING REMAINING REQUESTS
---	---

Refresh the filter list and rule list members to ensure the parameters are syntactically correct.

#### **Step 4 Start SG-Control.**

SVOS is the started task that activates SG-Control dynamically. If you are unfamiliar with the capabilities and functions of SVOS, it is important that you review operator services in the *MAINVIEW SRM User Guide and Reference*.

To activate the new database:

##### **4.A Shutdown SG-Control using the following command:**

```
P SVSGC
```

##### **4.B Restart SG-Control using the following command:**

```
S SVSGC
```

#### **Step 5 Resynchronize the database.**

The RESYNC function of the SGCMAINT program can be used to populate the database with application information quickly and efficiently. The database can be resynchronized with SG-Control active or inactive.

XYZ Financial Services decides to RESYNC the database with SG-Control active.

The RESYNC will be run against all mounted volumes. Audit reporting has been selected with the AUDIT option. The VOLSER value will trigger the generation of messages identifying the volume serial of the packs being processed. DSNAMES will trigger the generation of messages identifying the data sets being processed. The following example shows JCL used to resynchronize the database with SG-Control active:

**Figure 3-15 Scenario 4-Example JCL to Resynchronize the Database with SG-Control Active**

---

```
//*
//*      RESYNCHRONIZE THE DATABASE
//*
//RESYNC  EXEC  PGM=SGCMAINT
//STEPLIB DD    DISP=SHR,DSN=XXXXXXXX.BBLINK
//SYSPRINT DD   SYSOUT=*
//BUDTRACE DD   DUMMY
//SYSIN    DD    *
RESYNC AUDIT(DSNAME,VOLSER) ACTIVEDB
/*
```

---

**Step 6 Review and modify budget applications.**

All the applications in the database were created using the default application. The name, application level options, and budgets need to be customized to meet the needs of XYZ Financial Services.

The TSO BUDGET command can be used to add, list, query, and change applications in the database. The command can be executed from a TSO session or a batch TMP. Budget applications also can be customized using the MAINVIEW SRM ISPF Interface.

XYZ Financial Services decides to use a batch TMP to list and customize the budget applications. The following figure shows batch TMP used to list budget applications.

**Figure 3-16 Scenario 4-Example Batch TMP to List Budget Applications**


---

```

//*
//*      LIST THE BUDGET APPLICATIONS
//*
//LISTACCT EXEC PGM=IKJEFT01
//STEPLIB DD DISP=SHR,DSN=XXXXXXXXX.BBLINK
//SYSPRINT DD SYSOUT=*
//SYSTEM DD SYSOUT=*
//SYSIN DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
PROFILE NOPREFIX NOMSGID
BUDGET PROD
BUDGET TEST
BUDGET SYSTEMS
BUDGET (ACCT,EQ,'TDA*')
BUDGET (ACCT,EQ,'ILA*')
BUDGET (ACCT,EQ,'DDA*')
BUDGET (NAME,EQ,'TSO*')
/*

```

---

The following figure shows the output produced from the batch TMP.

**Figure 3-17 Scenario 4-Example Output Produced from Batch TMP**


---

```

PROFILE NOPREFIX NOMSGID
READY
BUDGET PROD
PROD
DATE-CREATED=2000/05/05 LAST-USED=2000/05/05 WARN=0% INDEX=2
NAME ='TSO USER'
TYPE=PERM CUR=7630219.12k MAX=200000.78k HWM=7630219.11k
TYPE=VSAM INCLUDED IN PERM COUNT

READY
BUDGET TEST
TEST
NAME ='TSO USER'
DATE-CREATED=2000/05/05 LAST-USED=2000/05/05 WARN=0% INDEX=3
TYPE=PERM CUR= 12562.45k MAX=200000.00k HWM=12562.32k
TYPE=VSAM INCLUDED IN PERM COUNT

READY
BUDGET SYSTEMS
SYSTEMS
NAME ='TSO USER'
DATE-CREATED=2000/05/05 LAST-USED=2000/05/05 WARN=0% INDEX=4
NAME ='TESTING'
TYPE=PERM CUR=1893262.11k MAX=200000.98k HWM=1893262.24k
TYPE=VSAM INCLUDED IN PERM COUNT

```

---

The BUDGET command is used to customize the name of the PROD, TEST, and SYSTEMS applications and all applications with application codes starting with TDA, ILA, or DDA. The name field of the remaining TSO user applications contains the TSO user ID.

The production application (PROD) and the product testing application (TEST) run in monitor mode.

The TDA, ILA, and DDA applications run in warning mode. The permanent allocation budget amount is customized in each of these applications.

TSO users run in reject mode. Initially, each user is given the same permanent space budget. The WARN parameter is used to set the warning threshold percentage. WARN(80) will trigger the generation of a warning message when more than 80% of a TSO user's space budget is used.

If necessary, the permanent space budgets can be customized further by the DASD Space Administrator. The following example shows batch TMP used to customize budget applications:

**Figure 3-18 Scenario 4-Example Batch TMP to Customize Budget Applications**

---

```

/*
/*      CUSTOMIZE THE BUDGET APPLICATIONS
/*
//UPDACCT EXEC PGM=IKJEFT01,DYNAMNBR=10
//STEPLIB      DD          DISP=SHR,DSN=XXXXXXXX.BBLINK
//SYSPRINT     DD          SYSOUT=*
//SYSTEM      DD          SYSOUT=*
//SYSIN        DD          SYSOUT=*
//SYSTSPRT    DD          SYSOUT=*
//SYSTSIN      DD          *
PROFILE NOPREFIX
BUDGET PROD          CHANGE  NAME( PRODUCTION)
BUDGET TEST          CHANGE  NAME( TESTING)
BUDGET SYSTEMS      CHANGE  NAME( SYSTEMS)
BUDGET (ACCT,EQ,'TDA*') CHANGE  NAME( TIME) WARNING  +
                    MAXPERM( 400000)
BUDGET (ACCT,EQ,'ILA*') CHANGE  NAME( INSTALLMENTS) +
                    WARNING MAXPERM( 500000)
BUDGET (ACCT,EQ,'DDA*') CHANGE  NAME( DEMAND)          +
                    WARNING MAXPERM( 9000000)
BUDGET (NAME,EQ,'TSO*') CHANGE  MAXPERM(100000)        +
                    WARN( 80) REJECT
/*

```

---

The following figure shows output produced after customization.

**Figure 3-19 Scenario 4-Example Output After Customization**

---

```
PROFILE NOPREFIX NOMSGID
READY
BUDGET PROD          CHANGE  NAME( PRODUCTION)
PROD
DATE-CREATED=2000/05/05 LAST-USED=2000/05/05 WARN=0% INDEX=2
NAME = ' PRODUCTION'
  TYPE=PERM  CUR=7630219.24k  MAX=200000.11k  HWM=7630219.67k
  TYPE=VSAM  INCLUDED IN PERM COUNT

READY
BUDGET TEST          CHANGE  NAME( TESTING)
TEST
NAME = ' TESTING'
DATE-CREATED=2000/05/05 LAST-USED=2000/05/05 WARN=0% INDEX=3
  TYPE=PERM  CUR= 12562.65k  MAX=200000.43k  HWM=12562.07k
  TYPE=VSAM  INCLUDED IN PERM COUNT

READY
BUDGET SYSTEMS      CHANGE  NAME( SYSTEMS)
SYSTEMS
NAME = ' SYSTEMS'
DATE-CREATED=2000/05/05 LAST-USED=2000/05/05 WARN=0% INDEX=4
  TYPE=PERM  CUR=1893262.38k  MAX=200000.47k  HWM=1893262.15k
  TYPE=VSAM  INCLUDED IN PERM COUNT
```

---

---

---

# Chapter 4 Managing SG-Control Applications

This chapter explains how to manage the SG-Control applications and create reports from the StorageGUARD data collector database.

Overview . . . . .	4-1
Maintaining Individual Applications . . . . .	4-3
Maintaining Multiple Applications . . . . .	4-8
WHERE Command . . . . .	4-12
Viewing the Application List . . . . .	4-14
Displaying Data Set Information . . . . .	4-17

## Overview

The tools you need to manage the SG-Control database are accessed from the EZSRM menu.

To access application maintenance views:

» From the EZSRM Menu, select **SG-Control Applications**.

The EZSRMSGC Menu displays, as shown in Figure 4-1.

**Figure 4-1 EZSRMSGC Menu**

```

14MAY2001 09:24:29 ----- INFORMATION DISPLAY -----
COMMAND ==>                                     SCROLL ==> PAGE
CURR WIN ==> 1           ALT WIN ==>
>W1 =EZSRMSGC=====SJSH=====14MAY2001==09:24:29====MVSMSGC=D====1
           EZSRMSGC  Menu

SG-Control          +-----+   SRM Administration
                   |         |   |
                   | Place cursor on |   |
                   | menu item and  |   |
                   | press ENTER    |   |
                   +-----+   |   |
. Application Maint  |         |   | > Parmlib Members
. Application Mass Upd |         |   | . Functions
. Application List    |         |   | . SRM Component Status
. Data Set Information |         |   | . MVSMSGC View List
                   |         |   | . MVSMSGC Batch Reports
                   |         |   | . Return....

```

The options available from the EZSRMSGC menu are:

**Table 4-1 MAINVIEW SRM Application Maintenance Options**

Option	Description	View Name
Application Maint	allows you to add new applications, change existing applications, or mark for deletion <i>individual</i> applications	APPLDET
Application Mass Update	allows you to change multiple applications	APPLGRPD
Application List	allows you to review application codes and format and sort allocation information	APPLTAB APPLDTL
Data Set Information	allows you to query applications by data set name	APPLDSNT

# Maintaining Individual Applications

Application maintenance is used to add new, change existing, or mark for deletion individual applications. Only fields in which you type data are affected. This panel builds and executes the TSO BUDGET command.

- Step 1** From the EZSRMSGC Menu, select **Application Maint.** The APPLDET dialog box displays, as shown in Figure 4-2.

**Figure 4-2 APPLDET Dialog Box**

```

30MAY2001 12:37:10 ----- INFORMATION DISPLAY -----
COMMAND ==>                                     SCROLL ==> PAGE
CURR WIN ==> 1           ALT WIN ==>
W1 =APPLDET=====SJSH=====*=====30MAY2001==12:37:10====MVSMSGC=D====1
<== Type S to process request

Application Add/Change.....

Action (Add/Change).....

Application Name.....

Model Application.....
    
```

- Step 2** Type an action, application code, and replacement values based on the information described in the following table.

**Table 4-2 Application Maintenance Valid Field Entries**

Field	Entry
Action	Type <b>A</b> (add) or <b>C</b> (change).
Application Code	Type a new (for add) or existing (for change) application code.
Model Application	Type an existing application code in the database to use as a model for the addition of a new application.

**Step 3** Type **S** in the field indicated to process the request. The APPLTAB view is displayed.

**Figure 4-3 Application Update Dialog Box**

```

30MAY2001 12:38:55 ----- INFORMATION DISPLAY -----
COMMAND ==>
CURR WIN ==> 1          ALT WIN ==>
>W1 =APPLDTL=====SJSH=====*=====30MAY2001==12:38:55===MVSMSGC=D====1
<== Type A to Add or          Create Date.....
      U to Update              LastUsed Date.....
                                Application Status
Application Name.... PAYROLL    User Field1.....
User Name.....                 User Field2.....
                                User Field3.....
Permanent Budget....          Track Temp..... Y
Permanent Allocated.          Track VSAM..... Y
Permanent H-W-M.....          Track HSM..... Y
Temporary Budget....          Add Temp..... N
Temporary Allocated.          Add VSAM..... N
Temporary H-W-M.....          Add HSM..... N
VSAM Budget.....              Reset H-W-M..... N
VSAM Allocated.....           Mark Delete..... N
VSAM H-W-M.....               Appl Mode.....
HSM Allocated.....            Warning Threshold. 0
HSM H-W-M.....                Volume Count..... 0
    
```

**Step 4** Make changes to the valid entry fields as described in Figure 4-3 and type **A** to Add or **U** to Update in the indicated field.

**Table 4-3 Application Maintenance Input Fields (Part 1 of 4)**

Field	Entry
Application Name	Application name in database. This is the full application name. <i>Do not use</i> the following characters in an application name: ( ) \ [ ] & < > and imbedded spaces
User Name	You can define a name up to 20 characters long. The name may contain blanks, as well as upper and lowercase letters. To update the applications by clearing the user name to spaces, enter the literal BLANK or blank in the user name. The following are valid entries: User Name.... Data Center User Name.... blank

**Table 4-3 Application Maintenance Input Fields (Part 2 of 4)**

<b>Field</b>	<b>Entry</b>
Permanent Budget	Contains the maximum amount of space allowed for permanent data sets assigned to this application code. To update this field, enter up to 9 characters, such as: Permanent Budget.. 2.5m Permanent Budget.. 4096 Permanent Budget.. 6G The maximum budget supported is 86 exabytes.
Permanent Allocated	Contains the total amount of space allocated on DASD for permanent data sets assigned to this application code.
Permanent H-W-M	Contains the largest amount of space used or reserved for permanent data sets assigned to this application code since the high-water marks were last reset.
Temporary Budget	Contains the maximum amount of space allowed for temporary data sets assigned to this application code. To update this field, enter up to 9 characters, such as: Permanent Budget.. 2.5m Permanent Budget.. 4096 Permanent Budget.. 6G The maximum budget supported is 86 exabytes.
Temporary Allocated	Contains the total amount of space allocated on DASD for temporary data sets assigned to this application code.
Temporary H-W-M	Contains the largest amount of space used or reserved for temporary data sets assigned to this application code since the high-water marks were last reset.
VSAM Budget	Contains the maximum amount of space allowed for VSAM data sets assigned to this application code. To update this field, enter up to 9 characters, such as: Permanent Budget.. 2.5m Permanent Budget.. 4096 Permanent Budget.. 6G The maximum budget supported is 86 exabytes.
VSAM Allocated	Contains the total amount of space allocated on DASD for VSAM data sets assigned to this application code.
VSAM H-W-M	Contains the largest amount of space used or reserved for VSAM data sets assigned to this application code since the high-water marks were last reset.
HSM Allocated	Contains the total amount of space allocated on DASD for HSM data sets assigned to this application code.

**Table 4-3 Application Maintenance Input Fields (Part 3 of 4)**

<b>Field</b>	<b>Entry</b>
HSM H-W-M	Contains the largest amount of space used or reserved for HSM data sets assigned to this application code since the high-water marks were last reset.
Create Date	Contains the date that the application was created.
LastUsed Date	Contains the date that the application was used.
Application Status	Indicates the status of the current application. The application status types are: <b>MDEL</b> Application has been manually flagged for deletion. The next time that the database is copied, this application will be deleted. <b>DEL</b> Application has been automatically flagged for deletion. This application was created, but never updated. Since no activity has taken place in the application, it will be deleted the next time the database is copied. <b>ACTV</b> Application is active.
User Field1	You can define a field up to 8 characters long. The field may contain blanks, as well as upper and lowercase letters. To update the applications by clearing the user name to spaces, enter the literal BLANK or blank in the user name. The following are valid entries: User Field1.. Day crew User Field1..blank
User Field2	See User Field1 (10 characters)
User Field3	See User Field1 (10 characters)
Track Temporary	Type <b>Y</b> to set the flag indicating that temporary space is to be tracked. <b>N</b> removes the flag indicating that temporary space is to be tracked.
Track VSAM	Type <b>Y</b> to set the flag indicating that VSAM space is to be tracked. <b>N</b> removes the flag indicating that VSAM space is to be tracked.
Track HSM	Type <b>Y</b> to set the flag indicating that HSM space is to be tracked. <b>N</b> removes the flag indicating that HSM space is to be tracked.
Add Temporary	Type <b>Y</b> if temporary space is to be accounted for as part of the permanent space budget. Type <b>N</b> if temporary space is not to be accounted for as part of the permanent space budget. Unless <b>N</b> has been specified for Track Temporary, entering an <b>N</b> will cause temporary space to be tracked separately.

**Table 4-3 Application Maintenance Input Fields (Part 4 of 4)**

<b>Field</b>	<b>Entry</b>
Add VSAM	Type <b>Y</b> if VSAM space is to be accounted for as part of the permanent space budget. Type <b>N</b> if VSAM space is not to be accounted for as part of the permanent space budget. Unless <b>N</b> has been specified for Track VSAM, typing an <b>N</b> will cause VSAM space to be tracked separately.
Add HSM	Type <b>Y</b> if HSM space is to be accounted for as part of the permanent space budget. Type <b>N</b> if HSM space is not to be accounted for as part of the permanent space budget. Unless <b>N</b> has been specified for Track HSM, typing an <b>N</b> will cause HSM space to be tracked separately.
Reset H-W-M	Type <b>Y</b> to cause all high-water marks to be reset to current allocation amounts. This function is relevant to CHANGE only.
Mark Delete	Type <b>Y</b> to cause the application to be deleted the next time the database is copied. <b>N</b> resets the flag.
Application Mode	Indicate how you want allocations treated for this Application. The modes are: <b>Monitor</b> tracks space allocations and deallocations as they occur allowing up-to-the-minute analysis of DASD space usage <b>Warning</b> generates a message if the current allocation exceeds the budget amount <b>Reject</b> rejects the allocation if the current allocation exceeds the budget amount <b>Default</b> sets the application mode back to the default value
Warning threshold	The percentage of the budget that can be used by this application before a warning message is issued. It can have any value up to 100, such as: Warning Threshold. 75 Warning Threshold. 100
Volume Count	Contains the number of volumes that contain at least one data set that is included in the allocation amounts for this application.

**Tip:** Budgets are used by StorageGUARD views as the basis for percentage usage calculation and graphic and tabular displays. BMC Software recommends that a budget be specified for all applications running in monitor mode. This budget will be used only for StorageGUARD view calculation reporting purposes.

# Maintaining Multiple Applications

The application Group Maintenance panel is used to change multiple applications. The APPLGRPD view allows you to enter values for the fields you want to update. Fields left blank will not be updated. For example, if you want to update only the permanent space budget and the temporary space budget, then enter the new values in these two fields and leave all the other fields blank.

When you apply the updates to the selected applications on the next page (APPLMUP), only the values in the permanent budget and the temporary budget will be changed. The updates will not be applied until you specify UPDATE on the APPLMUP view.

Use the following steps to process group maintenance/mass updates:

**Step 1** From the EZSRMSGC Menu, select **Application Mass Update**.

The APPLGRPD dialog box is displayed, as shown in Figure 4-4.

**Figure 4-4 Application Group Maintenance Update Panel**

```

30MAY2001 13:15:43 ----- INFORMATION DISPLAY -----
COMMAND ==>                                SCROLL ==> PAGE
CURR WIN ==> 1          ALT WIN ==>
>W1 =APPLGRPD=====SJSJ=====*=====30MAY2001==13:15:43====MVSMSGC=D====1
<== Type N for next page

Group Maintenance.....          Track Temporary...
                                   Track VSAM.....
Application Mask.....          Track HSM.....
User Name.....
User Field1.....              Add Temporary....
User Field2.....              Add VSAM.....
User Field3.....              Add HSM.....

Permanent Budget.....
Temporary Budget.....        Reset H-W-M.....
VSAM Budget.....

Application Mode.....
Warning Threshold.....        Mark for Deletion.
    
```

**Step 2** Complete replacement values based on the information described in Figure 4-4.

**Table 4-4 Application Group Maintenance Valid Field Entries (Part 1 of 3)**

<b>Field</b>	<b>Entry</b>
Application Mask	<p>Application mask used to filter the list of applications against which the updates can be applied. Leave blank to display a complete list of applications.</p> <p>After the application list is displayed, you will have an opportunity to filter the applications based on the values in any of the fields in the applications.</p> <p>Application Mask.. OP*</p> <p>Application Mask..</p>
User Name	<p>You can define a name up to 20 characters long. The name may contain blanks, as well as upper and lowercase letters.</p> <p>To update the applications by clearing the user name to spaces, enter the literal BLANK or blank in the user name. The following are valid entries:</p> <p>User Name.... Data Center</p> <p>User Name.... blank</p>
User Data 1	<p>You can define a field up to 10 characters long. The field may contain blanks, as well as upper and lowercase letters.</p> <p>To update the applications by clearing the user name to spaces, enter the literal BLANK or blank in the user name. The following are valid entries:</p> <p>User Field1.. Day crew</p> <p>User Field1..blank</p>
User Data 2	See User Data 1
User Data 3	See User Data 1
Permanent Budget	<p>Enter the maximum amount of space allowed for permanent data sets assigned to this application code. Budgets can be entered up to 9 characters. For example:</p> <p>Permanent Budget.. 2.5m</p> <p>Permanent Budget.. 4096</p> <p>Permanent Budget.. 6G</p> <p>The maximum budget supported is 86 exabytes.</p>
Temporary Budget	<p>Enter the maximum amount of space allowed for temporary data sets assigned to this application code. The amount may be expressed in bytes or units of kilobytes: M (megabytes), G (gigabytes), P (petabytes), T (terabytes) or E (exabytes). One kilobyte is 1024 bytes. Budgets can be entered up to 9 characters. For example:</p> <p>Temporary Budget.. 2.5m</p> <p>Temporary Budget.. 4096</p> <p>Temporary Budget.. 6G</p> <p>The maximum budget supported is 86 exabytes.</p>

**Table 4-4 Application Group Maintenance Valid Field Entries (Part 2 of 3)**

Field	Entry
VSAM Budget	<p>Enter the maximum amount of space allowed for VSAM data sets assigned to this application code. The amount may be expressed in bytes or units of kilobytes: M (megabytes), G (gigabytes), P (petabytes), T (terabytes) or E (exabytes). One kilobyte is 1024 bytes. Budgets can be entered up to 9 characters. For example:</p> <p>VSAM Budget..... 2.5m                      VSAM Budget..... 4096                      VSAM Budget..... 6G</p> <p>The maximum budget supported is 86 exabytes.</p>
Application Mode	<p>Indicate how you want allocations treated for this Application. The modes are:</p> <p><b>Monitor</b>                      tracks space allocations and deallocations as they occur allowing up-to-the-minute analysis of DASD space usage</p> <p><b>Warning</b>                      generates a message if the current allocation exceeds the budget amount</p> <p><b>Reject</b>                      rejects the allocation if the current allocation exceeds the budget amount</p> <p><b>Default</b>                      sets the application mode back to the default value</p>
Warning threshold	<p>Specify the percentage of the budget that can be used by this application before a warning message is issued up to 100, such as:</p> <p>Warning Threshold. 75                      Warning Threshold. 100</p>
Track Temporary	<p>Type <b>Y</b> to set the flag indicating that temporary space is to be tracked. <b>N</b> removes the flag indicating that temporary space is to be tracked.</p>
Track VSAM	<p>Type <b>Y</b> to set the flag indicating that VSAM space is to be tracked. <b>N</b> removes the flag indicating that VSAM space is to be tracked.</p>
Track HSM	<p>Type <b>Y</b> to set the flag indicating that HSM space is to be tracked. <b>N</b> removes the flag indicating that HSM space is to be tracked.</p>
Add Temporary	<p>Type <b>Y</b> if temporary space is to be accounted for as part of the permanent space budget. Type <b>N</b> if temporary space is not to be accounted for as part of the permanent space budget. Unless <b>N</b> has been specified for Track Temporary, entering an <b>N</b> will cause temporary space to be tracked separately.</p>
Add VSAM	<p>Type <b>Y</b> if VSAM space is to be accounted for as part of the permanent space budget. Type <b>N</b> if VSAM space is not to be accounted for as part of the permanent space budget. Unless <b>N</b> has been specified for Track VSAM, typing an <b>N</b> will cause VSAM space to be tracked separately.</p>

**Table 4-4 Application Group Maintenance Valid Field Entries (Part 3 of 3)**

Field	Entry
Add HSM	Type <b>Y</b> if HSM space is to be accounted for as part of the permanent space budget. Type <b>N</b> if HSM space is not to be accounted for as part of the permanent space budget. Unless <b>N</b> has been specified for Track HSM, typing an <b>N</b> will cause HSM space to be tracked separately.
Reset H-W-M	Type <b>Y</b> to cause all high-water marks to be reset to current allocation amounts. This function is relevant to CHANGE only.
Mark for Deletion	Type <b>Y</b> to cause the application to be deleted the next time the database is copied. <b>N</b> resets the flag.

**Tip:** Budgets are used by StorageGUARD views as the basis for percentage usage calculation and graphic and tabular displays. BMC Software recommends that a budget be specified for all applications running in monitor mode. This budget will be used only for StorageGUARD view calculation reporting purposes.

**Step 3** Type **N** in the indicated field to view the Application Mass Update Tabular View (APPLMUP) (Figure 4-5).

**Figure 4-5 Application Mass Update Tabular View**

```

30MAY2001 13:17:19 ----- INFORMATION DISPLAY -----
COMMAND ==>>>
CURR WIN ==>> 1          ALT WIN ==>>
>W1 =APPLMUP=====SJSH=====*=====30MAY2001==13:16:43====MVSMSGC=D=4203
CMD ApplicationName      Appl Permanent Permanent Permanent Temporary Tempo
--- -----            Stat Budget      Allocated H-W-M      Budget      Alloc
ACC09                    DEL    976.6Mi    113328    113328  512000000
ACC10                    DEL    976.6Mi    113328    113328  512000000
ACC11                    DEL    976.6Mi    113328    113328  512000000
ACC12                    DEL    976.6Mi    113328    113328  512000000
ACC13                    DEL    976.6Mi    113328    113328  512000000
ACC14                    DEL    976.6Mi    113328    113328  512000000
ACC15                    DEL    976.6Mi    113328    113328  512000000
ACC16                    DEL    976.6Mi    113328    113328  512000000
ACFUSER                  DEL    976.6Mi    283320    283320  512000000
ACFWK1                   ACTV   976.6Mi    9746208   10709496 512000000
ACFWK2                   ACTV   976.6Mi  255044664 255951288 512000000
ACFWK3                   ACTV   976.6Mi      2.6Gi      2.6Gi  512000000
ACFWRK                   ACTV   976.6Mi    1699920   1699920 512000000
ADSM                     DEL    976.6Mi    59667192  59667192 512000000
AES                       DEL    976.6Mi    35471664  35471664 512000000
AESCHL1                  DEL    976.6Mi     283320    283320  512000000
    
```

On the APPLMUP, you can select a list of the applications that you want to update with the values you entered in the previous view (APPLGRPD). See the actions in \*\*\*\* to see how to apply the update.

**Step 4** To update individual applications, type **U** in the line command to the left of the application(s) you want to update.

**Step 5** To update a number of applications with one command:

**5.A** Filter the application list based on the values in any of the fields by using the **WHERE** command. See “WHERE Command” on page 4-12

**5.B** Type **UPDATE \*** on the **COMMAND** line to update all applications that are currently displayed.

**Step 6** Type **UPDATE** on the **COMMAND** line to process the changes.

## WHERE Command

When you type the **WHERE** command on the **COMMAND** line, a pop-up panel is displayed on which you can specify conditions that filter data in a view without updating the data. The filter conditions are applied against the form and replace any existing filters. With the **WHERE** command, you can

- compare fields with values, or fields with other fields
- use the symbols **>**, **<**, **=**, **<=** and **>=**
- use **AND**, **OR**, **IN** and **BETWEEN** to combine conditions
- use asterisks and question marks in the mask

Use the following examples to help you learn how to use the **WHERE** command.

### Example

To define a filter condition that displays only the applications beginning with **J** or **M**, enter the **WHERE** command and:

```
(C06ACCT IN (j*,m*))
```

**Example**

To define a filter condition that displays only the values between 1024000 and 2048000 in the field with the element name of C06PERMM, enter the WHERE command and:

```
(C06PERMM BETWEEN 1024000 AND 2048000)
```

All space amount fields must specify the number of bytes. Budget amounts should not use units, such as K or M.

**Example**

To define a filter condition that displays only the applications beginning with OP which have values between 100000 and 500000 in the field with an element of C06TEMPM, enter the WHERE command and:

```
(C06ACCT = OP*) AND
```

```
(C06TEMPM BETWEEN 100000 AND 500000)
```

**Example**

To define a filter condition that displays only the applications where the permanent high water mark is greater than the permanent space allocated, enter the WHERE command and:

```
(C06PERMH > C06PERMC)
```

**Example**

To define a filter condition that displays only the applications that were created in October 2001, enter the WHERE command and

```
(C06CDATE = 2001?10*)
```

Where conditions should not contain special characters, like a slash or a colon. Use a question mark to mask these characters as shown in the example above.

## Viewing the Application List

The Application List is used to review application codes and format and sort allocation information. Individual applications can be selected for review and, if necessary, updated from this panel. The information on this panel is extracted from the SG-Control database at the time the Application List is selected.

» From the EZSRMSGC Menu, select **Application List**.

The APPLTAB view is displayed, as shown in Figure 4-6.

**Figure 4-6 APPLTAB View**

```

30MAY2001 13:18:20 ----- INFORMATION DISPLAY -----
COMMAND ==>
CURR WIN ==> 1          ALT WIN ==>
>W1 =APPLTAB=====SJSH=====*=====30MAY2001==13:17:59====MVSMSGC=D=4204
C ApplicationName      Appl Permanent Permanent Permanent Temporary Tempora
  -----            Stat Budget   Allocated H-W-M      Budget   Allocated
BAB304                ACTV   976.6Mi   1.7Gi    1.7Gi 512000000
BAB305                ACTV   976.6Mi   1.6Gi    1.6Gi 512000000
BAB306                ACTV   976.6Mi   1.3Gi    1.6Gi 512000000
BAB307                ACTV   976.6Mi   2.0Gi    2.0Gi 512000000
BAB308                ACTV   976.6Mi   1.9Gi    1.9Gi 512000000
BAB309                ACTV   976.6Mi   2.1Gi    2.1Gi 512000000
BAB310                ACTV   976.6Mi   1.7Gi    1.9Gi 512000000
BAB311                ACTV   976.6Mi   1.2Gi    1.3Gi 512000000
BAB312                ACTV   976.6Mi   1.4Gi    1.5Gi 512000000
BAB313                ACTV   976.6Mi   1.5Gi    1.6Gi 512000000
BAB314                ACTV   976.6Mi   1.8Gi    2.1Gi 512000000
BAB315                ACTV   976.6Mi   1.2Gi    1.4Gi 512000000
BAB316                ACTV   976.6Mi   1.7Gi    2.1Gi 512000000
BAB317                ACTV   976.6Mi   1.8Gi    1.8Gi 512000000
BAB318                ACTV   976.6Mi   1.9Gi    2.0Gi 512000000
BAB319                ACTV   976.6Mi   1.5Gi    1.5Gi 512000000

```

**Step 7** Perform the user actions described below to review application codes and, optionally, format and sort allocation information. All commands are typed on the command line.

- To locate application codes: type LOCATE followed by a partial or complete application code.
- To sort information displayed on the panel: type SORT with the operands in the following table.

The SORT command can be used to sort the application code and allocation information.

The SORT command can sort only one field at a time. Information can be sorted in ascending (A) or descending (D) order.

**Table 4-5 Valid Operands for the SORT Command**

<b>Operand</b>	<b>Result</b>
ACCT	Sorts by application code.
PERM	Sorts by current permanent allocation totals.
TEMP	Sorts by current temporary allocation totals.
VSAM	Sorts by current VSAM allocation totals.
HSM	Sorts by current HSM allocation totals.
HPERM	Sorts by the permanent allocation high-water marks.
HTEMP	Sorts by the temporary allocation high-water marks.
HVSAM	Sorts by the VSAM allocation high-water marks.
HHSM	Sorts by the DFHSM allocation high-water marks.
MPERM	Sorts by the permanent allocation budget.
MTEMP	Sorts by the temporary allocation budget.
MVSAM	Sorts by the VSAM allocation budget.

**Step 8** To return to the previous panel, either type **END** and press **Enter**, or press **PF3**.

## Selecting an Individual Application

Individual applications can be selected for review and update from the APPLTAB view. To select an application from the Application List:

**Step 1** In the APPLTAB view, place the cursor on the application name, press **Enter**.

**Step 2** Select **Detail View** and press **Enter**.

Detailed application information is displayed, as shown in Figure 4-7.

Figure 4-7 SG-Control Application Update Screen

```

30MAY2001 12:56:03 ----- INFORMATION DISPLAY -----
COMMAND ==>>                                SCROLL ==>> PAGE
CURR WIN ==>> 1          ALT WIN ==>>
>W1 =APPLTAB==APPLDTL==SJSH=====30MAY2001==12:55:45===MVSMSGC=D===1
  <== Type A to Add or          Create Date..... 2001/03/05
    U to Update                  LastUsed Date.... 2001/05/30
                                  Application Status ACTV
Application Name... PAYROLL      User Field1.....
User Name.....                  User Field2.....
                                  User Field3.....
Permanent Budget....            Track Temp..... Y
Permanent Allocated. 570.8Gi     Track VSAM..... Y
Permanent H-W-M.... 672.5Gi     Track HSM..... Y
Temporary Budget....           Add Temp..... N
Temporary Allocated. 558197064  Add VSAM..... N
Temporary H-W-M.... 809728560   Add HSM..... N
VSAM Budget.....              Reset H-W-M..... N
VSAM Allocated..... 156.1Gi     Mark Delete..... N
VSAM H-W-M..... 184.9Gi        Appl Mode.....
HSM Allocated.....            Warning Threshold. 0
HSM H-W-M.....                Volume Count..... 537

```

This view is used to review and update applications in real time. No TSO commands are used.

**Step 3** Perform the user actions described below to review and update applications in real time.

- To refresh the panel with the most current application information, press **Enter**. The panel is refreshed unless an update is pending.
- To update an application in the view, type the changes, type **U** in the detail line as indicated, and press **Enter**. Any items with validation errors are highlighted and an error message is displayed.
- To cancel an update and return to the previous panel, type **CANCEL** and press **Enter**.

**Step 4** To return to the previous panel, type **END** and press **Enter**, or press **PF3**.

**Tip:** Budgets are used by StorageGUARD views as the basis for percentage usage calculation and graphic and tabular displays. BMC Software recommends that a budget be specified for all applications running in monitor mode. This budget will be used only for StorageGUARD view calculation reporting purposes.

# Displaying Data Set Information

The data set information option is used to display the application codes associated with one or more data sets. This panel builds and executes the TSO BUDDSN command.

**Step 1** In the MAINVIEW SRM Application Maintenance panel, select **Data Set Info**.

The Data Set Information panel is displayed, as shown in the following figure.

**Figure 4-8 ACCTDET Dialog Box**

```

30MAY2001 13:20:12 ----- INFORMATION DISPLAY -----
COMMAND ==>                                     SCROLL ==> PAGE
CURR WIN ==> 1           ALT WIN ==>
W1 =ACCTDET=====SJSH=====*=====30MAY2001==13:20:12====MVSMSGC=D====1
  <== Type S to process request

Data Set Information .....

Data Set Name..... SYS1*

Volume..... *
```

**Step 2** Based on the information described in the following table, type your selection criteria.

**Step 3** Type **S** in the field indicated to process the request, and press **Enter**.

**Table 4-6 Data Set Information Entry Panel Fields**

Field	Description
Data set name or pattern	Specify a fully qualified data set name or a pattern. Use a fully qualified name if a single data set is to be selected. Use a pattern to select multiple data sets. The pattern consists of a partial data set name followed by an asterisk (*). An asterisk can be used when selecting all data sets on a volume.
Volume Serial	Specify a volume serial when selecting uncataloged data sets. Volume serial should also be specified when selecting all data sets on a volume.

Application information is displayed for the data set specified, as shown in Figure 4-9, or for all applications that meet the selection wildcard.

**Figure 4-9 APPLDSNT View**

```

30MAY2001 15:36:52 ----- INFORMATION DISPLAY -----
COMMAND ==>
CURR WIN ==> 1 ALT WIN ==>
>W1 =APPLDSNT=====SJSH=====*****30MAY2001==15:36:19====MVS RMSGC=D=2346
C Data Sets Data Set
- Applications Name
SGC3303 DSNAME=SYS1.AACBCNTL VOLSER=O26G13 SYS1.AACBCNTL
SGC3304 ---APPLICATION=OTHERS LEVEL=1 SYS1.AACBCNTL
SGC3304 ---APPLICATION=BMC-ATLANTA LEVEL=2 SYS1.AACBCNTL
SGC3304 ---APPLICATION=O26G13 LEVEL=3 SYS1.AACBCNTL
SGC3303 DSNAME=SYS1.AADFMAC1 VOLSER=O26G13 SYS1.AADFMAC1
SGC3304 ---APPLICATION=OTHERS LEVEL=1 SYS1.AADFMAC1
SGC3304 ---APPLICATION=BMC-ATLANTA LEVEL=2 SYS1.AADFMAC1
SGC3304 ---APPLICATION=O26G13 LEVEL=3 SYS1.AADFMAC1
SGC3303 DSNAME=SYS1.AADRLIB VOLSER=O26G13 SYS1.AADRLIB
SGC3304 ---APPLICATION=OTHERS LEVEL=1 SYS1.AADRLIB
SGC3304 ---APPLICATION=BMC-ATLANTA LEVEL=2 SYS1.AADRLIB
SGC3304 ---APPLICATION=O26G13 LEVEL=3 SYS1.AADRLIB
SGC3303 DSNAME=SYS1.AADRYLIB VOLSER=O26G13 SYS1.AADRYLIB
SGC3304 ---APPLICATION=OTHERS LEVEL=1 SYS1.AADRYLIB
SGC3304 ---APPLICATION=BMC-ATLANTA LEVEL=2 SYS1.AADRYLIB
SGC3304 ---APPLICATION=O26G13 LEVEL=3 SYS1.AADRYLIB
    
```

**Example**

To select all data sets that begin with WGPL, type

```
APPLDSNT WGPL.* *
```

**Example**

To select all data sets on volume WIDG01, type

```
APPLDSNT * WIDG01
```

**Step 4** To return to the previous panel, type **END** and press **Enter**, or press **PF3**.



---

---

# Chapter 5 Batch Utility Programs

This chapter describes the SGCMAINT and SGCRSYNC batch utility programs.

Initializing and Updating the Database . . . . .	5-2
Control Statements . . . . .	5-2
Comments . . . . .	5-3
Resynchronizing the Database . . . . .	5-11
SGCMAINT RESYNC Statement . . . . .	5-12
SGCRSYNC Multitasking RESYNC . . . . .	5-16
Diagnostic Reporting . . . . .	5-18
SG-Control Database Reports . . . . .	5-18
SG-Control Database Compare . . . . .	5-20

## Overview

You can use the SGCMAINT utility program to

- initialize and update the database
- resynchronize the database by individual volume
- generate diagnostic reports

You can use the SGCRSYNC is used to resynchronize multiple volumes simultaneously.

## Initializing and Updating the Database

The SGCMAINT utility program initializes and updates the SG-Control database. To ensure integrity, SGCMAINT never changes the existing database. SGCMAINT always creates a new database. After you verify that the new database is correct, change the SVOS started task (refer to the *MAINVIEW SRM Implementation Guide* for more information) to point to the new database and restart SG-Control. This provides you with the old database as a backup, if needed. Figure 5-1 contains the JCL required to run SGCMAINT.

**Figure 5-1 SGCMAINT JCL**

---

```
//INITDB EXEC PGM=SGCMAINT
//STEPLIB DD DISP=SHR,DSN=?prefix.BBLINK
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSUT1 DD UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN DD *
```

---

### Control Statements

Three control statements initialize the SG-Control database:

- **File Control**—This statement is required and is always the first statement.
- **Option Control**—This statement is required only when creating a new SG-Control database. You can include an option on an UPDATE if you want to change any of the options, for example, the size of the database.
- **Field Control**—This statement is the last initialization statement. One field control statement is needed for each application level associated with a data set.

**Note:** The order of these statements is important. Each statement is explained in greater detail on the following pages.

If you cannot fit all the parameters on a control statement, stop at the end of a keyword and repeat the INIT keyword on the next statement.

**Example**

The first control statement:

```
INIT FILE(NEW) DBNAME(EMP.SGC.DB) VOLUME(MVSRES)
```

is the same as the following:

```
INIT FILE(NEW) DBNAME(EMP.SGC.DB)
INIT VOLUME(MVSRES)
```

**Comments**

Comments and blank lines can be used to document and improve the readability of the SGCMAINT control statements. Comments should conform to the following conventions:

- Any statement with an asterisk in column 1 is considered a comment.
- Comments must be separated from the last operand on a line by at least one space followed by an asterisk.

```
*
* THIS IS A COMMENT
*
INIT FILE(NEW) DSNAME(EMP.SGC.DB)
* ANOTHER COMMENT
```

**File Control Statement**

The first control statement indicates whether a new database is being created or an updated copy is being created from an existing database. The format of the File Control statement follows and each keyword is explained below.

```
INIT FILE(NEW | UPDATE)
FROMDB(old-database)
DBNAME(new-database)
VOLUME(volser)
PURGE(days-inactive)
```

- **FILE(NEW | UPDATE)**—Required parameter. Specify NEW if the database is to be created. If the database options and application information records are to be copied from an existing file, specify UPDATE. All application information records are initialized to zero for a NEW file.
- **FROMDB(old-database)**—Required when FILE(UPDATE) is specified. It contains the name of the SG-Control database to be used as input. The database application records are copied after the new SG-Control database control record is created. Options can be overridden in the new database control record by including option control statements.
- **DBNAME(new-database)**—Required parameter that contains the data set name of the new SG-Control database. The capacity and physical size of the database is determined by the RECORDS parameter on the option control statement.
- **VOLUME(volser)**—The device track capacity of the volser specified here or by default must be at least as great as that of a 3380. Application space utilization is tracked at the volume detail when RESYNC by volume is enabled. New implementations of RESYNC by volume would require RESYNC over all old volumes to establish accurate volume totals. Any partial RESYNC by a new implementation would invalidate the application space totals until all volumes are resynchronized.
- **PURGE(days-inactive)**—Optional with the UPDATE option. Application information records that have not been used during the time period specified on this parameter for *days-inactive* are dropped from the output file. Only applications with zero space utilization are considered.

### Option Control Statement

The second control statement specifies system options that are used to control resources used by SG-Control, as well as the actions that can be taken when a budget is exceeded. The format of the option control statement follows and each keyword is explained below.

---

```

INIT RECORDS(#-of-records)
      MODE(MONITOR | WARNING | REJECT)
      REJECT(COND | YES)
      MAXKEY(50 | max-key-length)
      HSMTRACK(YES | NO)
      HBACKUPPREFIX(DFHSM backup prefix)
      HMIGRATEPREFIX(DFHSM migrate prefix)
    
```

---

- **RECORDS(# of records)**—Required parameter specifying the number of database records to create. One record is required for each application. Additional records will be needed when applications have data sets on more than the value specified for RECVOLS or 902 volumes. Each additional record will accommodate the value specified for RECVOLS or at most another 902 volumes for an application. If the total number of database records is exhausted, volume totals can be lost and new applications cannot be added. The SGCMAINT program must be rerun to change the number of database records, so calculate this number as accurately as possible. The volume totals enable a RESYNC by volume. Volume totals can be recaptured by creating a copy of the database with more records and then running RESYNC against volumes with lost totals. The default value is RECORDS(1000).
- **RECVOLS(# volume totals/record)**—Optional parameter used to implement RESYNC by volume. This keyword also specifies the number of volume space totals allowed per record for an application. A value greater than zero will collect application space totals at the volume level and enable RESYNC by volume. A zero value will disable the RESYNC by volume capability. This value is used to calculate the database record size. The maximum number of the volume totals in a database record is 902. If necessary, multiple records will be used to collect all volume totals associated with the application.
- **MODE(MONITOR | WARNING | REJECT)**—Optional parameter that sets the system mode option. If the system is running in MONITOR mode, SG-Control only records space allocation totals. If the system is running in WARNING mode, a warning message is generated if the budget amount is exceeded. In REJECT mode, the budget amount is used to place a limit on the amount of space that can be allocated. Any allocation request that would cause an application to exceed its budget is denied. The system mode option can be overridden by application by using the application mode option. The application mode option can be set using the TSO BUDGET command.
- **REJECT(COND | YES)**—Optional parameter used if MODE(REJECT) is specified. SG-Control can conditionally reject an allocation request on a specific volume when you specify REJECT(COND). A code is returned to DADSM, which requests a different volume for consideration. Conditionally rejecting a DADSM request makes sense only if the volume serial number is used to determine the budget application. Otherwise, conditional rejection makes DADSM call SG-Control with a new volume only to be rejected by the same application that rejected the first request. The default is YES, which fails the requested allocation.

- **MAXKEY(50 | max-key-len)**—Optional parameter that specifies the length of the largest budget application code. While this can be calculated from the field control statements, doing so does not take into application local installation standards or application codes that are built into the selection module, Select Exit, or Key Exit.

For example, the high-level qualifier of the data set name can be up to eight characters (your installation standard might restrict it to five). If this were your only or longest application code, you would specify a value of 5 for MAXKEY. The default is 50 characters, which is also the maximum. SG-Control builds a table in Extended CSA (for XA systems) or CSA (for 370 systems) that contains each application code and its associated record number. SG-Control uses the value specified by MAXKEY to determine the amount of CSA needed for this table. If you are an MVS/370 customer, the size of this table may affect your Virtual Storage Constraints.

- **HSMTRACK(YES | NO)**—Optional parameter that tells SG-Control to track DFHSM migrate, backup, recall, and recovery functions (when YES is specified). Selection of this option causes SG-Control to replace QUAL1, QUAL2, and the first two qualifiers of the DSNNAME field with the values extracted from the associated migration level 0 data set name. When NO is specified, SG-Control processes DFHSM requests as normal data set requests.
- **HBACKUPPREFIX(DFHSM backup prefix)**—Required parameter if HSMTRACK (YES) is specified. HBACKUPPREFIX is used to identify DFHSM backup data sets and must specify either the UID parameter specification from the DFHSM startup procedure or the override BACKUPPREFIX specification from the DFHSM control parameter.
- **HMIGRATEPREFIX(DFHSM migrate prefix)**—Required parameter if HSMTRACK (YES) is specified. HMIGRATE is used to identify DFHSM migrated data sets and must specify either the UID parameter specification from the DFHSM startup procedure or the override MIGRATEPREFIX specification from the DFHSM control parameters.

### Unsupported Option Control Statements

The following option control statements are no longer supported. System functionality has been improved, which eliminated the need for these parameters. Informational messages will tell you that these parameters are ignored. You can remove them from your control card to eliminate the messages.

- **MAXFREE**—The optimum value for MAXFREE is calculated automatically each time SG-Control is started.

- **MAXVOLS**—The maximum volume limitation has been removed.
- **MAXVOLSW**— Since there is no volume limitation, the maximum volume warning is no longer needed.
- **NOMAXVOLS**—Volume resync is always available.

### Field Control Statement

The last of the three control statements (the statements that define the default application code construction rules in the system options) is the Field Control statement. A separate field control statement must be specified to activate accounting for each of the levels. During space allocation request processing, the selection module will be called for each level. The select language *Level* statements can be used to test the level in the selection module.

**Warning!** *At least one level must be specified. If an INIT LEVEL parameter is not specified during database initialization, SG-Control will not create or update any applications.*

The format of the field control statement follows and each keyword is explained below.

---

```
INIT  RECALC
      LEVEL(1 | 2 | 3 | 4)
      FIELD1(fld, pos, len)
      FIELD2(fld, pos, len)
      FIELD3(fld, pos, len)
      FIELD4(fld, pos, len)
```

---

- **RECALC**—Optional parameter that explicitly defines the new application code as recalculated. All information used in defining the application code can be obtained from information already kept in the VTOC.
- **LEVEL(1 | 2 | 3 | 4)**—Required parameter that contains a 1, 2, 3, or 4 depending on the application being defined. This level number can be one of the fields that make up the application code to ensure there is no overlap between levels. There is no hierarchical relationship between applications.

- **FIELD1(fld,pos,len)**—Required parameter that contains the name of the first field used in creating this application code (*fld*), the position in that field (*pos*), and the number of characters to be used (*len*). Table 5-1 on page 5-11 contains a list of the fields available when building an application code. This table also contains the default lengths. If you do not specify *pos*, it is assumed to be 1. If you do not specify *len*, the default length is assumed.
- **FIELD2(fld,pos,len)**—Optional parameter that contains the second field used in creating this application code. This field is concatenated to FIELD1 to form the application code. Blanks are neither added nor removed. The description of the subfields is the same as FIELD1.
- **FIELD3(fld,pos,len)**—Optional parameter that contains the third field used in creating this application code. This field is concatenated to FIELD2 to form the application code. Blanks are neither added nor removed. The description of the subfields is the same as FIELD1.
- **FIELD4(fld,pos,len)**—Optional parameter that contains the fourth field used in creating this application code. This field is concatenated to FIELD3 to form the application code. Blanks are neither added nor removed. The description of the subfields is the same as FIELD1.

## Examples

The following examples illustrate different approaches to initializing and updating a SG-Control database:

### Example

In this example, the installation wants to establish a single-level SG-Control application based on the high-level qualifier. Only 200 records are allocated because the number of high-level qualifiers is small. Warning mode is also selected. The new database is named EMP.SGC.DB and is allocated on volume ABC001.

```
INIT  FILE(NEW) DBNAME(EMP.SGC.SPACEDB) VOLUME(ABC001)
INIT  RECORDS(200) MODE(WARNING) MAXKEY(8)
INIT  RECALC FIELD1(QUAL1) LEVEL(1)
```

**Example**

In this example, the installation wants to set up one application level to limit the total amount of space each user group can use. A user group is identified by the first three characters of the second-level qualifier of the data set name. Another application level is used to control the amount of space an individual can use. Individuals are identified by the first-level qualifier of the data set name. The number of applications anticipated is 1375. The new database is named EMP.SGC.DB and is allocated on volume ABC001.

```
INIT FILE(NEW) DBNAME(EMP.SGC.SPACEDB) VOLUME(ABC001)
INIT RECORDS(1375)
INIT MAXKEY(8) MODE(WARNING)
INIT RECALC FIELD1(QUAL2,1,3) LEVEL(1)
INIT RECALC FIELD1(QUAL1) LEVEL(2)
```

**Example**

In this example, the installation wants to establish four levels of accounting. The actual application code will be established by either the selection module or a user exit. Four levels are defined at initialization time to ensure that the select logic and user exit are used four times for each data set. The maximum application code length is 10. The installation also wants to be able to RESYNC by volume.

```
INIT FILE(NEW) DBNAME(EMP.SGC.SPACEDB) VOLUME(SUS001)
INIT RECORDS(1000) MAXKEY(10)
INIT LEVEL(1) FIELD1(QUAL1)
INIT LEVEL(2) FIELD1(QUAL2)
INIT LEVEL(3) FIELD1(QUAL3)
INIT LEVEL(4) FIELD1(QUAL4)
```

**Example**

In this example, the installation wants to change from warning to reject mode. The old database is named EMP.SGC.DB. The new database is named EMP.SGC.NEWDDB and is allocated on volume XY2001. Application information will be copied from the old to the new database.

---

```
INIT FILE(UPDATE) FROMDB(EMP.SGC.SPACEDB)
INIT DBNAME(EMP.SGC.NEWDB) VOLUME(XY2001)
INIT MODE(REJECT)
```

---

---

**Example**

In this example, the installation wants to implement the RESYNC by Volume feature and increase the size of the database to 1000 records.

The old database is named EMP.SGC.DB. The new database is named EMP.SGC.NEWDB and is allocated on volume XY2001. Application information will be copied from the old to the new database.

---

```
INIT FILE(UPDATE) FROMDB(EMP.SGC.SPACEDB)
INIT DBNAME(EMP.SGC.NEWDB) VOLUME(XY2001)
INIT RECORDS(1000)
```

---

---

**Example**

In this example, the installation wants to add HSM tracking to the current database. The prefix for HSM backup data sets is HSMBKP and the prefix for migrate data sets is HSMMIG.

The old database is named EMP.SGC.DB. The new database is named EMP.SGC.NEWDB.

---

```
INIT FILE(UPDATE) FROMDB(EMP.SGC.SPACEDB)
INIT DBNAME(EMP.SGC.NEWDB) VOLUME(XY2001)
INIT HSMTRACK(YES)
INIT HBACKUPPREFIX(HSMBKP) HMIGRATEPREFIX(HSMMIG)
```

---

## Application Code Fields

The following table describes the fields available when building an application code.

**Table 5-1 Application Code Fields**

Field Name	Length	Description
DATACLAS	08	SMS data class
DSNAME	44	Data set name
DSORG	02	Data set organization
LEVEL	01	Application code-level number
MGMTCLAS	08	SMS management class
QUAL1	08	First-level qualifier
QUAL2	08	Second-level qualifier
QUAL3	08	Third-level qualifier
QUAL4	08	Fourth-level qualifier
QUAL5	08	Fifth-level qualifier
QUAL6	08	Sixth-level qualifier
QUAL7	08	Seventh-level qualifier
QUAL8	08	Eighth-level qualifier
QUALL	08	Last qualifier
SMS	01	SMS flag
STORCLAS	08	SMS storage class
STORGRP	08	SMS storage group
TEMP	01	Temporary flag
USRC1	08	User field from the global work area
USRC2	08	User field from the global work area
VOLSER	06	Volume serial number

## Resynchronizing the Database

The SG-Control database resynchronization function can be used to populate a new SG-Control database with application information or to update application information in an existing database.

A production database may not contain up-to-date information if

- a volume was restored from a full volume backup.
- a volume was removed from the system.
- SG-Control was stopped while DASD files were created, extended, renamed, released, or deleted.

The RESYNC program processes each DASD volume individually. A volume is selected and all allocations to the volume are delayed while the VTOC is read and the appropriate applications are created and/or updated in the database.

The majority of the RESYNC elapse time is spent performing I/O operations. RESYNC elapse time for a volume can be roughly estimated by counting one I/O per data set and two I/Os each time a budget application is updated for a data set and then multiplying the number of I/Os by the average response time for the DASD devices being used. Initially, a computation adjustment of 50 percent should be added to application for RESYNC CPU utilization and operating system overhead. With time, you should be able to develop a more accurate range of computation adjustment for your site.

Individual volumes can be resynchronized with the active database. Before resynchronizing, users need to evaluate the impact of delaying allocations on each of the volumes. Volumes with little activity probably can be resynchronized at any time with little, if any, impact. Volumes with high activity ideally should be resynchronized during a period of minimal activity to limit the impact.

The multivolume RESYNC feature of SG-Control allows volume totals to be kept for each application. These totals make it possible to RESYNC individual volumes and maintain accurate application totals. If multivolume support is not selected and an attempt is made to RESYNC individual volumes, it is impossible to know what space was being tracked on these volumes prior to the RESYNC and inaccurate application totals may result.

There are two types of RESYNC: SGMAINT RESYNC and SGCRSYNC RESYNC.

## SGCMAINT RESYNC Statement

The SGMAINT RESYNC statement requests the resynchronization operation and specifies the database name and listing options. The format of the SGCMAINT RESYNC statement follows and each keyword is explained below.

---

```
RESYNC  DBNAME(database) | ACTIVEDB
        AUDIT(DSNAME, VOLSER)
        CLEAR | NOCLR
```

---

- **DBNAME(database)**—Specifies the name of the SG-Control database to be updated.

- **ACTIVEDB**—Updates the active SG-Control database (default).  
  
**Note:** If you use the ACTIVEDB option, SG-Control must be running. If neither DBNAME nor ACTIVEDB is specified, ACTIVEDB is assumed.
- **AUDIT(DSNAME,VOLSER)**—Optional parameter that specifies the level of audit reporting desired. The VOLSER value generates a message for each volume. The DSNAME value generates a message for each data set. The VOLSER value is recommended, especially if your installation wants to exclude certain volumes from SG-Control control. Either one or both values may be specified.
- **CLEAR | NOCLR**—Optional parameter that controls whether the current allocation totals are to be cleared before starting the RESYNC operation. If the RESYNC by volume feature is in effect, the volume totals are cleared automatically for each volume being resynchronized and the current allocation totals are recalculated. If the RESYNC by volume feature is not in effect, the current allocation totals are cleared by default.

The CLEAR | NOCLR parameter is provided to allow the RESYNC operation to be broken up into several steps. If the RESYNC by volume feature is in effect and a volume is being RESYNCD for the first time, new volume total entries are built and the current space allocation totals are recalculated. If RESYNC by volume is not in effect, a RESYNC of all of the DASD can be broken up into multiple steps with CLEAR specified in the first step and NOCLR specified in the subsequent steps. This will prevent the current allocation totals from being cleared in each step unnecessarily.

## VOLUME Selection Statement

VOLUME Selection statements are optional and restrict the resynchronization operation to specific volume serials. They allow you to specify the volumes to be included or excluded through the use of INCLUDE and EXCLUDE statements. The INCLUDE and EXCLUDE statements cannot be mixed in a RESYNC operation. You must use either all INCLUDE or all EXCLUDE statements. The CLEAR statement can be used to remove totals for specific volumes from the SG-Control database.

The volume serial numbers specified on these statements can use masking. Code only one volume serial number or a mask. The total number of selection statements cannot exceed 40.

The format of this control statement is

---

```
VOLUME INCLUDE(volser)
VOLUME EXCLUDE(volser)
VOLUME CLEAR(volser)
```

---

- **INCLUDE(volser)**—Specifies *a* volume serial number or *a* mask that SG-Control considers during the RESYNC operation. Volume serial numbers that are not specified or that do not match a specified mask are not used to rebuild the SG-Control database. For example, if you use `VOLUME INCLUDE( PROD/ , WORK/ )`, the system finds only those volumes that meet the `PROD/` mask and ignores the `WORK/` mask. To code both, use separate `INCLUDE` statements.
- **EXCLUDE(volser)**—Specifies a volume serial number or a mask that SG-Control does *not* consider during the RESYNC operation. The volumes that are not specified or that do not match the specified mask are used to rebuild the SG-Control database.
- **CLEAR(volser)**—Specifies a volume serial or mask that SG-Control uses in the RESYNC operation to remove totals for specific volumes from the SG-Control database. This statement can be used only if multivolume support has been selected. The `CLEAR` statement can be combined with `INCLUDE` or `EXCLUDE` statements.

## Examples

The following examples illustrate several different RESYNC scenarios:

### Example

In this example, the installation wants to create an SG-Control database and then populate it with current information by running RESYNC. RESYNC by volume is in effect and SG-Control is not active. Activity on the system is minimal. This is the fastest way to build a database with the least impact on end users.

```
RESYNC DBNAME(EMP.SGC.TEST)
```

### Example

In this example, the installation wants to RESYNC the active database. SG-Control is active and RESYNC by volume is in effect.

```
RESYNC ACTIVEDB
```

**Example**

In this example, the installation wants to RESYNC a group of volumes whose volume serial number begins with PROD with the active database. SG-Control is active and RESYNC by volume is in effect.

```
RESYNC ACTIVEDB  
VOLUME INCLUDE (PROD/)
```

**Example**

In this example, the installation wants to clear the totals for those volumes with a volume serial starting with TESTA from the SG-Control database. SG-Control is active and RESYNC by volume is in effect.

```
RESYNC ACTIVEDB  
VOLUME CLEAR (TESTA/)
```

**Example**

In this example, the installation wants to completely RESYNC the active database and list each volume.

```
RSYNC ACTIVEDB AUDIT(VOLSER)
```

**Example**

In this example, the installation wants to completely RESYNC the active database, list each volume, and list data set names on volume EMP001.

```
RSYNC ACTIVEDB AUDIT(DSNAME, VOLSER)  
VOLUME INCLUDE (EMP001)
```

## SGCRSYNC Multitasking RESYNC

Use SGCRSYNC to resynchronize multiple volumes simultaneously, which is much faster than SGMaint RESYNC.

A sample JCL for SGCRSYNC can be found in member SGCMTJR1 in *?prefix.BBSAMP*.

SGCRSYNC keywords are as follows:

- **RESYNC**—Select volumes to be resynchronized.
  - Multiple RESYNC statements can be specified.
  - INCLUDE()/EXCLUDE()/ALL VOLS—Used to select volumes; can include or exclude volumes using either a *single* volume serial or a *single* mask. For example, if you code  
RESYNC INCLUDE(BSD/ ,EMP/ ), the system finds only those volumes that start with the BSD/ mask and ignores the EMP/ mask. To code both, use separate INCLUDE statements. EXCLUDE control statements must appear *after* all INCLUDE control statements.
  - CLEAR—Clear volume totals in the database. Use this parameter to clear volume specific totals when the volume no longer exist on the system. The default is ALLVOLS.
- **MAXTASKS**—Controls the number of RESYNC tasks.
  - TASKS()—Specify the number of RESYNC tasks. The RESYNC tasks run simultaneously. Each task will RESYNC one volume at a time until all the selected volumes have been resynchronized. The default is 1 and the maximum is 16.
- **RESERVE**—Controls use of volume reserves.
  - INCLUDE()/EXCLUDE()/ALL VOLS—Used to select volumes. Can include or exclude volumes using either a single volume serial or mask.
    - RESERVE is optional; the default is ALLVOLS.
    - Multiple RESERVE statements may be specified.
- **AUDITDSN**—Displays application information for each data set.
- **AUDITVOL**—Displays status for each volume.

## Sample Control Statements

The following examples illustrate several different RESYNC control statements:

### Example

RESYNC all volumes using three RESYNC tasks.

```
RESYNC  
MAXTASKS TASKS(3)
```

### Example

RESYNC all volumes with volume serials starting with EMP. Three RESYNC tasks are used.

```
RESYNC INCLUDE (EMP/)  
MAXTASKS TASKS(3)
```

### Example

RESYNC all volumes with volume serials starting with EMP except EMP001. Ten RESYNC tasks are used.

```
RESYNC INCLUDE(EMP/)  
RESYNC EXCLUDE(EMP001)  
MAXTASKS TASKS(10)
```

## Diagnostic Reporting

The SGCMAINT batch utility program can be used to print database records compare databases.

### SG-Control Database Reports

The SGCMAINT batch utility program can be used to print database records through the use of the REPORT statement. The name of the SG-Control database can be specified in the DBNAME field of the Report Control statement. The two formats of the REPORT statement follow and each keyword is explained below.

---

```
REPORT FORMAT(DBC | DBI) DBNAME(database)
REPORT DUMP(DBC | DBI) DBNAME(database)
```

---

- **FORMAT(DBC | DBI)**—Optional parameter specifying that either the database control record (DBC) or each application information record (DBI) is to be printed with each field formatted.
- **DUMP(DBC | DBI)**—Optional parameter specifying that either the database control record (DBC) or each application information record (DBI) is to be printed in hexadecimal characters.
- **DBNAME(database)**—Optional parameter specifying the name of the SG-Control database to be printed. The active database is assumed if DBNAME is omitted.

### Examples

The following examples and sample output illustrate several different REPORT functions:

---

**Example**

---

In this example, system options contained in the database control record from the active database are formatted and printed.

```
REPORT FORMAT(DBC)
```

**Figure 5-2 Sample Output - Example 1**

---

```
SGC5400 TOTAL # OF SLOTS: 1000, USED SLOTS: 196
SGC5401 ACCOUNT RULE INFORMATION FOLLOWS FOR LEVEL 1
SGC540A ACCOUNT CODE INFORMATION WILL BE DYNAMICALLY RECALCULATED
SGC5403 THE ACCOUNT IS CONSTRUCTED USING THE FOLLOWING INFORMATION
SGC5404 FIELDNAME: QUAL1, POSITION: 1, LENGTH: -COMPLETE-
SGC5401 ACCOUNT RULE INFORMATION FOLLOWS FOR LEVEL 2
SGC540A ACCOUNT CODE INFORMATION WILL BE DYNAMICALLY RECALCULATED
SGC5403 THE ACCOUNT IS CONSTRUCTED USING THE FOLLOWING INFORMATION
SGC5404 FIELDNAME: QUAL1, POSITION: 1, LENGTH: -COMPLETE-
SGC5404 FIELDNAME: QUAL2, POSITION: 1, LENGTH: -COMPLETE-
SGC5401 ACCOUNT RULE INFORMATION FOLLOWS FOR LEVEL 3
SGC5409 NO ACCOUNT CODE CONSTRUCTION INFORMATION FOUND AT THIS LEVEL
SGC5401 ACCOUNT RULE INFORMATION FOLLOWS FOR LEVEL 4
SGC5409 NO ACCOUNT CODE CONSTRUCTION INFORMATION FOUND AT THIS LEVEL
SGC5405 SPACE INFORMATION IS KEPT AS KBYTES
SGC5406 REQUESTS WILL BE UNCONDITIONALLY REJECTED
SGC5407 SYSTEM IS IN WARNING MODE
SGC5408 MAXIMUM ACCOUNT CODE LENGTH: 16
SGC540J MAXIMUM NUMBER OF VOLUMES SUPPORTED: 0
SGC540K WARNING MESSAGE AFTER 80 % OF VOLUME ENTRIES HAVE BEEN USED
SGC540L TRACK DFHSM REQUESTS: YES
SGC540M USE PART OR ALL OF THE ML0 DSN: PART
SGC540N MIGRATE DATASET PREFIX: HSM26, BACKUP DATASET PREFIX: HSM26
```

---

### Example

In this example, application records from a named database are formatted and printed.

```
REPORT  FORMAT(DBI)  DBNAME(<DBHLQ>.SGC.SPACEDB)
```

**Figure 5-3 Sample Output - Example 2**

```
SGC540B ACCOUNT: CONTROL - RECORD ID: 1
SGC540C INFO FOLLOWS FOR SPACE TYPE PERMANENT
SGC540G MAX: 999999 - CURRENT: 0 - HIGH WATER MARK: 0
SGC540C INFO FOLLOWS FOR SPACE TYPE TEMPORARY
SGC540G MAX: 999999 - CURRENT: 0 - HIGH WATER MARK: 0
SGC540C INFO FOLLOWS FOR SPACE TYPE VSAM
SGC540G MAX: 999999 - CURRENT: 0 - HIGH WATER MARK: 0
SGC540C INFO FOLLOWS FOR SPACE TYPE HSM
SGC540G MAX: 0 - CURRENT: 0 - HIGH WATER MARK: 0
SGC540D WARNING PERCENTAGE: 80% - ACCOUNT: 'SG-Control DEFAULT' CREATED:
12/12/91 - LAST USED:

SGC540B ACCOUNT: TEST - RECORD ID: 2
SGC540C INFO FOLLOWS FOR SPACE TYPE PERMANENT
SGC540G MAX: 999999 - CURRENT: 19186 - HIGH WATER MARK: 19186
SGC540C INFO FOLLOWS FOR SPACE TYPE TEMPORARY
SGC540G MAX: 999999 - CURRENT: 0 - HIGH WATER MARK: 0
SGC540C INFO FOLLOWS FOR SPACE TYPE VSAM
SGC540G MAX: 999999 - CURRENT: 0 - HIGH WATER MARK: 0
SGC540C INFO FOLLOWS FOR SPACE TYPE HSM
SGC540G MAX: 0 - CURRENT: 0 - HIGH WATER MARK: 0
SGC540D WARNING PERCENTAGE: 80% - ACCOUNT: 'SG-Control DEFAULT' CREATED:...
```

## SG-Control Database Compare

The SGCMAINT batch utility program can be used to compare two databases. This may be useful after a RESYNC to populate a new version of the database. Each record is read sequentially from the TODB and a random read is made to the FROMDB. While this is generally sufficient to detect any differences between the two files, it does not detect a record in the FROMDB that is not in the TODB. You can run the compare both ways to be absolutely certain all differences are discovered.

All application information record (DBI) fields are compared except *date of last use* and *creation date*. These fields are updated on a RESYNC if the budget application is updated. The format of the COMPARE statement follows and each keyword is explained below.

---

```
COMPARE FROMDB(random-read-db)
COMPARE TODB(sequential-read-db)
```

---

- **FROMDB(random-read-db)**—Required parameter that specifies the input database to be read randomly.
- **TODB(sequential-read-db)**—Required parameter that specifies the input database to be read sequentially.

#### Example

The following example illustrates a COMPARE function in which two databases are compared.

```
COMPARE FROMDB(EMP.SGC.SPACEDB)
COMPARE TODB(EMP.SGC.TEST)
```



---

---

# Chapter 6 TSO Commands

This chapter describes two TSO commands used in SG-Control.

Overview . . . . .	6-1
BUDDSN Command . . . . .	6-2
Listing Application Codes for a Single Data Set . . . . .	6-2
Listing Application Codes for a Group of Data Sets . . . . .	6-3
Listing Application Codes for All Data Sets on a Volume . . . . .	6-3
BUDGET Command . . . . .	6-4
Command Coding Rules . . . . .	6-4
Listing a Single Application . . . . .	6-5
Querying Several Applications . . . . .	6-6
Field Table for the BUDGET Command . . . . .	6-8
Adding/Changing a Single Application . . . . .	6-9
Mass Changes . . . . .	6-14

## Overview

SG-Control provides two TSO commands:

- BUDDSN command lists the applications associated with a data set.
- BUDGET command lists, queries, and modifies applications.

During the implementation process, the system administrator can use the BUDGET command to define the applications for your installation. The DASD space administrators may use the BUDGET command to define the DASD budgets for users within their groups.

Since the BUDGET and BUDDSN TSO commands may be available to all users, consider using the Security Exit to control access to the various functions of these commands. Refer to “SG-Control System Exits” on page A-1 for more information about exits.

When either of these TSO commands is issued, the command processor determines whether the file name SGCDB has been allocated to your TSO session. When allocated, the commands use this locally allocated SG-Control database. Otherwise, the commands use the active database.

## BUDDSN Command

The BUDDSN command allows you to list the application codes associated with a disk data set. This command scans the entries in the VTOC and determines which application codes are associated with each data set. This TSO command can be used to list the application codes for a single data set, all cataloged data sets that match a pattern, or all data sets on a volume.

### Listing Application Codes for a Single Data Set

The BUDDSN command can list the SG-Control application codes associated with a single data set. If the data set is cataloged, only the data set name is required. If the data set is not cataloged, the VOLSER parameter must be specified. To list the applications for a single data set, use the following format:

```
BUDDSN dataset VOLSER(volume-serial)
```

- **dataset**—Positional parameter that specifies the fully qualified data set name to be listed. If the data set name is not enclosed in single quotation marks, TSO adds the profile prefix.
- **VOLSER(volume-serial)**—Optional parameter that specifies the volume serial of an uncataloged data set. If not specified, the BUDDSN command locates the volume by way of the catalog entry. This parameter is required when listing uncataloged data sets.

To list the applications associated with the data set EMP.ABC.LOAD, type:

```
BUDDSN 'EMP.ABC.LOAD'
```

The output should look similar to the following:

---

```
SGC3303 DSN=EMP.ABC.LOAD VOLSER=MVS003
SGC3304 ---ACCOUNT=SYSTEMS LEVEL=1
SGC3304 ---ACCOUNT=EMP LEVEL=2
```

---

## Listing Application Codes for a Group of Data Sets

The BUDDSN command also can list the applications associated with a range of cataloged data sets. The data set range is determined by a pattern given to catalog management.

To list all the data sets using a pattern, use the following format:

```
BUDDSN pattern
```

**pattern**—Positional parameter that specifies a partial data set name followed by an asterisk. An asterisk cannot be embedded in the pattern; the asterisk must be the last character in the pattern.

**Note:** If the BUDDSN command is issued without parameters, all data set names for your TSO profile prefix are listed.

To list the application codes associated with the data sets starting with EMP.ABC, type:

```
BUDDSN 'EMP.ABC.*'
```

The output should look similar to the following:

---

```
SGC3303 DSNNAME=EMP.ABC.MACLIB VOLSER=MVS002
SGC3304 ---ACCOUNT=SYSTEMS LEVEL=1
SGC3304 ---ACCOUNT=EMP LEVEL=2
SGC3303 DSNNAME=EMP.ABC.INSTALL VOLSER=MVS002
SGC3304 ---ACCOUNT=SYSTEMS LEVEL=1
SGC3304 ---ACCOUNT=EMP LEVEL=2
SGC3303 DSNNAME=EMP.ABC.LOAD VOLSER=MVS002
SGC3304 ---ACCOUNT=SYSTEMS LEVEL=1
SGC3304 ---ACCOUNT=EMP LEVEL=2
```

---

## Listing Application Codes for All Data Sets on a Volume

The third use of the BUDDSN command allows you to examine the applications associated with each of the data sets on a specific volume or set of volumes. This command scans the VTOC and lists each data set and the associated applications. If a volume has an indexed VTOC, the data sets are listed in alphabetic order. To list the applications associated with each of the data sets on a volume or set of volumes, use the following format:

```
BUDDSN * VOLSER(volume-serial)
```

- \*—When the BUDDSN command encounters an asterisk as the first positional parameter, all data sets found on the specified volume are listed.

- **VOLSER(volume-serial)**—Specifies a volume serial or a partial volume serial followed by an asterisk. This is required when the first positional parameter is an asterisk.

#### Example

For example, to list all of the data sets and their associated applications on volume MVS002, type:

```
BUDDSN * VOLSER(MVS002)
```

The output should look similar to the following:

---

```
SGC3303 DSNNAME=DEV.BAPROD.COBOL VOLSER=MVS002
SGC3304 ---ACCOUNT=APPLIC LEVEL=1
SGC3304 ---ACCOUNT=DEV LEVEL=2
SGC3303 DSNNAME=DEV.BATST.COBOL VOLSER=MVS002
SGC3304 ---ACCOUNT=APPLIC LEVEL=1
SGC3304 ---ACCOUNT=DEV LEVEL=2
SGC3303 DSNNAME=EMP.SGC.MACLIB VOLSER=MVS002
SGC3304 ---ACCOUNT=SYSTEMS LEVEL=1
SGC3304 ---ACCOUNT=EMP LEVEL=2
SGC3303 DSNNAME=SYS3.ABC.INSTALL VOLSER=MVS002
SGC3304 ---ACCOUNT=SYSTEMS LEVEL=1
SGC3304 ---ACCOUNT=SYS3 LEVEL=2
SGC3303 DSNNAME=SYS3.ABC.LOAD VOLSER=MVS002
SGC3304 ---ACCOUNT=SYSTEMS LEVEL=1
SGC3304 ---ACCOUNT=SYS3 LEVEL=2
```

---

## BUDGET Command

The TSO BUDGET command can list, query, add, and change application information. The list feature allows you to examine a single application or a range of applications. Your installation may elect to use the security exit provided to restrict the modifying/reviewing of application information. You may want to include this command in the logon CLIST to inform your users of their current DASD budget status.

### Command Coding Rules

To use the BUDGET command, ensure that the following coding rules are observed:

Enclose parameters that contain embedded blanks or special characters in single quotation marks, for example, (ACCT,EQ,'ABC\*').

When specifying commands, masking may be used when comparing against non-numeric fields. Refer to the *MAINVIEW SRM for OS/390 Implementation Guide* for information on masking.

**Note:** When using masking, the operator must be EQ.

## Listing a Single Application

This use of the BUDGET command displays the current information about a specific application. To list a single application, use the following format:

```
BUDGET acct-code LIST(ALL | ACCT)
```

- **acct-code**—Specifies the application to be listed. Enclose application codes that contain embedded blanks in single quotation marks.
- **LIST(ALL | ACCT)**—Displays application information.

**LIST(ALL)** displays the date created, last used date, warning level, index level, current space used, maximum space used, and high-water mark for permanent, temporary, HSM, and VSAM space. The name and user fields are also listed if they contain information. The default value is ALL.

**LIST(ACCT)** displays only the application codes for the application selected.

### Example

To list the budget application ACCTING, type:

#### BUDGET ACCTING

The output should look similar to the following:

```
SGC3202 ACCTING
SGC3203 DATE-CREATED=2000/02/27 LAST-USED=2000/08/12 WARN= 0% INDEX= 235 WARNING
SGC3204 NAME = 'ACCOUNTING GROUP'
SGC3205 TYPE=PERM CUR= 7640.00K MAX= 1000.00K HWM= 9230.00K
SGC3205 TYPE=TEMP CUR= 340.00K MAX= 900.00K HWM= 4350.00K
SGC3205 TYPE=VSAM CUR= 5460.00K MAX= 800.00K HWM= 5460.00K
SGC3205 TYPE=HSM CUR= 10920.00K HWM= 11420.00K
```

## Querying Several Applications

The BUDGET command also can query the SG-Control database. In this mode, the entire database is read. Applications can be selected by comparing a field in an application to a value or by comparing a field to a field.

### Comparing a Field to a Value

To compare a field to a value, use the following format:

```
BUDGET (field1,operator,value) LIST(ALL | ACCT | NONE) TOTAL
```

- **field1**—Specifies the name of the field whose contents are to be compared. Select a name for field1 from Table 6-1 on page 6-9.
- **operator**—Can be any of the following operators: EQ (equal), NE (not equal), GT (greater than), LT (less than), GE (greater than or equal to), or LE (less than or equal to).
- **value**—Specifies the value to which the field1 subparameter is to be compared. Can be any numeric or alphanumeric value. Enclose the value in single quotation marks if it contains blanks or special characters. Masking may be used.
- **LIST(NONE)**—When LIST(NONE) is specified, nothing is displayed. This can be used to limit output during mass database changes.

- **TOTAL**—Causes total space information to be displayed for all the selected applications; optional.

#### Example

To list the application codes of all the applications that currently have more than 1000 kilobytes of permanent storage allocated, type:

**BUDGET (CURPERM,GT,1000) LIST(ACCT)**

The output should look similar to the following:

```
SGC3202 ACCTING
SGC3202 PAYROLL
SGC3202 ENGINEER
SGC3202 PRODUCT
SGC3202 OPERATI
```

#### Example

To list the total space information for all the applications that have a value of US starting in the fourth position of user field 1, type:

**BUDGET (USERF1,EQ,'%%%'US\*') LIST(NONE) TOTAL**

The output should look similar to the following:

```
SGC3202 **TOTAL**
SGC3203 DATE-CREATED=2000/02/27 LAST-USED=2000/08/12 WARN= 0% INDEX= 235 WARNING
SGC3205 TYPE=PERM CUR= 1501.87K MAX= 1286665.38K HWM= 234099.00K
SGC3205 TYPE=TEMP CUR= 0.00K MAX= 1222221.00K HWM= 6951.00K
SGC3205 TYPE=VSAM CUR= 2202.71K MAX= 1251109.86K HWM= 220271.00K
SGC3205 TYPE=HSM CUR= 3703.68 HWM= 174562
```

### Comparing a Field to a Field

To compare the contents of a field to that of a second field in an application record, use the following format:

**BUDGET (*field1*,*operator*,*field2*) LIST(ALL | ACCT | NONE) TOTAL**

- **field2**—Specifies the field to which the field1 parameter is to be compared. Select the value for the field2 parameter from Table 6-1 on page 6-9.

**Example**

To list the total space information for all the applications that have a value of US starting in the fourth position of user field 1, type:

**BUDGET (CURVSAM,GT,CURPERM) LIST(ACCT)**

The output should look similar to the following:

---

```
SGC3202 ACCTING
SGC3202 PRODUCT
SGC3202 OPERATI
```

---

## Field Table for the BUDGET Command

The following table provides the field names and descriptions of the fields that can be used in field-to-value and field-to-field comparisons.

Table 6-1 Field Table for BUDGET Command

Field name	Description
CURPERM	Current permanent space allocation
CURTEMP	Current temporary space allocation
CURVSAM	Current VSAM space allocation
CURHSM	Current DFHSM space allocation
HWMPERM	High-water mark permanent space
HWMTEMP	High-water mark temporary space
HWMVSAM	High-water mark VSAM space
HWMHSM	High-water mark DFHSM space
ACCT	Application code
MAXP	Permanent space allocation budget
MAXPERM	Permanent space allocation budget
MAXT	Temporary space allocation budget
MAXTEMP	Temporary space allocation budget
MAXV	VSAM space allocation budget
MAXVSAM	VSAM space allocation budget
NAME	User name
USERF1	User field 1
USERF2	User field 2
USERF3	User field 3
USRF1	User field 1 (alias for USERF1)
USRF2	User field 2 (alias for USERF2)
USRF3	User field 3 (alias for USERF3)
WARN	Warning percentage

## Adding/Changing a Single Application

This section describes how the BUDGET command adds or changes a single application. To add or change a single application, use the following format:

---

```

BUDGET acct-code ADD|CHANGE
      FROM(from-acct-cd)
      NAME(user-info)
      USERF1(user-info)
      USERF2(user-info)
      USERF3(user-info)
      MAXPERM(numeric)
      MAXTEMP(numeric) | NOTEMP | PTEMP | TTEMP
      MAXVSAM(numeric) | NOVSAM | PVSAM | TVSAM
      NOHSM | PHSM | THSM
      WARN(numeric < 100)
      RESET
      MONITOR | WARNING | REJECT | DEFAULT
      DELETE | NODELETE
      KILOBYTE | MEGABYTE | GIGABYTE | TERABYTE | PETABYTE | EXABYTE
      K1000 | K1024

```

---

- **acct-code**—Required positional parameter that specifies the application you want to change. Enclose application codes that contain blanks or special characters in single quotation marks.
- **ADD | CHANGE**—Required positional parameter. Specify ADD to create a new entry; CHANGE to alter an existing entry.
- **FROM(from-acct-cd)**—Optional parameter that specifies the default or model application from which the following are copied: MAXPERM, MAXTEMP, MAXVSAM, WARN, NAME, and User fields. Parameters specified on the BUDGET command override values copied from the application.
- **NAME(user-info)**—Optional parameter in which you can specify up to 20 characters of information. Enclose the information in single quotation marks if it contains embedded blanks or special characters.
- **USERF1(user-info)**—Optional parameter in which you can specify up to eight characters of information. Enclose the information in single quotation marks if it contains embedded blanks or special characters.
- **USERF2(user-info)**—Optional parameter in which you can specify up to 10 characters of information. Enclose the information in single quotation marks if it contains embedded blanks or special characters.
- **USERF3(user info)**—Optional parameter in which you can specify up to 10 characters of information. Enclose the information in single quotation marks if it contains embedded blanks or special characters.
- **MAXPERM(numeric)**—Optional parameter that specifies the maximum amount of permanent space kilobytes (depending on the system option) for this application. If you specify the PTEMP and/or PVSAM parameters, the value specified for the MAXPERM parameter should also include the maximum temporary and VSAM space allocations.
- **MAXTEMP(numeric)**—Optional parameter that specifies the maximum amount of temporary space kilobytes for this application. TTEMP is not required if MAXTEMP is specified.
- **NOTEMP**—Optional parameter that directs SG-Control to ignore all allocations for temporary space.
- **PTEMP**—Optional parameter that directs SG-Control to consider temporary space the same as permanent space and include it in that total. If the PTEMP option is changed after temporary space has been allocated, it is *not* added into the total for permanent space, nor is it cleared.

- **TTEMP**—Optional parameter that directs SG-Control to track temporary space. TTEMP is the default when adding an application.  
**Note:** You can specify MAXTEMP, NOTEMP, PTEMP, or TTEMP. TTEMP is the default.
- **MAXVSAM(numeric)**—Optional parameter that specifies the maximum amount of VSAM space in tracks or kilobytes for this application. TVSAM is not required if MAXVSAM is specified.
- **NOVSAM**—Optional parameter that directs SG-Control to ignore all allocations for VSAM space.
- **PVSAM**—Optional parameter that directs SG-Control to consider VSAM space the same as permanent space and include it in that total. If the PVSAM option is changed after VSAM space has been allocated, it is *not* added into the total for permanent space, nor is it cleared.
- **TVSAM**—Optional parameter that directs SG-Control to track VSAM space. TVSAM is the default when adding an application.  
**Note:** You can specify MAXVSAM, NOVSAM, PVSAM, or TVSAM. TVSAM is the default.
- **NOHSM**—Optional parameter that directs SG-Control to ignore all DADSM requests for DFHSM migrate or backup data sets. This option overrides the HSMTRACK (YES) system option.
- **PHSM**—Optional parameter that directs SG-Control to consider DFHSM migrate and backup space the same as permanent space and include it in that total. If the PHSM option is changed after DFHSM migrate and backup space has been allocated, it is *not* added into the total for permanent space. This option has an effect only if the system option, HSMTRACK (YES) is specified.
- **THSM**—Optional parameter that directs SG-Control to track HSM space. THSM is the default when adding an application.  
**Note:** You can specify NOHSM, PHSM, or THSM.
- **WARN(n < 100)**—Optional parameter that specifies the percentage of the budget maximum for this type of space to be used before a warning message is generated. It must be an integer between 0 and 100.
- **RESET**—Optional parameter that causes all the high-water marks to be set to the current allocation amounts.

- **MONITOR | WARNING | REJECT | DEFAULT**—Optional parameter that sets the application mode option. The application mode option overrides the mode specified as a system option.

Applications running in MONITOR mode only record space allocation totals.

If an application is running in WARNING mode, a warning message is generated if the budget is exceeded.

If an application is running in REJECT mode, the budget amount is used to place a limit on the amount of space that can be allocated. Any allocation request that would cause an application to exceed its budget is denied.

The DEFAULT mode specification can be used to clear the application mode option. The default (or system option) mode is used.

- **DELETE | NODELETE**—Optional parameter that sets the deletion requested flag. Applications flagged for deletion can be dropped from the database by using the INIT FILE (UPDATE) function of SGCMAINT. NODELETE resets the deletion request flag.

**Note:** Budgets are used by StorageGUARD as the basis for percentage usage calculation and graphic and tabular displays. BMC Software recommends that a budget be specified for all applications running in monitor mode. This budget will be used only for StorageGUARD calculation reporting purposes.

- **KILOBYTE | MEGABYTE | GIGABYTE | TERABYTE | PETABYTE | EXABYTE** (aliases are **KB | MB | GB | TB | PB | EB** respectively)—Optional scaling parameter that determines the units of all numeric values entered. The default value is KILOBYTE.
- **K1000 | K1024**—Optional parameter that defines a kilobyte and higher powers for all input and output values, for example, a terabyte is  $1000^4$  if K1000 is specified and  $1024^4$  if K1024 is specified. The default value is derived from the definition in the KBYTECALC global parameter. For definitions of global parameters, see the *MAINVIEW SRM User Guide and Reference*.

## Examples

### Example

To add an application with an application code of ACCTA, type:

```
BUDGET ACCTA ADD
```

### Example

To add an application using application MODEL#1 as a model, type:

```
BUDGET ACCTB ADD FROM(MODEL#1)
```

### Example

To add an application overriding the value of NAME from the model, type:

```
BUDGET ACCTC ADD FROM(MODEL#1)  
NAME ('ACCOUNT C')
```

### Example

To add an application overriding the value of MAXPERM from the default or CONTROL application, type:

```
BUDGET ACCTD ADD MAXPERM(50000)
```

### Example

To change the value of MAXVSAM for ACCTA, type:

```
BUDGET ACCTA CHANGE MAXVSAM(60000)
```

### Example

To set the deletion request flag for ACCTB, type:

```
BUDGET ACCTB CHANGE DELETE
```

## Mass Changes

Changes can be made to multiple applications by using the mass change function of the BUDGET command.

**Warning!** Use this function with extreme care to avoid making unintentional changes.

Before making mass changes, you may want to test the command syntax against a copy of the active database.

As with the LIST parameter, you can select applications by comparing a field to a value or a field to another field. To compare a field to a value, use the following format:

---

```

BUDGET (field1,operator,value)
CHANGE FROM(from-acct-cd)
      NAME(user-info)
      USERF1(user-info)
      USERF2(user-info)
      USERF3(user-info)
      MAXPERM(numeric)
      MAXTEMP(numeric) | NOTEMP | PTEMP | TTEMP
      MAXVSAM(numeric) | NOVSAM | PVSAM | TVSAM
      NOHSM | PHSM | THSM
      WARN(n < 100)
      RESET
      MONITOR | WARNING | REJECT | DEFAULT
      KILOBYTE | MEGABYTE | GIGABYTE | TERABYTE | PETABYTE | EXABYTE
      K1000 | K1024
      K1000 | K1024

```

---

### Example

To change the maximum space a permanent data set can allocate for every budget application that contains a user field 2 of ACCTING, type:

```
BUDGET (USERF2,EQ,'ACCTING') CHANGE MAXPERM(2000)
```

To compare a field to another field, use the following format:

---

```
BUDGET (field1,operator,field2)
CHANGE FROM(from-acct-cd)
NAME(user-info)
USERF1(user-info)
USERF2(user-info)
USERF3(user-info)
MAXPERM(numeric)
MAXTEMP(numeric) | NOTEMP | PTEMP | TTEMP
MAXVSAM(numeric) | NOVSAM | PVSAM | TVSAM
NOHSM | PHSM | THSM
WARN(n < 100)
RESET
MONITOR | WARNING | REJECT | DEFAULT
KILOBYTE | MEGABYTE | GIGABYTE | TERABYTE | PETABYTE | EXABYTE
K1000 | K1024
K1000 | K1024
```

---

**Example**

To set the warning percentage to 70 percent in all applications where user field2 matches user field3, type:

```
BUDGET (USRF2,EQ,USRF3) CHANGE WARN(70)
```

**Note:** Several special functions of the BUDGET command allow authorized users to change current allocation amounts or high-water marks. Refer to Table 6-1 on page 6-9 for more information.



---

---

# Chapter 7 Reports

This chapter describes methods of monitoring budget applications on your system and producing reports.

Overview . . . . .	7-1
TSO BUDGET Command as a Batch TMP . . . . .	7-2
DFHSM Migration and Backup Report . . . . .	7-3
SAS Reporting Procedures . . . . .	7-4
COBOL Report Programs . . . . .	7-5

## Overview

The SG-Control database contains information that may be useful to various departments in your organization. SG-Control has monitoring programs that can use either the active or other versions of the database to produce reports. The monitoring facilities in SG-Control can be used to examine the following information:

- Compare current allocation amounts to budget
- Analyze current allocation amount in contrast to available space
- Capacity planning using current allocation amount and/or high-water marks
- DASD billing based on current allocation amount and high-water marks
- DFHSM backup and migrate data set space utilization

In addition, information can be extracted from the SG-Control database at regular intervals to produce a trend analysis report.

There are four methods of monitoring the budget applications on your system and producing reports:

- Execute COBOL reporting programs. Sample programs are provided in the *?prefix.BBSAMP* library.

**Tip:** When using sample programs in the *?prefix.BBSAMP* library, you may want to copy the sample member to a user library for modification.

- Execute the TSO BUDGET command as a batch TMP.
- Execute SAS procedures. Sample procedures are included in the *?prefix.BBSAMP* library.
- Execute the DFHSM Migration and Backup Report batch program.

Each of these batch reporting tools is explained in detail in the remaining pages of this chapter.

## TSO BUDGET Command as a Batch TMP

The TSO BUDGET command can display an individual application or a range of applications. Space totals can be accumulated for all the requested applications. (Refer to Chapter 6, “TSO Commands” for more information on TSO commands.)

For example, if all the applications for the accounting department have a user field set to ACCTING, the BUDGET command can list all these applications and display the space totals. Also, you can suppress the individual applications and print only the totals.

To produce a batch report with the TSO BUDGET command, first develop the command under TSO. Then you can execute the BUDGET command in batch using IKJEFT01. The following example produces an ACCTING space report that displays only the space totals for the accounting department.

**Note:** This example assumes that the first user field has been set to ACCTING for all the applications used by the accounting department. If you want to use other than the active SG-Control database, you can add a SGCDB DD statement.

**Figure 7-1 BUDGET TSO Command Batch TMP**


---

```
//JOBNAME JOB ...
//IKJACCT EXEC PGM=IKJEFT01,DYNAMNBR=10
//STEPLIB DD DISP=SHR, DSN=<APFLIB>
//SYSPRINT DD SYSOUT=*
//SYSTEM DD SYSOUT=*
//SYSIN DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
PROFILE NOPREFIX
BUDGET (USERF1,EQ,ACCTING) LIST(NONE) TOTAL
/*
```

---

**Figure 7-2 Sample Output**


---

```
SGC3202 **TOTAL**
SGC3203 DATE-CREATED=2000/02/27 LAST-USED=2000/03/05 WARN= 0% INDEX=235
SGC3205 TYPE=PERM CUR= 5533.59K MAX= 1234.00K HWM= 22813.46K
SGC3205 TYPE=TEMP CUR= 0.00K MAX= 0.00K HWM= 0.00K
SGC3205 TYPE=VSAM CUR= 0.00K MAX= 0.00K HWM= 0.00K
SGC3205 TYPE=HSM CUR= 0.00K HWM= 0.00K
```

---

## DFHSM Migration and Backup Report

The DFHSM Migration and Backup report enables installations using DFHSM to monitor the usage of DFHSM space by each SG-Control application. Like the DFHSM accounting in the SG-Control system, the use of this report assumes that the SG-Control application name is constructed from the first two qualifiers of the original data set name. The figures represented on the report are each application name.

Sample execution JCL for this report can be found in member SGCHSMJ1 of the *?prefix.BBSAMP* library.

The following sample report shows the number of kilobytes used by the migrated and backup data sets associated with each application.

The Migration Level 0 allocation total shows the amount of DASD that was originally allocated by the data sets before they were migrated from Level 0.

The Backup Level 0 allocation total shows the amount of DASD that was allocated by the data sets when they were backed up from Level 0. The total is only accumulated for data sets that no longer exist at the Level 0 and provides you with an estimate of the amount of DASD that would be needed if all the backup data sets were to be recovered at one time.

The Migration Level 1 and Level 2 data totals show the total size in KB of the migrated data set that currently resides at migration Levels 1 and 2. If compaction is in use, this value represents the compacted size.

The backup totals show the total size in KB of the backup data sets. If compaction is in use, this value represents the compacted size.

**Figure 7-3 DFHSM Migration and Backup Totals**

SG-CONTROL DFHSM MIGRATION AND BACKUP TOTALS		VERSION 6.1			PAGE : 1	
REPORT DATE : 12/31/2000		UNITS : KBYTES				
ACCOUNT CODE	MIGRATION TOTALS			BACKUP TOTALS		
	LEVEL 0 ALLOCATION	LEVEL 1 DATA	LEVEL 2 DATA	LEVEL 0 ALLOCATION	BACKUP DATA	
BATCH	195,237.25K	21,741.59K	1,342.62K	967.48K	101,296.03K	
PROD	217,634.98K	33,010.24K	21,305.07K	31,895.36K	52,090.99K	
TEST	131,467.50K	40,298.60K	19,281.89K	460.35K	51,050.55K	
ONLINE	146,803.02K	28,452.00K	8,529.89K	67,457.45K	62,050.08K	

## SAS Reporting Procedures

The third method of generating batch reports is with the SAS interface. This interface defines all the fields in the SG-Control database in terms of SAS variables and produces the sample reports. With this facility, the powerful selection, sorting, summarizing, graphs, and statistical analysis features of SAS are available to your system and DASD space administrators.

The SAS interface includes several sample reports. Since all the variables are defined with meaningful labels, you can easily define reports that include only the applications and fields required by your installation.

Several SAS procedures are located in the SGCSAS01 member of the *?prefix.BBSAMP* library. These procedures can produce the following reports:

- tabular report of all space budgeted, by type with totals
- bar graph report of each application by type
- sorted list of all space budgeted, all active applications, and all current totals

- list of the 25 largest budget applications (in descending order)
- sorted list of all budget applications that exceed their established warning levels

Another SAS procedure, located in the SGCSAS02 member of the *?prefix.BBSAMP* library, maintains a trend analysis file for use as input to a subsequent SAS procedure (SGCSAS03) that produces a graphical representation of budget application growth trends.

**Note:** Output from the SAS reports is the same as the COBOL report output. For report samples, refer to the examples that follow.

## COBOL Report Programs

SG-Control provides three COBOL report programs. The first program can produce six reports that highlight the current status of the SG-Control applications. The remaining two programs can gather application information periodically and produce a trend report showing the changes in DASD space utilization.

The sample COBOL programs have been tested using the COBOL/II compiler. Sample JCL for compile and link edit is provided in the *?prefix.BBSAMP* library with member names SGC1CMLK, SGC2CMLK, and SGC3CMLK. These correspond to the source code SGCCOB01, SGCCOB02, and SGCCOB03. The JCL we provide must be customized to conform to your installation standards, your choice of data set names, and your compile procedure. (COB2UC is used in the samples.)

### Reporting with the SGCCOB01 Member

The first COBOL program, located in the SGCCOB01 member of the *?prefix.BBSAMP* library produces the following reports:

- Report 1—List of all active applications, current allocation amounts, and budgets.
- Report 2 —List of up to 25 of the largest users of permanent and VSAM space (in descending order).

- Report 3—List of all applications over their warning threshold. The current allocation amounts, budgets, and warning percentages are listed for each entry over the warning level. A comment flag marks the entries that have exceeded their budget.
- Report 4—Summary report of the high-water marks associated with a user (USERFn) field. The user field is selected by the REPORT4 specifying REPORT4=1 | 2 | 3 in the parameter field to select the required user field. The default value is 1 (USERF1). You can suppress this report by specifying REPORT4=N in the parameter field.
- Report 5—Graph of current allocation amounts versus budgets for selected applications. The current amount is represented with an A, the high-water mark with an H, and the budget with an M. This report gives a graphical representation of how each application stands relative to its budget.
- Report 6—Alphabetic listing of applications.

**Note:** All reports listed above (with the exception of Report 4) can be selected or suppressed by specifying Y or N next to the report name in the parameter string for SGCCOB01. Samples of these reports follow.

A sample execution JCL for this COBOL reporting program can be found in member SGCCOBJ1 of the *?prefix*.BBSAMP library.

Figure 7-4 Database Detail Report

```

REPORT 1                                SG-CONTROL DATABASE DETAIL REPORT
VERSION 6.1
REPORT DATE : 01/31/2001 PAGE : 1

ACCOUNT/TYPE      CURRENT      BUDGET      HIGH WATER      NAME
USER FIELDS

CONTROL                                01/08/2001 BUDGET-DASD DEFAULT
RESYNC
PERMANENT          0          0          0 KBYTES
TEMPORARY          0          0          0 KBYTES
VSAM               0          0          0 KBYTES
HSM               0          0          0 KBYTES

ACCTNG                                01/17/2001 12/31/2000 ACCOUNTING/EDP
RESYNC
PERMANENT          36,935      100,000     66,432 KBYTES
TEMPORARY          0          10,000      0 KBYTES
VSAM               13,345      25,000     14,756 KBYTES
HSM               10,267      11,638     11,638 KBYTES

DEVLPMT                                01/21/2001 12/31/2000 PROJECT
DEVELOPMENT
RESYNC
PERMANENT          13,949      65,000     24,187 KBYTES
TEMPORARY          0          1,000      0 KBYTES
VSAM               5,005      4,500      5,005 KBYTES
HSM               3,264      6,032     6,032 KBYTES

MISC                                    01/08//2001 12/28/2000 NON CLASSED DATA
RESYNC
PERMANENT          1,390      2,500      1,689 KBYTES
TEMPORARY          0          500        0 KBYTES
VSAM               741      1,500      2,342 KBYTES
HSM               342      1,255     1,255 KBYTES

ONLINE                                01/08/2001 12/31/2000 ONLINE CICS
RESYNC
PERMANENT          198,741     999,999     198,741 KBYTES
VSAM INCLUDED IN PERMANENT
HSM INCLUDED IN PERMANENT

PROD                                    01/08/2001 12/31/2000 PRODUCTION
APPLICATIONS
RESYNC
PERMANENT          98,992     250,000     101,745 KBYTES
TEMPORARY          0          50,000      0 KBYTES
VSAM               103,342     150,000     159,641 KBYTES
HSM               99,331     100,438     100,438 KBYTES

SALES                                    02/01/2001 12/29/2000 SALES ANALYSIS
RESYNC
PERMANENT          145,565     130,000     145,565 KBYTES
TEMPORARY          0          45,000      0 KBYTES
VSAM               17,517     12,500      19,602 KBYTES
HSM               10,344     11,498     11,498 KBYTES

TEST                                    01/30/2001 12/30/2000 TEST APPLICATIONS
RESYNC
PERMANENT          37,183     50,000     52,905 KBYTES
TEMPORARY          0          9,000      0 KBYTES
VSAM               56,997     65,000     63,103 KBYTES
HSM               34,278     12,365     12,365 KBYTES

```

**Figure 7-5 Top Five Users of Permanent and VSAM Space**

```

REPORT 2                      TOP  5 USERS OF PERMANENT AND VSAM SPACE
VERSION 6.1
REPORT DATE : 01/31/2001
PAGE : 1
    
```

ACCOUNT UNITS	PERMANENT			VSAM		
	CURRENT	BUDGET	HIGH WATER	CURRENT	BUDGET	HIGH WATER
PROD KBYTES	98,992	250,000	101,745	103,342	150,000	159,641
ONLINE KBYTES	198,741	999,999	198,741	0	0	0
SALES KBYTES	145,565	130,000	145,565	17,517	12,500	19,602
TEST KBYTES	37,183	50,000	52,905	56,997	65,000	63,103
ACCTNG KBYTES	36,935	100,000	66,432	13,345	25,000	14,756
TOTAL TOP KBYTES	517,416	1,529,999	455,366	191,201	252,500	257,102
TOTAL ALL KBYTES	532,755	1,597,499	455,366	196,947	258,500	264,449

**Figure 7-6 Applications over Warning Threshold Report**

```

REPORT 3                      SG-CONTROL ACCOUNTS OVER WARNING THRESHOLD
VERSION 6.1
REPORT DATE : 01/31/2001
PAGE : 1
                                UNITS : KBYTES
    
```

ACCOUNT WARN	PERMANENT		TEMPORARY		VSAM	
	BUDGET COMMENT	CURRENT	BUDGET	CURRENT	BUDGET	CURRENT
DEVLPMT 80	65,000	13,949	1,000	0	4,500	5,005
		** VSAM **				
SALES 80 ** PERM **	130,000	145,565	45,000	0	12,500	17,517
		** VSAM **				



## Trend Analysis Reporting

The second COBOL program, located in the SGCCOB02 member of the Installation Control library, maintains a trend analysis file for use as input to the third COBOL program. This third program, stored in the SGCCOB03 member of the *?prefix.BBSAMP* library, produces a tabular representation of application growth trends over the last 12 months. Sample execution JCL can be found in members SGCCOBJ2 and SGCCOBJ3 of *?prefix.BBSAMP*. A sample trend analysis report is presented in the following figure.

**Figure 7-10 Database Trend Report**

SG-CONTROL DATABASE TREND REPORT												VERSION 6.1
REPORT DATE : 12/31/2000												
NUMBERS REPRESENT THE AMOUNT OF DASD USED IN KEYTES												
TYPE	JAN2000	FEB2000	MAR2000	APR2000	MAY2000	JUN2000	JUL2000	AUG2000	SEP2000	OCT2000	NOV2000	DEC2000
ACCOUNT: ACCTNG												
CUR-PERM	24,671	26,205	28,122	31,156	30,830	32,543	33,913	35,872	38,452	35,543	35,943	36,935
HWM-PERM	24,671	27,105	28,455	31,156	31,156	32,978	33,913	36,456	38,501	38,501	66,432	66,432
CUR-VSAM	10,205	11,437	10,386	12,365	13,187	12,675	11,873	12,904	14,002	14,009	13,646	13,345
HWM-VSAM	11,403	11,437	11,437	12,365	13,845	13,845	13,977	13,987	14,002	14,009	14,009	14,756
ACCOUNT: CONTROL												
ACCOUNT: DEVLPMT												
CUR-PERM						8,231	9,408	19,127	11,436	11,365	12,521	13,949
HWM-PERM						8,231	9,408	24,187	24,187	24,187	24,187	24,187
CUR-VSAM						3,690	3,690	3,690	5,005	5,005	5,005	5,005
HWM-VSAM						3,690	3,690	3,690	5,005	5,005	5,005	5,005
ACCOUNT: MISC												
CUR-PERM	1,081	1,114	1,122	1,156	733	1,342	1,231	1,689	1,451	1,345	1,523	1,390
HWM-PERM	1,081	1,114	1,321	1,321	1,321	1,342	1,342	1,689	1,689	1,689	1,689	1,689
CUR-VSAM	405	723	809	914	1,145	923	1,363	1,559	1,007	909	646	741
HWM-VSAM	703	824	824	914	1,234	1,234	1,477	1,559	1,559	1,559	2,342	2,342
ACCOUNT: ONLINE												
CUR-PERM	178,801	181,511	182,341	182,555	183,377	188,212	189,903	191,398	191,815	194,342	196,321	198,741
HWM-PERM	178,801	181,511	182,341	182,555	183,377	188,212	189,903	191,398	191,815	194,342	196,321	198,741
ACCOUNT: PROD												
CUR-PERM	77,199	82,769	81,628	83,389	84,343	81,331	85,302	85,932	87,324	91,342	94,233	98,992
HWM-PERM	77,199	82,769	82,769	83,389	84,567	84,567	85,532	86,229	90,329	93,324	101,745	101,745
CUR-VSAM	92,132	92,429	96,393	95,331	95,215	94,292	96,789	145,839	159,641	132,893	121,825	103,342
HWM-VSAM	92,132	93,244	96,393	96,393	96,393	96,393	96,789	145,839	159,641	159,641	159,641	159,641
ACCOUNT: SALES												
CUR-PERM											145,565	145,565
HWM-PERM											145,565	145,565
CUR-VSAM											16,543	17,517
HWM-VSAM											17,364	19,602
ACCOUNT: TEST												
CUR-PERM											32,541	37,183
HWM-PERM											52,905	52,905
CUR-VSAM											51,847	56,997
HWM-VSAM											55,735	63,101

---

---

# Appendix A SG-Control System Exits

This appendix explains how to install exits that allow you to extend the functions of the product.

Overview . . . . .	A-1
Installing Exits . . . . .	A-2
Exit Flow . . . . .	A-3
Security Exit . . . . .	A-4
Application Code Build Exit . . . . .	A-6
Add Exit . . . . .	A-8
Check Exit . . . . .	A-9
Default Exit . . . . .	A-11
Select Exit . . . . .	A-12
Table of Sample Exits . . . . .	A-14
Application Records . . . . .	A-15
Global Work Area . . . . .	A-17
Special Considerations . . . . .	A-17

## Overview

SG-Control provides exits that allow you to extend the functions of the product. With the possible exception of the Security Exit, many installations can use SG-Control without the use of exits.

Exit	Description
Application Code Build	Called every time an application code is built. This exit allows your installation to assign application codes programmatically. In most cases, FLST and RLST parameters can provide the necessary capabilities and is recommended over the use of the Application Code Build Exit.
Add	Allows your installation to control the updating of the current allocation amounts for an application.

Exit	Description
Check	Allows your installation to control the checking of a space allocation request and budget for an application.
Default	Invoked whenever an application is created dynamically. This exit allows your installation to override the defaults assigned to the application.
Security	Called when the BUDGET or BUDDSN TSO commands are invoked. This exit allows your installation to restrict the use of functions within the TSO command.
Select	Called once during the preprocessing exit and once during the postprocessing exit. This exit allows your installation to restrict SG-Control to certain data sets or volumes. FLST and RLST parameters provide the same capability and is recommended over the use of the Select Exit.

## Installing Exits

Each exit must be linked to a separate load module in an authorized library that can be accessed by SVOS started task and by any function that requires access to any of the SG-Control functions.

The following global parameters establish the link to the load module:

- SGC\_ADDEXIT—Load module name of the Add Exit.
- SGC\_CHKEXIT—Load module name of the Check Exit. If blank, all DASD allocations are monitored.
- SGC\_DEFEXIT—Load module name of the Default Exit. If blank, all applications created dynamically use the defaults from the CONTROL budget application.
- SGC\_KEYEXIT—Load module name of the Application Code Build Exit. If blank, all application codes are built from the fields specified when the SG-Control database was initialized. Application codes also can be constructed using the NEWAPPL statement in the selection language.
- SGC\_SECEXIT—Load module name of the Security Exit. If blank, SG-Control security is not in effect.

- **SGC\_SELEXIT**—Load module name of the Select Exit. If blank, space allocation requests are monitored. FLST and RLST parameters provide a similar function and its use is recommended.

**Note:** Review “Special Considerations” on page A-17 before designating exits.

## Exit Flow

Exits can be called several times for each space allocation request. The best way to describe the exit flow is to consider the worst case. For this example, assume that the user is about to allocate a new data set. Allocation of this data set will cause new applications to be added to the SG-Control database.

### During CHECK processing

- Calls the Select Exit. The Select Exit is used to determine if a space allocation request should be tracked.
- Calls the Application Code Build Exit twice for each application code. The default application code construction rules are passed to the exit the first time. You can change the rules or bypass tracking this application code. The application code is passed to the exit the second time it is called. You can change the application code or bypass processing for this application.
- Calls the Default Exit twice each time a new application is created. The first call passes the application code of the default application. You can change the application code of the default application. The second call is made after the new application is built. You can change any field in the application record. On either call, you can choose not to track the space allocation request for this application code.
- Calls the Check Exit for each application with the amount of space requested and the application record. You can change the amount of space to be checked, skip the check, ignore the result, or reject the allocation.

### During ADD processing

- Calls the Select Exit.
- Calls the Application Code Build Exit. This call must generate the same application codes the preprocessing call generated.

- Calls the Add Exit for each application with the amount of space requested and the application record. You can change the amount of space to be updated or skip the updating of the application.

## Security Exit

The Security Exit is called each time the BUDGET or BUDDSN TSO command is executed. Misuse of the TSO BUDGET command could compromise the integrity of the SG-Control database.

For example, the BUDGET command can be used to list and/or change budget and allocation amounts. Since it would be impossible to control DASD space if all users could change application budgets at will, you should either use the exit or restrict access to the BUDGET command.

A mode flag is passed to the Security Exit indicating whether the user wants to read, change, or add a record. The exit is called once with a special mode flag if the user wants to process the entire database. Both the original and updated copies of the records are presented to the exit. This allows the exit to identify any fields that have changed. Possible uses of this exit are to

- Control who can add to or make changes to the SG-Control database
- Control who can reset the high water marks
- Distribute database control
- Force the use of a backup copy of the database for reporting and query

### Control Access

The primary function of the Security Exit is to control access to the applications. The simplest way is to hard code lists of user IDs that are allowed to read, change, or add selected applications. This approach can be difficult to maintain and does not allow for the distribution of authority.

If you have RACF, you can set up a new RACF class with the application as the entry and have RACF maintain the access list. All RACF facilities are then available to maintain the list. The security exit simply issues a RACHECK. RACF access of READ, UPDATE, and ALTER can correspond to READ, CHANGE, and ADD in SG-Control. RACF entry must be added and an access list associated with the RACF entry any time a new application is added.

Another technique is to encode or store in a user field the attributes of individuals allowed to perform certain functions against the application. For example, a user could be tied to a group, giving the user the ability to list all applications within that group.

---

The Security Exit also can be used to interface with other security systems such as CA-ACF2 and CA-TOP SECRET.

### **Control Data**

An installation may want to restrict the ability to change fields in an application. For example, it may be advisable to restrict the ability to change any fields that may be used for capacity planning and costing.

### **Performance**

An installation may want to restrict the use of commands that force the database to be read sequentially. In many cases, a backup copy of the database is acceptable, especially if the user is processing a large group of records. The user can allocate a database file with ISPF panels or with the TSO ALLOC command. In this case, the BUDGET command uses this locally allocated database instead of the active database. This feature also is useful if SG-Control is not active. When a user allocates an SG-Control database file, the high-order bit of BCVTFLG1 is set on. If a sequential read is required, the installation may want to force all users to use a backup database. The system or DASD administrators should be allowed to process mass changes against the active database.

### **Register Conventions**

On entry to the Security Exit, the registers contain

- Register 0
  - 0 Read
  - 4 Change
  - 8 Add
  - 20 Sequential Read
- Register 1—Address of the updated/new application record. Zero for sequential read.
- Register 2—Address of a 300-byte work area common to all exit routines. The high-order byte of the register contains x'01'.
- Register 3—Address of the original application record. Zero for ADD or sequential read.
- Register 10—Address of static area.
- Register 11—Address of Global Work Area mapped by the DSECT BUDWORK.

- Register 13—Address of a 72-byte save area. Standard linkage conventions must be used.
- Register 14—Return Address.
- Register 15—Entry-point address of the exit.

The Security Exit is returned upon completion with Register 15 containing one of the following codes:

- 0 Allow the request to be completed.
- 8 Fail the request for this application.

**Note:** Table A-1 on page A-14 contains a list of the sample Security Exits on the distribution tape

## Application Code Build Exit

The Application Code Build Exit is called twice for each level associated with a data set. The default application code construction rules are passed to the exit on the first call. The application code is passed to the exit on the second call. The Application Code Build Exit can modify the default application code construction rules or replace the application code. FLST and RLST parameters provide SG-Control with a similar function and is the recommended technique.

FLST and RLST parameters and the Application Code Build Exit can be used to bypass processing on an application code basis. FLST and RLST parameters or the Select Exit can be used to skip processing on a data set basis. Application codes can be constructed at three points in SG-Control processing:

- Default application codes are constructed using the default application code construction rules
- FLST and RLST parameters can then be used to replace the default application code
- The Application Code Build Exit can be called to make the final changes

Possible uses of this exit are to

- Extend the SG-Control application code building facilities

- Include installation-unique information in application codes, such as special group codes that are derived using an algorithm
- Provide different application code logic for different groups of data sets
- Limit building an application to certain groups of data sets

### **Register Conventions**

On the first entry to the Application Code Build Exit, the registers contain

- Register 0—Zero.
- Register 1—Address of the default application code construction rules mapped by the DSECT KBF.
- Register 2—Address of a 450-byte work area common to all exit routines. The high-order byte of the register contains x'03'.
- Register 10—Address of static area.
- Register 11—Address of the Global Work Area mapped by the DSECT BUDWORK.
- Register 13—Address of a 72-byte save area. Standard linkage must be used.
- Register 14—Return Address.
- Register 15—Entry-point address of the exit.

On the second entry to the Application Code Build Exit, the registers contain the same information with the following exceptions:

- Register 0—Contains 4.
- Register 1—Address of the application code.

On completion, the Application Code Build Exit returns with register 15 containing one of the following codes:

- 0 Use application code pointed to by register 1.
- 4 Skip this application code.

- 8 Skip the space allocation request.

**Note:** If Register 0 contains zero (first call), Register 1 points to application-code flag fields. If Register 0 contains 4 (second call), Register 1 points to an application code.

Table A-1 on page A-14 contains a list of the sample Application Code Build Exits on the distribution tape

## Add Exit

The Add Exit is called during ADD processing. The exit is called once for each application code associated with the space allocation request. The amount of requested space is passed to the Add Exit. The amount is negative in the case of a SCRATCH or RENAME. The application also is passed to the exit. The installation can change the amount of space being requested or skip the update. Possible uses of this exit are to

- Skip updating if the data set meets some installation standard
- Add a surcharge if the allocation is on certain volumes

Any changes made in the Check Exit should be replicated in the Add Exit. You may want to use the same code for both exits. The space allocation request cannot be rejected in the Add Exit because the space has been allocated or deallocated already.

### Register Conventions

On the first entry to the Add Exit, the registers contain

- Register 0—Address of the application record.
- Register 1—Amount of space requested (positive or negative) in Kbytes.
- Register 2—Address of a 450-byte work area common to all exit routines. The high-order byte of the register contains x'06'.
- Register 10—Address of the static area. Used by the SWA mod routine.
- Register 11—Address of the Global Work Area mapped by the DSECT BUDWORK.
- Register 13—Address of a 72-byte save area. Standard linkage should be used.

- Register 14—Return address.
- Register 15—Entry-point address of the exit.

On completion, the Add Exit returns with

- Register 1—Amount of space to be used for the update.
- Register 15—Allow the update.

Skip update processing for the application code.

- 8 Skip the space allocation request

**Note:** Table A-1 on page A-14 contains a list of the sample Add Exits on the distribution tape.

## Check Exit

The Check Exit is called during CHECK processing. It can be used to check the current space allocation request, current allocation total, and budget for an application code. The exit is called once for each application code associated with the space allocation request. The amount of requested space is passed to the Check Exit. The application also is passed to the exit. The installation can change the requested space, ignore the result of the check, skip the check, or reject the space allocation request. Possible uses of this exit are to

- Skip checking if the data set meets some installation standard
- Add a surcharge if the allocation is on certain volumes
- Enforce date or time of day restrictions
- Enforce an installation standard on the maximum space any data set can request

### Suggestions

You may want to reward users who allocate small data sets by excluding them from control under SG-Control.

You might want to impose a surcharge or discount if space is requested on certain volumes. This can be implemented with the Check and the Add Exits. Since the amount of space can be negative on a SCRATCH or RENAME, the exit should check the sign of the space amount before adding or subtracting any additional amounts.

Since the Check Exit can reject a space allocation request, it could be used to enforce time of day or day of the week standards. Installation standards that restrict the amount of space a data set can request can be controlled with the Check Exit. For example, it is not possible to limit with SG-Control the total amount of space a data set may occupy eventually if all secondary requests are satisfied; SG-Control can limit only the amount of requested space for this space allocation request.

### Register Conventions

On entry to the Check Exit, the registers contain

- Register 0—Address of the database Information (application) Record mapped by the DSECT DBI.
- Register 1—Amount of space (negative or positive) requested in Kbytes.
- Register 2—Address of a 450-byte work area common to all exit routines. The high-order byte of the register contains x'05'.
- Register 10—Address of static area.
- Register 11—Address of the Global Work Area mapped by the DSECT BUDWORK.
- Register 13—Address of a 72-byte save area. Standard linkage should be used.
- Register 14—Return address.
- Register 15—Entry-point address of the exit.

On completion, the Check Exit returns with

- Register 1    Amount of space to be used for the check.
- Register 15   0    Allow the check.
  - 4    Skip check processing for this application code.
  - 8    Skip the space allocation request.
  - 12    Reject the space allocation request.

**Note:** This return causes the space allocation request to be rejected

**Note:** Table A-1 on page A-14 contains a list of the sample Check Exits on the distribution tape.

---

## Default Exit

The Default Exit is used each time an application is created. The exit is called first to allow the installation to change the application code of the default (or model) application and a second time to alter specific fields within the new application. The CONTROL application is used as a model when there is no Default Exit. The maximum permanent space, temporary space, VSAM space, NAME field, and user fields are extracted from the model. The Default Exit allows installations to override the application code of the model and to tailor fields in the application record. Possible uses of the Default Exit are to

- Determine the application code of the model based on user criteria
- Fill a user field with the TSO user ID of the creator
- Enforce installation standards governing the maximum budget any application can use

### Budget Models

The Default Exit and the TSO BUDGET command can use models. The use of model applications allows installations to set up standards that can be used when creating budget applications. SG-Control does not distinguish between real applications and model applications. It is up to the user to ensure that model applications are not used to control allocations. It is suggested that the NAME or one of the user fields be set to a special code to ensure these applications are excluded from application total reports.

One application of the Default Exit is to set up standard defaults for each user group. The Default Exit could then select the default application based on job accounting information, RACFGRP, or high-level qualifier.

Assuming an installation explicitly sets up all budget applications with the TSO BUDGET command, the budgets in the model application could be set to zero to reject any allocations against dynamically created applications. The Default Exit could be used to override this standard for selected situations.

The Default Exit also can be used to set the NAME and user fields in dynamically created applications.

### Register Conventions

On first entry to the Default Exit, the registers contain

- Register 0—Address of the application code of the model application.
- Register 1—Address of the new application record.

- Register 2—Address of a 450-byte work area common to all exit routines. The high-order byte of the register contains x'02'.
- Register 10—Address of static area.
- Register 11—Address of the Global Work Area mapped by the DSECT BUDWORK.
- Register 13—Address of a 72-byte save area. Standard linkage should be used.
- Register 14—Return Address.
- Register 15—Entry-point address of the exit.

On the second entry to the Default Exit, the registers contain the same information with the exception of Register 0, which contains zero.

On completion, the Default Exit returns with Register 15 containing one of the following codes:

- 0 Allow the application to be created.
- 8 Do not create an application. This application is ignored.

**Note:** Table A-1 on page A-14 contains a list of the sample Default Exits on the distribution tape.

## Select Exit

The Select Exit is called during CHECK and ADD processing. It is called once for each application code associated with a space allocation request. The exit can be used to decide if a request should be tracked by SG-Control. FLST and RLST parameters provide the same capability and is recommended over the use of the Select Exit. The Select Exit is used also by the BUDGET and BUDDSN TSO commands.

Possible uses of the Select Exit are to

- Restrict SG-Control to certain data sets or volumes
- Exclude certain user groups or programs
- Phased implementation of SG-Control

---

An installation may want to restrict SG-Control to a set of data sets or volume serials or to exclude certain users from SG-Control control. The Select Exit or FLST and RLST parameters allow this type of control. All fields supported in the selection language are available in the Global Work Area. The address of the Global Work Area is passed to the exit.

### Register Conventions

On entry to the Select Exit, the registers contain

- Register 0—Zero.
- Register 1—Zero.
- Register 2—Address of a 450-byte work area common to all exit routines. The high-order byte of the register contains x'04'.
- Register 10—Address of static area.
- Register 11—Address of the Global Work Area mapped by the DSECT BUDWORK.
- Register 13—Address of a 72-byte save area. Standard linkage should be used.
- Register 14—Return Address.
- Register 15—Entry-point address of the exit.

On completion, the Select Exit returns with

- Register 15
  - 0 Allow SG-Control to control this function.
  - 8 Skip all processing.

**Note:** Table A-1 on page A-14 contains a list of the sample Select Exits on the distribution tape.

## Table of Sample Exits

The following table contains a list of sample exits delivered with SG-Control. The example exits do not represent an exhaustive list of all possible combinations of exit options, but rather serve as models to be tailored to your environment.

**Note:** Users are encouraged to send SG-Control exits to BMC Software in machine readable format (3420 tapes, please). Exits that have universal application will be considered for inclusion in future releases.

The following sample exits are located in the Installation Control library *?prefix.BBSAMP*.

**Table A-1 Sample Exits (Part 1 of 2)**

Exit name	Type	Description
BUDSEL01	SELECT	Restricts SG-Control to a particular job name.
BUDSEL02	SELECT	Restricts SG-Control to a particular range of volumes. The exit contains a table of volume serials (or partial volume serials) that are <i>not</i> to be tracked.
BUDSEC01	SECURITY	Issues a RACHECK to a RACF Class of BUDGET using the application code as the resource. This allows an installation to use RACF to control the updating of budget applications.
BUDSEC02	SECURITY	Allows only the TSO user ID that matches the user ID stored in user field 1 to update the application. The exit also contains a list of user IDs that are not restricted. When the budget application is set up, the administrator sets user field 1 to the user ID of the TSO user who is allowed to update it. Since delegated users can change any field, they can update user field 1 when necessary.
BUDSEC03	SECURITY	Prohibits users from changing the space allocation totals.
BUDSEC04	SECURITY	Issues a RACROUTE to a RACF class of BUDGET using the application code as the resource. This is an identical function to BUDSEC01, but it uses the RACROUTE macro instead of RACHECK.
BUDDEF01	DEFAULT	Sets the application code of the default application to CONTROL1 for LEVEL 1, CONTROL2 for LEVEL 2, and so on. This exit can be used if there are different defaults based on the LEVEL or application being created.

**Table A-1 Sample Exits (Part 2 of 2)**

Exit name	Type	Description
BUDKEY01	APPLICATION CODE BUILD	Builds an application code that consists of the address space ID. This can be used to limit the total amount of temporary space a job can have during its execution. If there is more than one system, the SMF ID must be added to the application code.
BUDCHK01	CHECK/ADD	Doubles the amount of space that is charged if the data set is allocated on certain volumes. This might be used to add a surcharge if the space is allocated on a volume with a cached controller or on a high speed channel. The volume list is coded in the exit in the same manner as BUDSEL02.

## Application Records

The following table documents the format of the application record. An assembler DSECT for the application record is located in member DBI41 in the *?prefix.BBSAMP*.

**Note:** DPFP = floating point long format.

**Table A-2 Application Record Format (Part 1 of 3)**

Offset	Name	Type	LEN	Description
00 (00)	DBI	DSECT		
(00)	DBIKEY	CHAR	50	Application code
50 (32)	DBIFLAG1 DBIACT DBIDEL DBIMDEL DBIFMT41	HEX EQU EQU EQU EQU	1 X'80' X'40' X'20' X'08'	Status flag 1 Active Inactive Manually flagged for deletion DBI in Release 4.1.0 format
51 (33)	DBIFLAG DBIWARNM DBIMONM DBIREJM DBIDEF	HEX EQU EQU EQU EQU	1 X'80'X' 40' X'20'X' 00'	Status flag 2 Warning mode Monitor mode Reject mode Default mode
52 (34)	DBIRECID	HEX	4	Relative record number
	* * PERMANENT TOTALS *			
	DBIPERMS			
56 (38)	SPCMAX	DPFP	8	Budget amount — floating point
64 (40)	SPCCUR	DPFP	8	Current allocation — floating point

**Table A-2 Application Record Format (Part 2 of 3)**

Offset	Name	Type	LEN	Description
72 (48)	SPCHWM	DPFP	8	High-water mark —floating point
80 (50)	SPCFLAGS SPCPERM SPCNONE	HEX EQU EQU	1 X'80' X'40'	Space flags Add to perm Do not track
81 (51)		HEX	3	Slack
	* * TEMPORARY TOTALS *			
	DBITEMPS			
84 (54)	SPCMAX	DPFP	8	Budget amount— floating point
92 (5C)	SPCCUR	DPFP	8	Current allocation—floating point
100 (64)	SPCHWM	DPFP	8	High-water mark—floating point
108 (6C)	SPCFLAGS SPCPERM SPCNONE	HEX EQU EQU	1 X'80' X'40'	Space flags Add to perm Do not track
109 (6D)		HEX	3	Slack
	* * VSAM TOTALS *			
	DBIVSAMS			
112 (70)	SPCMAX	DPFP	8	Budget amount — floating point
120 (78)	SPCCUR	DPFP	8	Current allocation — floating point
128 (80)	SPCHWM	DPFP	8	High-water mark —floating point
136 (88)	SPCFLAGS SPCPERM SPCNONE	HEX EQU EQU	1 X'80' X'40'	Space flags Add to perm Do not track
137 (89)		HEX	3	Slack
	* * HSM TOTALS *			
	DBIHSMS			
140 (8C)	SPCFILL	HEX	8	Filler
148 (94)	SPCCUR	DPFP	8	Current allocation — floating point
156 (9C)	SPCHWM	DPFP	8	High-water mark — floating point
164 (A4)	SPCFLAGS SPCPERM SPCNONE	HEX EQU EQU	1 X'80' X'40'	Space flags Add to perm Do not track
165 (A5)		HEX	3	Slack
168 (A8)	DBIWARN	HEX	4	Warning percent
172 (AC)	DBINAME	CHAR	20	Application descriptive name

**Table A-2 Application Record Format (Part 3 of 3)**

Offset	Name	Type	LEN	Description
192 (C0)	DBIUSRF1	CHAR	8	User field 1
200 (C8)	DBIUSRF2	CHAR	10	User field 2
210 (D2)	DBIUSRF3	CHAR	10	User field 3
220 (DC)	DBICRTDT	HEX	4	Create date CCYYMMDD
224 (E0)	DBIDLUPD	HEX	4	Date last updated CCYYMMDD
228 (E4)		HEX	27	Slack

## Global Work Area

The Global Work Area is described in member BUDWORK in the Installation Control library *?prefix.BBSAMP*.

## Special Considerations

The exits must be linked into individual load module. FLST and RLST parameters are loaded into CSA or ECSA when SVOS is started or the global parameters are refreshed. If you have MVS/XA or MVS/ESA and have written the exits in 31-bit mode, the module is loaded into ECSA. SG-Control does *not* have to be stopped and restarted to change any of the exits.

Since the exits may be called from SG-Control code executing in the DADSM pre- and postprocessing exits, the following restrictions must be observed:

- Runs in supervisor state in key 10
- Must be coded and marked as reentrant by the linkage editor
- Must be entered in 31-bit mode (MVS/XA systems only)
- Cannot allocate a file dynamically
- Cannot issue SVCs that ENQ on the VTOC, like LSPACE
- Cannot issue SVCs that do catalog lookups that might cause a volume to be allocated

**Note:** If the security exit is used to control access to the database from the ISPF interface, the security exit also must be coded to run in problem state, key 8.

In an MVS/XA 2.2 or higher environment, certain Scheduler control blocks (JCT, ACT, SCT, SIOT, JFCB) must be accessed through the use of the SWA Manager. Pointers to these control blocks may contain a SWA Virtual Address (SVA) or the TTR pointer on disk. The comments in the global work area contain the characters SVA for those fields that should use the SWA Manager. SG-Control has routines to read and write SVA fields.

To use the SWA read routine, load R15 with GWASWAR and R1 with the SVA of the control block. R1 is returned with the address of a copy of the control block. The SWA write routine can be used to rewrite a modified Scheduler control block. To use this routine load:

- R15 with GWASWAW
- R0 with the virtual address of the control block
- R1 with the SVA of the control block to be rewritten

Users not running in a MVS/XA 2.2 environment may still want to use the SWA manager to provide upward compatibility.

---

---

# Appendix B Special Functions of the TSO BUDGET Command

This appendix discusses how the BUDGET command is used to change current allocation amounts or high-water marks.

In general, you should never need to change the allocation amounts. The SG-Control system provides the RESYNC program to rebuild the SG-Control database. These extensions to the TSO BUDGET command offer you the ability to change the allocation amounts without the overhead of a RESYNC.

**Note:** The use of the following commands should be monitored closely. For that reason, they are documented separately from the ADD/CHANGE description of the BUDGET command.

As with the standard functions, the BUDGET command can change a single application or be used to make mass changes. The format to change a single application follows.

---

```
BUDGET acct-code CHANGE
CURPERM(numeric)
CURTEMP(numeric)
CURVSAM(numeric)
CURHSM(numeric)
HWMHSM(numeric)
HWMPERM(numeric)
HWMTEMP(numeric)
HWMVSAM(numeric)
```

---

## Command Parameters

- **acct-code**—Required parameter that specifies the budget application to be changed.

- **CURPERM(numeric)**—Optional parameter that changes the amount of space allocated to permanent data sets. This field may include temporary and VSAM space.
- **CURTEMP(numeric)**—Optional parameter that changes the amount of space allocated to temporary data sets.
- **CURVSAM(numeric)**—Optional parameter that changes the amount of space allocated to VSAM data sets.
- **CURHSM(numeric)**—Optional parameter that changes the amount of space allocated to DFHSM migrate and backup data sets.
- **HWMHSM(numeric)**—Optional parameter that changes the high-water mark for DFHSM migrate and backup data sets.
- **HWMPERM(numeric)**—Optional parameter that changes the high-water mark for permanent data sets.
- **HWMTEMP(numeric)**—Optional parameter that changes the high-water mark for temporary data sets.
- **HWMVSAM(numeric)**—Optional parameter that changes the high-water mark for VSAM data sets.

#### Example

To change the amount of space allocated to permanent data sets for application ACCTA, type:

**BUDGET ACCTA CHANGE CURPERM(123456)**

To update multiple applications, you can select applications by comparing a field to a value. The format used to compare a field to a value is

**Note:** Parameters previously defined are not repeated here.

```
BUDGET (field1,operator,value)
      CHANGE  CURPERM(numeric)
            CURTEMP( numeric)
            CURVSAM( numeric)
            CURHSM( numeric)
            HWMPERM( numeric)
            HWMTEMP( numeric)
            HWMVSAM( numeric)
            HWMHSM( numeric)
```

---

### Example

To set the amount of permanent space to zero in all applications with an application code starting with ACCT, type:

```
BUDGET (ACCT,EQ,'ACCT*') CHANGE CURPERM(0)
```

You can also select applications by comparing the contents of two fields within an application. The format for this variation of the BUDGET command is

---

```
BUDGET (field1,operator,field2)
      CHANGE  CURPERM(numeric)
           CURTEMP(numeric)
           CURVSAM(numeric)
           CURHSM(numeric)
           HWMPERM(numeric)
           HWMTEMP(numeric)
           HWMVSAM(numeric)
           HWMHSM(numeric)
```

---

### Example

To set the amount of currently used VSAM space to zero in all applications where user field2 matches user field3, type:

```
BUDGET (USRF2,EQ,USRF3) CHANGE CURVSAM(0)
```



---

---

# Appendix C SG-Control Error Resolution

Following is a list of items that may be needed by the BMC Software Customer Support staff so that they may answer your questions or resolve a problem with SG-Control. All of these items are not needed for any one call, but knowing where to get the information, having the JCL ready to generate the requested reports, and being familiar with the relevant facilities can help BMC Software provide you with the service you need in a timely and efficient manner.

Please have the following information available:

- SG-Control version, release, and maintenance level. This can be obtained by browsing the SGC load module in the load library containing the system or by issuing the following command to the SVOS:

```
/STATUS VER
```

*OR*

```
F SVOS,STATUS VER
```

- A list of any error resolution or customization zaps you have applied to SG-Control.
- OS/390 and DFP version and release. This can be obtained from your systems programming staff.

Have available any relevant job output.

- The output from a job contains any related messages. Make sure you have a copy of the output available for BMC Software Customer Support. If the situation has occurred a number of times, have available a reasonable number of copies.

How often does this occur?

- If the situation has just started, attempt to identify any changes that have been made to the operating system, to SG-Control, the FLST and RLST parameters, or user exits.

Try to find a pattern and develop a job stream that recreates the situation.

## Technical Problems

Did a SG-Control moduleabend? Read the “SGCDUMP DD Statement” on page C-3. Be prepared to generate and format a system dump. Dumps should be forwarded to BMC Software on tape or paper.

### **Reporting and dumping the contents of the SG-Control database**

The REPORT function of the SGCMAINT utility is used to produce formatted reports and dump the contents of the database control record (DBC) and application records (DBI). Sample JCL for SGCMAINT is supplied in the installation job stream.

The PRINT function of the IBM IDCAMS utility can be used to print the contents of selected database records. Review the SKIP and COUNT parameters of the IDCAMS print command.

The IDCAMS REPRO command can be used to copy a database to tape.

## Fax Procedures

Faxes are received by BMC Software Customer Support 24 hours a day, 7 days a week.

If this is a new contact, address the fax as follows:

New Contact  
Level 1 Support  
SG-Control  
BMC Software

Include your name, telephone number, company name, and a detailed description of the problem. Please indicate the severity of the problem.

Faxes associated with existing contacts should be addressed to the BMC Software support person currently handling your problem.

The BMC Software Customer Support Fax number is (770) 785-2074.

## Reserved DD Statements

SG-Control contains a number of diagnostic aids designed to assist in the identification and resolution of problems. These diagnostic aids are activated by adding DD statements with reserved DD names to the step JCL in the end user's job stream.

The dump facility (activated by the inclusion of a SGCDUMP DD statement) can be used to take a snap dump at a selected point in SG-Control processing, produce a module trace showing how each program is entered, or take a snapshot dump when an abend occurs.

### SGCTRACE DD Statement

The application update tracing facility can be activated by including a SGCTRACE DD statement in the step JCL in the end user's job stream. This facility was designed to allow a users to trace how applications are being updated.

#### Syntax

```
//SGCTRACE DD DUMMY
```

### SGCDUMP DD Statement

The SG-Control dump facility produces dumps and traces for the BMC Software Customer Support staff. This facility should *not* be used unless requested by the Customer Support staff.

#### Syntax

```
//SGCDUMP DD DUMMY, VOL=SER=(TRACE, module-id, system),  
// SPACE=(TRK, (call-count, mod-count))
```

Variable	Description
<i>module-id</i>	represents the last four digits of the module name
<i>call-count</i>	represents the number of times SG-Control is called for this jobstep before a dump can be taken
<i>mod-count</i>	represents the number of times a module is called before a dump can be taken
<i>system</i>	required parameter

**Note:** Call-count must be satisfied before mod-count can be applied.

#### Example

To invoke the module tracing facility, add the following DD statement:

```
//SGCDUMP DD DUMMY,VOL=SER=TRACE,UNIT=SYSALLDA
```

#### Example

To take a snapshot dump when an abend occurs in SG-Control, add the following DD statement:

```
//SGCDUMP DD DUMMY
```

#### Example

To take an SVC dump when an abend occurs in SG-Control, add the following DD statement:

```
//SGCDUMP DD DUMMY, VOL=SER=(AAA,BBB,SYSTEM)
```

#### Example

To request an abend in module BUD0020, add the following DD statement:

```
//SGCDUMP DD DUMMY,VOL=SER=(TRACE,0020,SYSTEM)
```

**Example**

To request an abend in module BUD0035 on the third call to the DADSM exit during the second pass through BUD0035, add the following DD statement:

```
//SGCDUMP DD DUMMY,VOL=SER=(TRACE,0035,SYSTEM),  
//          SPACE=(TRK,(3,2))
```



---

---

## Appendix D    **SGCDBCNV Utility**

An application record should not be used by a *prior* release once it has been migrated to the release 6.1 format. If you want to delay altering programs and procedures that produce reports from data read directly from an SG-Control database, you can use the **SGCDBCNV** utility. The output of the utility is a sequential file to be used in batch reporting and is unsuitable for collection.

SGCDBCNV copies an SG-Control 6.1 database and converts the records back to 5.1 format during the copy. You may then execute the procedures and programs against the copy of the database to produce reports.

A copy database created by SGCDBCNV should be used only for purposes of reporting, and it should *not* be updated by SG-Control. Also SGCDBCNV should *not* be used to make a back-up copy of a database.

A sample of the SGCDBCNV utility is provided in *?prefix.BBSAMP*. Carefully follow the user instructions in the utility.



---

---

# Glossary

This glossary defines BMC Software terminology. Other dictionaries and glossaries can be used in conjunction with this glossary.

Since this glossary pertains to BMC Software-related products, some of the terms defined might not appear in this book.

To help you find the information you need, this glossary uses the following cross-references:

*Contrast with* indicates a term that has a contrary or contradictory meaning.

*See* indicates an entry that is a synonym or contains expanded information.

*See also* indicates an entry that contains related information.

<b>action</b>	Defined operation, such as modifying a MAINVIEW window, that is performed in response to a command. <i>See</i> object.
<b>active window</b>	Any MAINVIEW window in which data can be refreshed. <i>See</i> alternate window, current window, window.
<b>administrative view</b>	Display from which a product's management tasks are performed, such as the DSLIST view for managing historical data sets. <i>See</i> view.
<b>ALT WIN field</b>	Input field that allows you to specify the window identifier for an alternate window where the results of a hyperlink are displayed. <i>See</i> alternate window.
<b>Alternate Access</b>	<i>See</i> MAINVIEW Alternate Access.
<b>alternate form</b>	View requested through the FORM command that changes the format of a previously displayed view to show related information. <i>See also</i> form, query.

---

<b>alternate window</b>	(1) Window that is specifically selected to display the results of a hyperlink. (2) Window whose identifier is defined to the ALT WIN field. <i>Contrast with</i> current window. <i>See</i> active window, window, ALT WIN field.
<b>analyzer</b>	(1) Online display that presents a snapshot of status and activity data and indicates problem areas. (2) Component of CMF MONITOR. <i>See</i> CMF MONITOR Analyzer.
<b>application</b>	(1) Program that performs a specific set of tasks within a MAINVIEW product. (2) In MAINVIEW VistaPoint, combination of workloads to enable display of their transaction performance data in a single view.
<b>application trace</b>	<i>See</i> trace.
<b>ASCH workload</b>	Workload comprising Advanced Program-to-Program Communication (APPC) address spaces.
<b>AutoCustomization</b>	Online facility for customizing the installation of products. AutoCustomization provides an ISPF panel interface that both presents customization steps in sequence and provides current status information about the progress of the installation.
<b>automatic screen update</b>	Usage mode wherein the currently displayed screen is refreshed automatically with new data at an interval you specify. Invoked by the ASU command.
<b>batch workload</b>	Workload consisting of address spaces running batch jobs.
<b>BBI</b>	Basic architecture that distributes work between workstations and multiple OS/390 targets for BMC Software MAINVIEW products.
<b>BBI-SS PAS</b>	<i>See</i> BBI subsystem product address space.
<b>BBI subsystem product address space (BBI-SS PAS)</b>	OS/390 subsystem address space that manages communication between local and remote systems and that contains one or more of the following products: <ul style="list-style-type: none"> <li>• MAINVIEW AutoOPERATOR</li> <li>• MAINVIEW for CICS</li> <li>• MAINVIEW for DB2</li> <li>• MAINVIEW for DBCTL</li> <li>• MAINVIEW for IMS Online</li> <li>• MAINVIEW for MQSeries (formerly Command MQ for S/390)</li> <li>• MAINVIEW SRM</li> <li>• MAINVIEW VistaPoint (for CICS, DB2, DBCTL, and IMS workloads)</li> </ul>
<b>BBPARM</b>	<i>See</i> parameter library.

---

<b>BBPROC</b>	<i>See</i> procedure library.
<b>BBPROF</b>	<i>See</i> profile library.
<b>BBSAMP</b>	<i>See</i> sample library.
<b>BBV</b>	<i>See</i> MAINVIEW Alternate Access.
<b>BBXS</b>	BMC Software Subsystem Services. Common set of service routines loaded into common storage and used by several BMC Software MAINVIEW products.
<b>border</b>	Visual indication of the boundaries of a window.
<b>bottleneck analysis</b>	Process of determining which resources have insufficient capacity to provide acceptable service levels and that therefore can cause performance problems.
<b>CA-Disk</b>	Data management system by Computer Associates that replaced the DMS product.
<b>CAS</b>	Coordinating address space. One of the address spaces used by the MAINVIEW windows environment architecture. The CAS supplies common services and enables communication between linked systems. Each OS/390 or z/OS image requires a separate CAS. Cross-system communication is established through the CAS using VTAM and XCF communication links.
<b>CFMON</b>	<i>See</i> coupling facility monitoring.
<b>chart</b>	Display format for graphical data. <i>See also</i> graph.
<b>CICSplex</b>	User-defined set of one or more CICS systems that are controlled and managed as a single functional entity.
<b>CMF MONITOR</b>	Comprehensive Management Facility MONITOR. Product that measures and reports on all critical system resources, such as CPU, channel, and device usage; memory, paging, and swapping activity; and workload performance.
<b>CMF MONITOR Analyzer</b>	Batch component of CMF MONITOR that reads the SMF user and 70 series records created by the CMF MONITOR Extractor and/or the RMF Extractor and formats them into printed system performance reports.
<b>CMF MONITOR Extractor</b>	Component of CMF that collects performance statistics for CMF MONITOR Analyzer, CMF MONITOR Online, MAINVIEW for OS/390, and RMF postprocessor. <i>See</i> CMF MONITOR Analyzer, CMF MONITOR Online, MAINVIEW for OS/390.

---

**CMF MONITOR Online**

Component of CMF that uses the MAINVIEW window interface to present data on all address spaces, their use of various system resources, and the delays that each address space incurs while waiting for access to these resources. *See* CMF MONITOR, MAINVIEW for OS/390.

**CMF Type 79 API**

Application programming interface, provided by CMF, that provides access to MAINVIEW SMF-type 79 records.

**CMFMON**

Component of CMF MONITOR that simplifies online retrieval of information about system hardware and application performance and creates MAINVIEW SMF-type 79 records.

The CMFMON *online facility* can be used to view data in one or more formatted screens.

The CMFMON *write facility* can be used to write collected data as MAINVIEW SMF-type 79 records to an SMF or sequential data set.

**CMRDETL**

MAINVIEW for CICS data set that stores detail transaction records (type 6E) and abend records (type 6D). Detail records are logged for each successful transaction. Abend records are written when an abend occurs. Both records have the same format when stored on CMRDETL.

**CMRSTATS**

MAINVIEW for CICS data set that stores both CICS operational statistic records, at five-minute intervals, and other records, at intervals defined by parameters specified during customization (using CMRSOPT).

**column**

Vertical component of a view or display, typically containing fields of the same type of information, that varies by the objects associated in each row.

**collection interval**

Length of time data is collected. *See also* delta mode, total mode.

**command delimiter**

Special character, usually a ; (semicolon), used to stack commands typed concurrently on the COMMAND line for sequential execution.

**COMMAND line**

Line in the control area of the display screen where primary commands can be typed. *Contrast with* line command column.

**Command MQ Automation D/S**

Command MQ agents, which provide local proactive monitoring for both MQSeries and MSMQ (Microsoft message queue manager). The Command MQ agents operate at the local node level where they continue to perform functions regardless of the availability of the MQM (message queue manager) network. Functionality includes automatic monitoring and restarts of channels, queue managers, queues and command servers. In cases where automated recovery is not possible, the agents transport critical alert information to a central console.

---

### **Command MQ Automation S/390**

Command MQ component, which monitors the MQM (message queue manager) networks and intercedes to perform corrective actions when problems arise. Solutions include:

- Dead-Letter Queue management
- System Queue Archival
- Service Interval Performance solutions
- Channel Availability

These solutions help ensure immediate relief to some of the most pressing MQM operations and performance problems.

### **Command MQ for D/S**

Command MQ for D/S utilizes a true client/server architecture and employs resident agents to provide configuration, administration, performance monitoring and operations management for the MQM (message queue manager) network.

### **Command MQ for S/390**

*See* MAINVIEW for MQSeries.

### **COMMON STORAGE MONITOR**

Component of MAINVIEW for OS/390 that monitors usage and reconfigures OS/390 or z/OS common storage blocks.

### **composite workload**

Workload made up of a WLM workload or other workloads, which are called *constituent workloads*.

### **constituent workload**

Member of a composite workload. Constituent workloads in a composite usually belong to a single workload class, but sometimes are mixed.

### **contention**

Occurs when there are more requests for service than there are servers available.

### **context**

In a Plex Manager view, field that contains the name of a target or group of targets specified with the CONTEXT command. *See* scope, service point, SSI context, target context.

### **CONTEXT command**

Specifies either a MAINVIEW product and a specific target for that product (*see* target context) or a MAINVIEW product and a name representing one or more targets (*see* SSI context) for that product.

---

**control statement** (1) Statement that interrupts a sequence of instructions and transfers control to another part of the program. (2) Statement that names samplers and other parameters that configure the MAINVIEW components to perform specified functions. (3) In CMF MONITOR, statement in a parameter library member used to identify a sampler in the extractor or a report in the analyzer, or to describe either component's processing requirements to the operating system.

**coupling facility monitoring (CFMON)**

Coupling facility views that monitor the activity of your system's coupling facilities.

**current data**

Data that reflects the system in its current state. The two types of current data are real-time data and interval data. *Contrast with* historical data. *See also* interval data, real-time data.

**current window**

In the MAINVIEW window environment, window where the main dialog with the application takes place. The current window is used as the default window destination for commands issued on the COMMAND line when no window number is specified. *Contrast with* alternate window. *See* active window, window.

**DASD**

(Direct Access Storage Device) (1) A device with rotating recording surfaces that provides immediate access to stored data. (2) Any device that responds to a DASD program.

**DASD ADVISOR**

An interactive software tool that diagnoses DASD performance problems and makes recommendations to reduce overall service time. This tool measures and reports on the operational performance of IBM and IBM-compatible devices.

**data collector**

Program that belongs to a MAINVIEW product and that collects data from various sources and stores the data in records used by views. For example, MAINVIEW for OS/390 data collectors obtain data from OS/390 or z/OS services, OS/390 or z/OS control blocks, CMF MONITOR Extractor control blocks, and other sources. *Contrast with* extractor.

**delta mode**

(1) In MAINVIEW for DB2 analyzer displays, difference between the value sampled at the start of the current statistics interval and the value sampled by the current analyzer request. *See also* statistics interval. (2) In CMFMON, usage mode wherein certain columns of data reflect the difference in values between one sample cycle and the next. Invoked by the DELta ON command. *See also* collection interval, sample cycle, total mode.

**DFSMS**

(Data Facility Storage Management System) Data management, backup, and HSM software from IBM for OS/390 or z/OS mainframes.

**DMR**

*See* MAINVIEW for DB2.

---

<b>DMS</b>	(Data Management System) <i>See</i> CA-Disk.
<b>DMS2HSM</b>	<i>See</i> MAINVIEW SRM DMS2HSM.
<b>DSO</b>	(Data Set Optimizer) CMF MONITOR Extractor component that uses CMF MONITOR Extractor data to produce reports specifying the optimal ordering of data sets on moveable head devices.
<b>EasyHSM</b>	<i>See</i> MAINVIEW SRM EasyHSM.
<b>EasyPOOL</b>	<i>See</i> MAINVIEW SRM EasyPOOL.
<b>EasySMS</b>	<i>See</i> MAINVIEW SRM EasySMS.
<b>element</b>	(1) Data component of a data collector record, shown in a view as a field. (2) Internal value of a field in a view, used in product functions.
<b>element help</b>	Online help for a field in a view. The preferred term is <i>field help</i> .
<b>Enterprise Storage Automation</b>	<i>See</i> MAINVIEW SRM Enterprise Storage Automation.
<b>event</b>	A message issued by Enterprise Storage Automation. User-defined storage occurrences generate events in the form of messages. These events provide an early warning system for storage problems and are routed to user-specified destinations for central viewing and management.
<b>Event Collector</b>	Component for MAINVIEW for IMS Online, MAINVIEW for IMS Offline, and MAINVIEW for DBCTL that collects data about events in the IMS environment. This data is required for Workload Monitor and optional for Workload Analyzer (except for the workload trace service). This data also is recorded as transaction records (X'FA') and program records (X'F9') on the IMS system log for later use by the MAINVIEW for IMS Offline components: Performance Reporter and Transaction Accountant.
<b>expand</b>	Predefined link from one display to a related display. <i>See also</i> hyperlink.
<b>extractor</b>	Program that collects data from various sources and keeps the data control blocks to be written as records. Extractors obtain data from services, control blocks, and other sources. <i>Contrast with</i> data collector.
<b>extractor interval</b>	<i>See</i> collection interval.
<b>fast path</b>	Predefined link between one screen and another. To use the fast path, place the cursor on a single value in a field and press <b>Enter</b> . The resulting screen displays more detailed information about the selected value. <i>See also</i> hyperlink.

---

<b>field</b>	Group of character positions within a screen or report used to type or display specific information.
<b>field help</b>	Online help describing the purpose or contents of a field on a screen. To display field help, place the cursor anywhere in a field and press <b>PF1</b> (HELP). In some products, field help is accessible from the screen help that is displayed when you press <b>PF1</b> .
<b>filter</b>	Selection criteria used to limit the number of rows displayed in a view. Data that does not meet the selection criteria is not displayed. A filter is composed of an element, an operator, and an operand (a number or character string). Filters can be implemented in view customization, through the PARM/QPARM commands, or through the Where/QWhere commands. Filters are established against elements of data.
<b>fire</b>	The term used to indicate that an event has triggered an action. In MAINVIEW AutoOPERATOR, when a rule selection criteria matches an incoming event and <i>fires</i> , the user-specified automation actions are performed. This process is also called <i>handling</i> the event.
<b>fixed field</b>	Field that remains stationary at the left margin of a screen that is scrolled either right or left.
<b>FOCAL POINT</b>	MAINVIEW product that displays a summary of key performance indicators across systems, sites, and applications from a single terminal.
<b>form</b>	One of two constituent parts of a view; the other is query. A form defines how the data is presented; a query identifies the data required for the view. <i>See also</i> query, view.
<b>full-screen mode</b>	Display of a MAINVIEW product application or service on the entire screen. There is no window information line. <i>Contrast with</i> windows mode.
<b>global command</b>	Any MAINVIEW window interface command that can affect all windows in the window area of a MAINVIEW display.
<b>graph</b>	Graphical display of data that you select from a MAINVIEW window environment view. <i>See also</i> chart.
<b>hilevel</b>	For MAINVIEW products, high-level data set qualifier required by a site's naming conventions.
<b>historical data</b>	(1) Data that reflects the system as it existed at the end of a past recording interval or the duration of several intervals. (2) Any data stored in the historical database and retrieved using the TIME command. <i>Contrast with</i> current data, interval data and real-time data.

---

<b>historical database</b>	Collection of performance data written at the end of each installation-defined recording interval and containing up to 100 VSAM clusters. Data is extracted from the historical database with the TIME command. <i>See</i> historical data.
<b>historical data set</b>	In MAINVIEW products that display historical data, VSAM cluster file in which data is recorded at regular intervals.
<b>HSM</b>	(Hierarchical Storage Management) Automatic movement of files from hard disk to slower, less-expensive storage media. The typical hierarchy is from magnetic disk to optical disk to tape.
<b>hyperlink</b>	<p>(1) Preset field in a view or an EXPAND line on a display that permits you to</p> <ul style="list-style-type: none"> <li>• access cursor-sensitive help</li> <li>• issue commands</li> <li>• link to another view or display</li> </ul> <p>The transfer can be either within a single product or to a related display/view in a different BMC Software product. Generally, hyperlinked fields are highlighted. (2) Cursor-activated short path from a topic or term in online help to related information. <i>See also</i> fast path.</p>
<b>Image log</b>	<p>Collection of screen-display records. Image logs can be created for both the BBI-SS PAS and the BBI terminal session (TS).</p> <p>The BBI-SS PAS Image log consists of two data sets that are used alternately: as one fills up, the other is used. Logging to the BBI-SS PAS Image log stops when both data sets are filled and the first data set is not processed by the archive program.</p> <p>The TS Image log is a single data set that wraps around when full.</p>
<b>IMSplex System Manager (IPSM)</b>	MVIMS Online and MVDBC service that provides Single System Image views of resources and bottlenecks for applications across one or more IMS regions and systems.
<b>interval data</b>	<p>Cumulative data collected during a collection interval. Intervals usually last from 15 to 30 minutes depending on how the recording interval is specified during product customization. <i>Contrast with</i> historical data.</p> <p><b>Note:</b> If change is made to the workloads, a new interval will be started.</p> <p><i>See also</i> current data and real-time data.</p>
<b>InTune</b>	Product for improving application program performance. It monitors the program and provides information used to reduce bottlenecks and delays.

---

<b>IRUF</b>	IMS Resource Utilization File (IRUF). IRUFs can be either detail (one event, one record) or summarized (more than one event, one record). A detail IRUF is created by processing the IMS system log through a program called IMFLEEDIT. A summarized IRUF is created by processing one or more detail IRUFs, one or more summarized IRUFs, or a combination of both, through a sort program and the TASCOSTR program.
<b>job activity view</b>	Report about address space consumption of resources. <i>See</i> view.
<b>journal</b>	Special-purpose data set that stores the chronological records of operator and system actions.
<b>Journal log</b>	<p>Collection of messages. Journal logs are created for both the BBI-SS PAS and the BBI terminal session (TS).</p> <p>The BBI-SS PAS Journal log consists of two data sets that are used alternately: as one fills up, the other is used. Logging to the BBI-SS PAS Journal log stops when both data sets are filled and the first data set is not being processed by the archive program.</p> <p>The TS Journal log is a single data set that wraps around when full.</p>
<b>line command</b>	Command that you type in the line command column in a view or display. Line commands initiate actions that apply to the data displayed in that particular row.
<b>line command column</b>	Command input column on the left side of a view or display. <i>Contrast with</i> COMMAND line.
<b>Log Edit</b>	In the MAINVIEW for IMS Offline program named IMFLEEDIT, function that extracts transaction (X'FA') and program (X'F9') records from the IMS system log. IMFLEEDIT also extracts certain records that were recorded on the system log by IMS. IMFLEEDIT then formats the records into a file called the IMS Resource Utilization File (IRUF).
<b>MAINVIEW</b>	BMC Software integrated systems management architecture.
<b>MAINVIEW Alarm Manager (MV ALARM)</b>	In conjunction with other MAINVIEW products, notifies you when an exception occurs. MAINVIEW Alarm Manager is capable of monitoring multiple systems simultaneously, which means that MAINVIEW Alarm Manager installed on one system keeps track of your entire sysplex. You can then display a single view that shows exceptions for all MAINVIEW performance monitors within your OS/390 or z/OS enterprise.

---

**MAINVIEW Alternate Access**

Enables MAINVIEW products to be used without TSO by providing access through EXCP and VTAM interfaces.

**MAINVIEW Application Program Interface (MVAPI)**

A CLIST- or REXX-based, callable interface that allows MAINVIEW AutoOPERATOR EXECs to access MAINVIEW monitor product view data.

**MAINVIEW AutoOPERATOR**

Product that uses tools, techniques, and facilities to automate routine operator tasks and provide online performance monitoring, and that achieves high availability through error minimization, improved productivity, and problem prediction and prevention.

**MAINVIEW control area**

In the MAINVIEW window environment, first three lines at the top of the view containing the window information line and the COMMAND, SCROLL, CURR WIN, and ALT WIN lines. The control area cannot be customized and is part of the information display. *Contrast with* MAINVIEW display area, MAINVIEW window area.

**MAINVIEW Desktop** Version of the MAINVIEW window interface designed to run on OS/2 and Windows workstations.

**MAINVIEW display area**

*See* MAINVIEW window area.

**MAINVIEW Explorer** Product that provides access to MAINVIEW products from a Web browser running under Windows. MAINVIEW Explorer replaces MAINVIEW Desktop.

**MAINVIEW for CICS** Product (formerly MV MANAGER for CICS) that provides real-time application performance analysis and monitoring for CICS system management.

**MAINVIEW for DB2** Product (formerly MV MANAGER for DB2) that provides real-time and historical application performance analysis and monitoring for DB2 subsystem management.

**MAINVIEW for DBCTL (MVDBC)**

Product that provides real-time application performance analysis and monitoring for DBCTL management.

**MAINVIEW for IMS (MVIMS) Offline**

Product with a Performance Reporter component that organizes data and prints reports used to analyze IMS performance and a Transaction Accountant component that produces cost accounting and user charge-back records and reports.

---

**MAINVIEW for IMS (MVIMS) Online**

Product that provides real-time application performance analysis and monitoring for IMS management.

**MAINVIEW for IP**

Product that monitors OS/390 and z/OS mission-critical application performance as it relates to TCP/IP stack usage. Collected data includes availability, connections, response times, routers, service levels, storage, traffic, Web cache, and so on.

**MAINVIEW for Linux–Servers**

Product that allows you to monitor the performance of your Linux systems from the MAINVIEW windows interface.

**MAINVIEW for MQSeries (formerly known as Command MQ for S/390)**

Delivers comprehensive capabilities for configuration, administration, performance monitoring and operations management for an entire MQM (message queue manager) network.

**MAINVIEW for OS/390**

System management application (formerly known as MAINVIEW for MVS prior to version 2.5). Built upon the MAINVIEW window environment architecture, it uses the window interface to provide access to system performance data and other functions necessary in the overall management of an enterprise.

**MAINVIEW for UNIX System Services**

System management application that allows you to monitor the performance of the Unix System Services from a MAINVIEW window interface.

**MAINVIEW for VTAM**

Product that displays application performance data by application, transaction ID, and LU name. This collected data includes connections, response time statistics, application availability, and application throughput.

**MAINVIEW for WebSphere Application Server (formerly known as MAINVIEW for WebSphere)**

Product that provides extensive monitoring for the IBM WebSphere Application Server for z/OS and OS/390 environment.

**MAINVIEW Selection Menu**

ISPF selection panel that provides access to all MAINVIEW windows-mode and full-screen mode products.

**MAINVIEW SRM**

*See* MAINVIEW Storage Resource Manager (SRM).

**MAINVIEW SRM DMS2HSM**

Product that facilitates the conversion of CA-Disk, formerly known as DMS, to HSM.

---

**MAINVIEW SRM EasyHSM**

Product that provides online monitoring and reporting to help storage managers use DFHSM efficiently.

**MAINVIEW SRM EasyPOOL**

Product that provides control over data set allocation and enforcement of allocation and naming standards. EasyPOOL functions operate at the operating system level to intercept normal job processing, thus providing services without any JCL changes.

**MAINVIEW SRM EasySMS**

Product that provides tools that aid in the conversion to DFSMS and provides enhancement to the DFSMS environment after implementation. EasySMS consists of the EasyACS functions, the SMSACSTE function, and the Monitoring and Positioning Facility.

**MAINVIEW SRM Enterprise Storage Automation**

Product that delivers powerful event generation and storage automation technology across the storage enterprise. Used in conjunction with MAINVIEW AutoOPERATOR, automated solutions to perform pool, volume, application, or data set-level manipulation can be created and used in response to any condition or invoked to perform ad hoc requests.

**MAINVIEW SRM SG-Auto**

Product that provides early warning notification of storage anomalies and automated responses to those anomalies based on conditions in the storage subsystem.

**MAINVIEW SRM SG-Control**

Product that provides real-time monitoring, budgeting, and control of DASD space utilization.

**MAINVIEW SRM StopX37/II**

Product that provides enhancements to OS/390 or z/OS space management, reducing the incidence of space-related processing problems. The StopX37/II functions operate at the system level to intercept abend conditions or standards violations, thus providing services without any JCL changes.

**MAINVIEW SRM StorageGUARD**

Product that monitors and reports on DASD consumption and provides historical views to help control current and future DASD usage.

**MAINVIEW Storage Resource Manager (SRM)**

Suite of products that assist in all phases of OS/390 or z/OS storage management. MAINVIEW SRM consists of products that perform automation, reporting, trend analysis, and error correction for storage management.

---

## **MAINVIEW SYSPROG Services**

*See SYSPROG services.*

## **MAINVIEW VistaPoint**

Product that provides enterprise-wide views of performance. Application and workload views are available for CICS, DB2, DBCTL, IMS, OS/390, or z/OS. Data is summarized at the level of detail needed; for example, views can be for a single target, an OS/390 or z/OS image, or an entire enterprise.

## **MAINVIEW window area**

Portion of the information display that is not the control area and in which views are displayed and windows opened. It includes all but the first three lines of the information display. *Contrast with* MAINVIEW control area.

## **monitor**

Online service that measures resources or workloads at user-defined intervals and issues warnings when user-defined thresholds are exceeded.

## **Multi-Level Automation (MLA)**

The user-defined, multiple step process in Enterprise Storage Automation that implements solutions in a tiered approach, where solutions are invoked one after another until the condition is resolved.

## **MVALARM**

*See* MAINVIEW Alarm Manager.

## **MVAPI**

*See* MAINVIEW Application Program Interface.

## **MVCICS**

*See* MAINVIEW for CICS.

## **MVDB2**

*See* MAINVIEW for DB2.

## **MVDBC**

*See* MAINVIEW for DBCTL.

## **MVIMS**

*See* MAINVIEW for IMS.

## **MVIP**

*See* MAINVIEW for IP.

## **MVLNX**

*See* MAINVIEW for Linux–Servers.

## **MVMQ**

*See* MAINVIEW for MQSeries.

## **MVMVS**

*See* MAINVIEW for OS/390.

## **MVScope**

MAINVIEW for OS/390 application that traces both CPU usage down to the CSECT level and I/O usage down to the channel program level.

## **MVSRM**

*See* MAINVIEW Storage Resource Manager (SRM).

## **MVSRMHSM**

*See* MAINVIEW SRM EasyHSM.

---

<b>MVSRMSGC</b>	<i>See</i> MAINVIEW SRM SG-Control.
<b>MVSRMSGD</b>	<i>See</i> MAINVIEW SRM StorageGUARD.
<b>MVSRMSGP</b>	<i>See</i> MAINVIEW SRM StorageGUARD.
<b>MVUSS</b>	<i>See</i> MAINVIEW for UNIX System Services.
<b>MVVP</b>	<i>See</i> MAINVIEW VistaPoint.
<b>MVVTAM</b>	<i>See</i> MAINVIEW for VTAM.
<b>MVWEB</b>	<i>See</i> MAINVIEW for WebSphere Application Server.
<b>nested help</b>	Multiple layers of help pop-up windows. Each successive layer is accessed by clicking a hyperlink from the previous layer.
<b>object</b>	<p>Anything you can manipulate as a single unit. MAINVIEW objects can be any of the following: product, secondary window, view, row, column, or field.</p> <p>You can issue an action against an object by issuing a line command in the line command column to the left of the object. <i>See</i> action.</p>
<b>OMVS workload</b>	Workload consisting of OS/390 OpenEdition address spaces.
<b>online help</b>	Help information that is accessible online.
<b>OS/390 and z/OS Installer</b>	BMC Software common installation system for mainframe products.
<b>OS/390 product address space (PAS)</b>	Address space containing OS/390 or z/OS data collectors, including the CMF MONITOR Extractor. Used by MAINVIEW for OS/390, MAINVIEW for UNIX System Services, and CMF MONITOR products. <i>See</i> PAS.
<b>parameter library</b>	<p>Data set consisting of members that contain parameters for specific MAINVIEW products or a support component There can be several versions:</p> <ul style="list-style-type: none"> <li>• the distributed parameter library, called BBPARM</li> <li>• a site-specific parameter library or libraries</li> </ul> <p>These can be</p> <ul style="list-style-type: none"> <li>• a library created by AutoCustomization, called UBBPARM</li> <li>• a library created manually, with a unique name</li> </ul>

---

<b>PAS</b>	Product address space. Used by the MAINVIEW products. Contains data collectors and other product functions. <i>See also</i> OS/390 product address space (PAS) <i>and</i> BBI subsystem product address space (BBI-SS PAS).
<b>performance group workload</b>	Collection of address spaces defined to OS/390 or z/OS. If you are running OS/390 or z/OS with WLM in compatibility mode, MAINVIEW for OS/390 creates a performance group workload instead of a service class.
<b>PERFORMANCE MANAGER</b>	MAINVIEW for CICS online service for monitoring and managing current performance of CICS regions.
<b>Performance Reporter (MVIMS)</b>	MVIMS Offline component that organizes data and prints reports that can be used to analyze IMS performance.
<b>Performance Reporter</b>	Product component that generates offline batch reports. The following products can generate these reports: <ul style="list-style-type: none"> <li>• MAINVIEW for DB2</li> <li>• MAINVIEW for CICS</li> </ul>
<b>Plex Manager</b>	Product through which cross-system communication, MAINVIEW security, and an SSI context are established and controlled. Plex Manager is shipped with MAINVIEW window environment products as part of the coordinating address space (CAS) and is accessible as a menu option from the MAINVIEW Selection Menu.
<b>pop-up display</b>	Full-screen panel that displays additional information about a selected event in a detail trace.
<b>pop-up window</b>	Window containing help information that, when active, overlays part of the window area. A pop-up window is displayed when you issue the HELP command while working in windows-mode.
<b>PRGP workload</b>	In MVS/SP 5.0 or earlier, or in compatibility mode in MVS/SP 5.1 or later, composite of service classes. MAINVIEW for OS/390 creates a performance group workload for each performance group defined in the current IEAIPS.xx member.
<b>procedure library</b>	Data set consisting of members that contain executable procedures used by MAINVIEW AutoOPERATOR. These procedures are execute command lists (EXECs) that automate site functions. There can be several versions: <ul style="list-style-type: none"> <li>• the distributed parameter library, called BBPROC</li> <li>• a site-specific parameter library or libraries</li> </ul>

---

These can be

- a library created by AutoCustomization, called UBBPROC
- a library created manually, with a unique name

The site-created EXECs can be either user-written or customized MAINVIEW AutoOPERATOR-supplied EXECs from BBPROC.

**product address space**

*See* PAS.

**profile library**

Data set consisting of members that contain profile information and cycle refresh definitions for a terminal session connected to a BBI-SS PAS. Other members are dynamically created by MAINVIEW applications. There can be several versions:

- the distributed profile library, called BBPROF
- a site-specific profile library or libraries

These can be

- a library created by AutoCustomization, called SBBPROF
- a library created manually, with a unique name

The site library is a common profile shared by all site users. The terminal session CLIST creates a user profile automatically if one does not exist; it is called `userid.BBPROF`, where `userid` is your logon ID. User profile libraries allow each user to specify unique PF keys, CYCLE commands, target system defaults, a Primary Option Menu, and a unique set of application profiles.

**query**

One of two constituent parts of a view; the other is form. A query defines the data for a view; a form defines the display format. *See also* form, view.

**real-time data**

Performance data as it exists at the moment of inquiry. Real-time data is recorded during the smallest unit of time for data collection. *Contrast with* historical data. *See also* current data and interval data.

**Resource Analyzer**

Online real-time displays used to analyze IMS resources and determine which are affected by specific workload problems.

**Resource Monitor**

Online data collection services used to monitor IMS resources and issue warnings when defined utilization thresholds are exceeded.

**row**

(1) Horizontal component of a view or display comprising all the fields pertaining to a single device, address space, user, and so on. (2) Horizontal component of a DB2 table consisting of a sequence of values, one for each column of the table.

---

<b>RxD2</b>	Product that provides access to DB2 from REXX. It provides tools to query the DB2 catalog, issue dynamic SQL, test DB2 applications, analyze EXPLAIN data, generate DDL or DB2 utility JCL, edit DB2 table spaces, perform security administration, and much more.
<b>sample cycle</b>	<p>Time between data samples.</p> <p>For the CMF MONITOR Extractor, this is the time specified in the extractor control statements (usually 1 to 5 seconds).</p> <p>For real-time data, the cycle is not fixed. Data is sampled each time you press <b>Enter</b>.</p>
<b>sample library</b>	<p>Data set consisting of members each of which contains one of the following items:</p> <ul style="list-style-type: none"> <li>• sample JCL that can be edited to perform specific functions</li> <li>• macro that is referenced in the assembly of user-written services</li> <li>• sample user exit routine</li> </ul> <p>There can be several versions:</p> <ul style="list-style-type: none"> <li>• the distributed sample library, called BBSAMP</li> <li>• a site-specific sample library or libraries</li> </ul> <p>These can be</p> <ul style="list-style-type: none"> <li>• a library created by AutoCustomization, called UBBSAMP</li> <li>• a library created manually, with a unique name</li> </ul>
<b>sampler</b>	Program that monitors a specific aspect of system performance. Includes utilization thresholds used by the Exception Monitor. The CMF MONITOR Extractor contains samplers.
<b>SBBPROF</b>	<i>See</i> profile library.
<b>scope</b>	Subset of an SSI context. The scope could be all the data for the context or a subset of data within the context. It is user- or site-defined. <i>See</i> SSI context, target.
<b>screen definition</b>	Configuration of one or more views that have been stored with the SAVEScr command and assigned a unique name. A screen includes the layout of the windows and the view, context, system, and product active in each window.
<b>selection view</b>	In MAINVIEW products, view displaying a list of available views.

---

**service class workload**

Collection of address spaces defined to OS/390 or z/OS. If you are running Workload Manager (WLM) in goal mode, MAINVIEW for OS/390 creates a service class workload for each service class that you define through WLM definition dialogs.

If you are running MVS 4.3 or earlier, or MVS/SP 5.1 or later with WLM in compatibility mode, OS/390 creates a performance group workload instead of a service class. *See* performance group workload.

**service objective**

Workload performance goal, specified in terms of response time for TSO workloads or turnaround time for batch workloads. Performance group workloads can be measured by either objective. Composite workload service objectives consist of user-defined weighting factors assigned to each constituent workload. For compatibility mode, neither OS/390 nor z/OS provides any way to measure service.

**service point**

Specification, to MAINVIEW, of the services required to enable a specific product. Services can be actions, selectors, or views. Each target (for example, CICS, DB2, or IMS) has its own service point.

The PLEX view lists all the defined service points known to the CAS to which the terminal session is connected.

**service request block (SRB)**

Control block that represents a routine to be dispatched. SRB mode routines generally perform work for the operating system at a high priority. An SRB is similar to a task control block (TCB) in that it identifies a unit of work to the system. *See also* task control block.

**service select code** Code entered to invoke analyzers, monitors, and general services. This code is also the name of the individual service.

**session**

Total period of time an address space has been active. A session begins when monitoring can be performed. If the product address space (PAS) starts after the job, the session starts with the PAS.

**SG-Auto**

*See* MAINVIEW SRM SG-Auto.

**SG-Control**

*See* MAINVIEW SRM SG-Control.

**single system image (SSI)**

Feature of the MAINVIEW window environment architecture where you can view and perform actions on multiple OS/390 or z/OS systems as though they were a single system. The rows of a single tabular view can contain rows from different OS/390 or z/OS images.

---

## **Skeleton Tailoring Facility**

A facility in MAINVIEW AutoOPERATOR that allows skeleton JCL to be used during job submission. Skeleton JCL can contain variables within the JCL statements to be substituted with data values at job submission time. Directive statements can be used in the skeleton JCL to cause the repetition of a set of skeleton statements. This facility functions similar to the TSO skeleton tailoring facility.

**SRB** *See* service request block.

**SSI** *See* single system image.

**SSI context** Name created to represent one or more targets for a given product. *See* context, target.

## **started task workload**

Address spaces running jobs that were initiated programmatically.

**statistics interval** For MAINVIEW for DB2, cumulative count within a predefined interval (30-minute default set by the DB2STATS parameter in the distributed BBPARM member BBIISP00) for an analyzer service DELTA or RATE display. Specifying the DELTA parameter displays the current value as the difference between the value sampled by the current analyzer request and the value sampled at the start of the current interval. Specifying the RATE parameter displays the current value by minute (DELTA divided by the number of elapsed minutes).

**stem variables** A REXX facility, supported in MAINVIEW AutoOPERATOR REXX EXECs and the Skeleton Tailoring Facility, where variable names end with a period followed by a number, such as &POOL.1. This configuration allows each variable to actually represent a table or array of data, with the zero variable containing the number of entries in the array. For example, &POOL.0 = 5 would indicate variables &POOL.1 through &POOL.5 exist.

**StopX37/II** *See* MAINVIEW SRM StopX37/II.

**StorageGUARD** *See* MAINVIEW SRM StorageGUARD.

**summary view** View created from a tabular view using the Summarize option in view customization. A summary view compresses several rows of data into a single row based on the summarize criteria.

**SYSPROG services** Component of MAINVIEW for OS/390. Over 100 services that detect, diagnose, and correct OS/390 or z/OS system problems as they occur. Accessible from the OS/390 Performance and Control Main Menu. Note that this component is also available as a stand-alone product MAINVIEW SYSPROG Services.

---

<b>system resource</b>	<i>See</i> object.
<b>target</b>	Entity monitored by one or more MAINVIEW products, such as an OS/390 or z/OS image, an IMS or DB2 subsystem, a CICS region, or related workloads across systems. <i>See</i> context, scope, SSI context.
<b>target context</b>	Single target/product combination. <i>See</i> context.
<b>TASCOSTR</b>	MAINVIEW for IMS Offline program that summarizes detail and summary IMS Resource Utilization Files (IRUFs) to be used as input to the offline components.
<b>task control block (TCB)</b>	Address space-specific control block that represents a unit of work that is dispatched in the address space in which it was created. <i>See also</i> service request block.
<b>TCB</b>	<i>See</i> task control block.
<b>terminal session (TS)</b>	Single point of control for MAINVIEW products, allowing data manipulation and data display and providing other terminal user services for MAINVIEW products. The terminal session runs in a user address space (either a TSO address space or a stand-alone address space for EXCP/VTAM access).
<b>TDIR</b>	<i>See</i> trace log directory.
<b>threshold</b>	Specified value used to determine whether the data in a field meets specific criteria.
<b>TLDS</b>	<i>See</i> trace log data set.
<b>total mode</b>	Usage mode in CMFMON wherein certain columns of data reflect the cumulative value between collection intervals. Invoked by the DELta OFF command. <i>See also</i> collection interval, delta mode.
<b>trace</b>	(1) Record of a series of events chronologically listed as they occur. (2) Online data collection and display services that track transaction activity through DB2, IMS, or CICS.
<b>trace log data set (TLDS)</b>	Single or multiple external VSAM data sets containing summary or detail trace data for later viewing or printing. The trace log(s) can be defined as needed or dynamically allocated by the BBI-SS PAS. Each trace request is assigned its own trace log data set(s).

---

**trace log directory (TDIR)**

VSAM linear data set containing one entry for each trace log data set. Each entry indicates the date and time of data set creation, the current status of the data set, the trace target, and other related information.

**transaction** Specific set of input data that initiates a predefined process or job.

**Transaction Accountant**

MVIMS Offline component that produces cost accounting and user charge-back records and reports.

**TS** *See* terminal session.

**TSO workload** Workload that consists of address spaces running TSO sessions.

**UAS** *See* user address space.

**UBBPARM** *See* parameter library.

**UBBPROC** *See* procedure library.

**UBBSAMP** *See* sample library.

**user address space**

Runs a MAINVIEW terminal session (TS) in TSO, VTAM, or EXCP mode.

**User BBPROF** *See* profile library.

**view** Formatted data within a MAINVIEW window, acquired from a product as a result of a view command or action. A view consists of two parts: query and form. *See also* form, job activity view, query.

**view definition** Meaning of data that appears online, including source of data, selection criteria for data field inclusion and placement, data format, summarization, context, product, view name, hyperlink fields, and threshold conditions.

**view command** Name of a view that you type on the COMMAND line to display that view.

**view command stack**

Internal stack of up to 10 queries. For each command, the stack contains the filter parameters, sort order, context, product, and time frame that accompany the view.

**view help** Online help describing the purpose of a view. To display view help, place the cursor on the view name on the window information line and press **PF1** (HELP).

---

<b>window</b>	Area of the MAINVIEW screen in which views and resources are presented. A window has visible boundaries and can be smaller than or equal in size to the MAINVIEW window area. <i>See</i> active window, alternate window, current window, MAINVIEW window area.
<b>window information line</b>	Top border of a window. Shows the window identifier, the name of the view displayed in the window, the system, the scope, the product reflected by the window, and the tomfooleries for which the data in the window is relevant. <i>See also</i> window status field.
<b>window number</b>	Sequential number assigned by MAINVIEW to each window when it is opened. The window number is the second character in the window status field. <i>See also</i> window status field.
<b>window status</b>	One-character letter in the window status field that indicates when a window is ready to receive commands, is busy processing commands, is not to be updated, or contains no data. It also indicates when an error has occurred in a window. The window status is the first character in the window status field. <i>See also</i> window information line, window status field.
<b>window status field</b>	Field on the window information line that shows the current status and assigned number of the window. <i>See also</i> window number, window status.
<b>windows mode</b>	Display of one or more MAINVIEW product views on a screen that can be divided into a maximum of 20 windows. A window information line defines the top border of each window. <i>Contrast with</i> full-screen mode.
<b>WLM workload</b>	In goal mode in MVS/SP 5.1 and later, a composite of service classes. MAINVIEW for OS/390 creates a workload for each WLM workload defined in the active service policy.
<b>workflow</b>	Measure of system activity that indicates how efficiently system resources are serving the jobs in a workload.
<b>workload</b>	(1) Systematic grouping of units of work (for example, address spaces, CICS transactions, IMS transactions) according to classification criteria established by a system administrator. (2) In OS/390 or z/OS, a group of service classes within a service definition.
<b>workload activity view</b>	Tracks workload activity as the workload accesses system resources. A workload activity view measures workload activity in terms of resource consumption and how well the workload activity meets its service objectives.
<b>Workload Analyzer</b>	Online data collection and display services used to analyze IMS workloads and determine problem causes.

---

**workload definition** Workload created through the WKLIST view. Contains a unique name, a description, an initial status, a current status, and selection criteria by which address spaces are selected for inclusion in the workload. *See* Workload Definition Facility.

**Workload Definition Facility**

In MAINVIEW for OS/390, WKLIST view and its associated dialogs through which workloads are defined and service objectives set.

**workload delay view**

Tracks workload performance as the workload accesses system resources. A workload delay view measures any delay a workload experiences as it contends for those resources.

**Workload Monitor**

Online data collection services used to monitor IMS workloads and issue warnings when defined thresholds are exceeded.

**workload objectives**

Performance goals for a workload, defined in WKLIST. Objectives can include measures of performance such as response times and batch turnaround times.

---

# Index

## Symbols

- \* parameter
  - BUDDSN command 6-3

## A

- access control
  - security exit A-4
- ACCT
  - operand of SORT command 4-15
- acct-code parameter
  - BUDGET command 6-5, 6-10, B-1
- ACTIVEDB parameter
  - RESYNC statement 5-13
- add exit A-1, A-8
  - register conventions A-8
  - uses A-8
- add processing 1-5
- ADD/CHANGE parameter
  - BUDGET command 6-10
- allocation requests
  - tracking in SG-Control 3-2
- application
  - definition 1-2
  - reviewing 1-7
  - updating 1-7
- application code
  - build exit A-1, A-6
  - BUDKEY01 A-15
  - register conventions A-7
  - uses A-6

- data sets 1-6
- definition 1-6
- determining 1-7
- lengths 5-11
- types 5-11
- application group maintenance 4-8
- applications
  - application code build exit 3-6
  - assigning to data sets 3-3
  - code construction methods 3-4
  - code parameters 3-5
  - creating 3-9
  - default code construction rules 3-4
  - modifying 3-17
  - reviewing 3-17
  - TSO BUDGET COMMAND 3-9
- AUDIT parameter
  - RESYNC statement 5-13
- auditing
  - COMPARE function 5-20

## B

- backup 3-2
  - report 7-3
- batch reporting 1-8
- batch reporting utilities 7-1
  - BUDGET TSO command 7-2
  - COBOL programs 7-2, 7-5
  - DFHSM migration and backup 7-2
  - SAS procedures 7-2, 7-4
- batch utilities
  - initializing a database 5-2

---

SGCMAINT 5-1, 5-2  
 SGCRSYNC 5-1  
 updating a database 5-2  
**BBSAMP**  
 sample exits A-14  
 sample programs 7-2  
**BBSAMP members**  
 BUDWORK A-17  
 DBI41 A-15  
 SGC1CMLK 7-5  
 SGC2CMLK 7-5  
 SGCCOB02 7-5  
 SGCCOB03 7-10  
 SGCCOBJ2 7-10  
 SGCCOBJ3 7-10  
 SGCDBCNV D-1  
 SGCHSMJ1 7-3  
 SGCMTJR1 5-16  
 SGCSAS01 7-4  
 SGCSAS02 7-5  
**BUDCHK01**  
 check exit A-15  
**BUDDEF01**  
 default exit A-14  
**BUDDSN command 6-2**  
 \* parameter 6-3  
 application codes, single data set 6-2  
 data set parameter 6-2  
 listing application codes for data sets on  
 volume 4-17, 6-3  
 listing application codes, group of data sets  
 6-3  
 listing budget applications 6-2  
 masking 6-3  
 pattern parameter 6-3  
 single data set 6-2  
 VOLSER parameter 6-4  
**budget**  
 assigning 3-8  
 definition 1-2  
**BUDGET command 6-4**  
 acct-code parameter 6-5, 6-10, B-1  
 add/change multiple application codes 6-14  
 ADD/CHANGE parameter 6-10  
 add/change single budget application code  
 6-9  
 CHANGE/ADD parameter 6-10  
 CURPERM parameter B-2  
 CURTEMP parameter B-2  
 CURVSAM parameter B-2  
 DEFAULT parameter 6-12  
 Field Table 6-8  
 field1 subparameter 6-6  
 field2 subparameter 6-8  
 FROM parameter 6-10  
 HWMTEMP parameter B-2  
 HWMVSAM parameter B-2  
 LIST parameter 6-5  
 list/change budget applications 6-4  
 listing information, multiple application  
 codes 6-6  
 listing information, single budget  
 application code 6-5  
 MAXPERM parameter 6-10  
 MAXTEMP parameter 6-10  
 MAXVSAM parameter 6-11  
 MONITOR parameter 6-12  
 NAME parameter 6-10  
 NOHSM parameter 6-11  
 NOTEMP parameter 6-10  
 NOVSAM parameter 6-11  
 operator subparameter 6-6  
 PHSM parameter 6-11  
 PTEMP parameter 6-10  
 PVSAM parameter 6-11  
 REJECT parameter 6-12  
 RESET parameter 6-11  
 special functions B-1  
 THSM parameter 6-11  
 TOTAL parameter 6-7  
 TTEMP parameter 6-11  
 TVSAM parameter 6-11  
 USERF1 parameter 6-10  
 USERF2 parameter 6-10  
 USERF3 parameter 6-10  
 value subparameter 6-6  
 WARN parameter 6-11  
 WARNING parameter 6-12  
**budget models**  
 default exit A-11  
**BUDGET TSO 3-8**  
**BUDGET TSO command**  
 JCL required 7-3  
**BUDKEY01**  
 application code build exit A-15

---

BUDSAS 7-4  
BUDSAS01  
    SAS reporting procedure 7-4  
BUDSAS02  
    SAS reporting procedure 7-5  
BUDSEC01  
    security exit A-14  
BUDSEC02  
    security exit A-14  
BUDSEC03  
    security exit A-14  
BUDSEC04  
    security exit A-14  
BUDSEL01  
    select exit A-14  
BUDSEL02  
    select exit A-14

## C

CANCEL command 4-16  
cancel panel updates 4-16  
CHANGE/ADD parameter  
    BUDGET command 6-10  
check exit A-2, A-9  
    BUDCHK01 A-15  
    register conventions A-10  
    uses A-9  
check processing 1-3  
CLEAR 5-13  
CLEAR/NOCLR parameter  
    RESYNC statement 5-13  
COBOL programs  
    batch reporting utilities 7-5  
COBOL reporting program  
    SGCCOB01 7-5  
    SGCCOB02 7-10  
    SGCCOB03 7-10  
coding rules  
    SGCMAINT control statements 5-2  
    system exits A-17  
    TSO commands 6-4  
commands  
    SG-Control  
        CANCEL 4-16  
        SORT 4-14  
COMPARE statement 5-20

    after change 5-20  
    after RESYNC 5-20  
    for auditing 5-20  
    TODB parameter 5-21  
comparing  
    database records 5-20  
conversion utility D-1  
converting  
    SG-Control to RESOLVE SRM for OS/390  
    D-1  
creating applications 3-9  
    with ISPF panels 1-6  
    with TSO commands 1-6  
CURPERM parameter  
    BUDGET command B-2  
CURTEMP parameter  
    BUDGET command B-2  
CURVSAM parameter  
    BUDGET command B-2

## D

DADSM  
    definition 1-2  
    post-processing exit 1-2  
    pre-processing exit 1-2  
    use with SG-Control 1-2  
DADSM exits A-17  
DASD  
    monitoring usage 1-2  
    reporting space availability 1-8  
    space usage 1-2  
    storage management 1-2  
DASD space administrator  
    definition 1-2  
data control  
    security exit A-5  
data set information 4-17  
data set parameter  
    BUDDSN command 6-2  
database  
    backup and recovery 3-2  
    conversion D-1  
DBNAME parameter  
    file control statement 5-4  
    REPORT statement 5-18  
default application code construction rules

---

- definition 1-7
- default exit A-2
  - BUDDEF01 A-14
  - budget models A-11
  - register conventions A-11
  - uses A-11
- DEFAULT parameter
  - BUDGET command 6-12
- defining applications 6-1
- DELETE
  - parm on TSO BUDGET Command 6-12
- DFHSM
  - migration and backup report 7-3
- DUMP parameter
  - REPORT statement 5-18

## E

- EVENTID 2-4
- examples
  - BUDDSN command 6-4
  - BUDGET command 6-7, 6-14, B-3
  - COMPARE operation 5-21
  - initializing database 3-13
  - REPORT operation 5-18
  - RESYNC function 5-14
  - setting up defaults 3-14
  - symbolic substitution 2-6
- EXCLUDE 5-13
  - VOLUME Selection statement 5-14
- exits
  - coding rules A-17
  - flow A-3
  - restrictions A-17
  - system exits A-3
  - table samples A-14

## F

- field control statement 5-7
  - FIELDn parameter 5-8
  - LEVEL parameter 5-7
  - RECALC parameter 5-7
- field layout
  - global work area A-17
- field table

- BUDGET command 6-8
  - field1 subparameter
    - BUDGET command 6-6
  - field2 subparameter
    - BUDGET command 6-8
- FIELDn parameter
  - field control statement 5-8
- file control statement 5-3
  - DBNAME parameter 5-4
  - FILE parameter 5-4
  - FROMDB parameter 5-4
  - PURGE parameter 5-4
  - VOLUME parameter 5-4
- FILE parameter
  - file control statement 5-4
- filter list and rule list parameters
  - tracking space allocation requests 3-2
- FORMAT parameter
  - REPORT statement 5-18
- FROM parameter
  - BUDGET command 6-10
- FROMDB parameter
  - COMPARE statement 5-21
  - file control statement 5-4

## G

- global work area A-17
  - description A-17
  - record layout A-17

## H

- HBACKUPPREFIX parameter
  - option control statement 5-6
- HHSM
  - operand of SORT command 4-15
- high-water marks
  - definition 1-6
- HIT operand
  - use of 2-6
- HMIGRATEPREFIX parameter
  - option control statement 5-6
- HPERM
  - operand of SORT command 4-15
- HSM

---

- operand of SORT command 4-15
- HSMTRACK parameter
  - option control statement 5-6
- HTEMP
  - operand of SORT command 4-15
- HVSAM
  - operand of SORT command 4-15
- HWMPERM parameter
  - BUDGET command B-2
- HWMTEMP parameter
  - BUDGET command B-2
- HWMVSAM parameter
  - BUDGET command B-2

## I

- IGGPOST00
  - DADSM post-processing exit 1-2
- IGGPRES00
  - DADSM pre-processing exit 1-2
- INCLUDE parameter 5-13
  - VOLUME Selection statement 5-14
- INIT statements
  - SGCMAINT control statements 5-2
- initialization
  - of the SG-Control database 5-2
  - SGCMAINT 5-2
- ISPF panels
  - application list 3-18
  - use in creating applications 1-6

## J

- JCL
  - BUDGET TSO command 7-3
  - SGCMAINT 5-2

## K

- keyword operand
  - NEWAPPL statement 2-6

## L

- LEVEL parameter

- field control statement 5-7
- LIST parameter
  - BUDGET command 6-5
- listing applications 6-1

## M

- management reporting
  - with COBOL programs 7-5
- masking 5-13
  - BUDDSN command 6-3
- MAXKEY parameter
  - option control statement 5-6
- MAXPERM parameter
  - BUDGET command 6-10
- MAXTEMP parameter
  - BUDGET command 6-10
- MAXVSAM parameter
  - BUDGET command 6-11
- migrating
  - SG-Control to RESOLVE SRM for OS/390
    - D-1
- MODE parameter
  - option control statement 5-5
- modes
  - combination 1-6
  - monitor 1-5
  - reject 1-6
  - selecting 3-7
  - warning 1-5
- MONITOR parameter
  - BUDGET command 6-12
- MPERM
  - operand of SORT command 4-15
- MTEMP
  - operand of SORT command 4-15
- multitasking RESYNC
  - SGCRSYNC 5-16
- multi-volume RESYNC 5-12
- MVSAM
  - operand of SORT command 4-15

## N

- NAME parameter
  - BUDGET command 6-10

---

NEWACCT  
  see NEWAPPL 2-4  
NEWAPPL 2-4  
  defined 2-4  
  keyword operand 2-6  
  text operand 2-6  
NOCLR/CLEAR parameter  
  RESYNC statement 5-13  
NODELETE  
  parm on TSO BUDGET command 6-12  
NOHSM parameter  
  BUDGET command 6-11  
NOTEMP parameter  
  BUDGET command 6-10  
NOVSAM parameter  
  BUDGET command 6-11

## O

online reporting 1-8  
operator subparameter  
  BUDGET command 6-6  
option control statement 5-4  
  HBACKUPPREFIX parameter 5-6  
  HMIGRATEPREFIX 5-6  
  HSMTRACK parameter 5-6  
  MAXKEY parameter 5-6  
  MODE parameter 5-5  
  RECORDS parameter 5-5  
  RECVOLS parameter 5-5  
  REJECT parameter 5-5  
overriding applications 6-1

## P

panels  
  application list 3-18  
pattern parameter  
  BUDDSN command 6-3  
performance A-5, A-17  
PERM  
  operand of SORT command 4-15  
PHSM parameter  
  BUDGET command 6-11  
processing  
  add 1-5

  check 1-3  
PTEMP parameter  
  BUDGET command 6-10  
PURGE parameter  
  file control statement 5-4  
PVSAM parameter  
  BUDGET command 6-11

## Q

query 1-7

## R

RECALC parameter  
  field control statement 5-7  
record layout  
  application record A-15  
  global work area A-17  
RECORDS parameter  
  option control statement 5-5  
recovery 3-2  
register conventions  
  add exit A-8  
  application code build exit A-7  
  check exit A-10  
  default exit A-11  
  security exit A-5  
  select exit A-13  
REJECT parameter  
  BUDGET command 6-12  
  option control statement 5-5  
REPORT statement 5-18  
  DBNAME parameter 5-18  
  DUMP parameter 5-18  
  FORMAT parameter 5-18  
reporting  
  batch 1-8  
  COBOL 7-5  
  database records 5-18  
  determining requirements 3-10  
  DFHSM migration and backup 7-3  
  diagnostic 5-1  
  online 1-8  
  trend analysis 7-1  
  trends 1-8

---

- with the TSO BUDGET command 3-10
- reserved DD statements
  - SGCDUMP C-3
  - SGCTRACE C-3
- RESET parameter
  - BUDGET command 6-11
- RESYNC
  - functionality 3-9
  - multivolume feature 5-12
  - SGCRSYNC 5-12
  - SGMAINT 5-12
  - single pack restore 5-13
  - statement 5-12
    - ACTIVEDB parameter 5-13
    - AUDIT parameter 5-13
    - CLEAR/NOCLR parameter 5-13
- resynchronizing
  - database 5-11
  - reasons for 5-11

## S

- sample exit table A-14
- SAS reporting procedure
  - batch 7-4
  - BUDSAS01 7-4
  - BUDSAS02 7-5
- SCAN 2-4
- scenarios for implementing SG-Control 3-11
- security considerations
  - TSO commands 6-1
- security exit A-2
  - access control A-4
  - BUDSEC01 A-14
  - BUDSEC02 A-14
  - BUDSEC03 A-14
  - data control A-5
  - performance A-5
  - register conventions A-5
  - uses A-4
- select exit A-2, A-12
  - BUDSEL01 A-14
  - BUDSEL02 A-14
  - register conventions A-13
- SGCCOB01
  - COBOL reporting program 7-5
- SGCCOB02
  - COBOL reporting program 7-10
- SGCCOB03
  - COBOL reporting program 7-10
- SGCDBCNV
  - SG-Control Conversion Utility D-1
- SGCDUMP DD statement
  - use of C-3
- SGCMAINT 5-1
  - COMPARE function 5-20
  - COMPARE statement 5-20
  - control statements
    - sequence of 5-2
  - field control statement 5-7
  - initialization 5-2
  - JCL required 5-2
  - option control statement 5-4
  - REPORT function 5-18
  - REPORT statement 5-18
  - RESYNC statement 5-12
  - VOLUME Selection statement 5-13
- SG-Control
  - add processing 1-5
  - check processing 1-3
  - converting
    - parameters D-1
  - description 1-2
  - functionality 1-2, 1-3
  - implementation scenarios 3-11
  - ISPF interface
    - data set information 4-17
  - upgrading to RESOLVE SRM for OS/390
    - D-1
    - use with DADSM exits 1-2, 1-3
- SG-Control database
  - initializing 5-2
  - reports 5-18
  - updating 5-2
- SGCRSYNC 5-1
  - described 5-16
  - RESYNC 5-12
- SGCTRACE DD statement
  - using C-3
- SGMAINT
  - RESYNC 5-12
- SORT command
  - ascending/descending 4-15
  - operands 4-14
  - SG-Control 4-14

---

sorting  
    allocation information 4-15  
    application codes 4-15

space usage  
    limiting 1-2  
    monitoring 1-2

space utilization  
    SG-Control monitoring and controlling 3-6

special functions  
    BUDGET command B-1

substrings  
    use in NEWAPPL 2-6

symbolic substitution  
    use of 2-6

syntax  
    BUDDSN command  
        data sets on volume 6-3  
        group of data sets 6-3  
    BUDGET command  
        add/change multiple application codes  
            B-2  
        add/change single application code 6-10,  
            B-1  
        field to field compare 6-7, B-3  
        field to value compare 6-6, B-2  
        list single application code 6-5  
        listing multiple application codes 6-6,  
            6-7  
    SGCMAINT  
        COMPARE statement 5-20  
        control statements 5-2  
        file control statement 5-4  
        option control statement 5-4  
        REPORT statement 5-18  
        RESYNC statement 5-12  
        VOLUME Selection statement 5-13

system exits  
    add exit A-8  
    check exit A-9  
    select exit A-12

## T

TEMP  
    operand of SORT command 4-15

text operand  
    NEWAPPL statement 2-6

THSM parameter  
    BUDGET command 6-11

TODB parameter  
    COMPARE statement 5-21

TOTAL parameter  
    BUDGET command 6-7

tracking space allocation requests  
    with filter and rule list parameters 3-2  
    with optional user routines 3-3

trend reporting  
    analysis report 7-1

TSO  
    BUDGET 3-8

TSO BUDGET Command  
    DELETE parm 6-12  
    NODELETE parm 6-12

TSO command  
    creating applications 1-6  
    reviewing an application 1-7  
    updating an application 1-7

TSO commands 6-1  
    Allocating SGADB 6-2  
    BUDDSN command 6-2  
    BUDGET command 6-4

TTEMP parameter  
    BUDGET command 6-11

TVSAM parameter  
    BUDGET command 6-11

## U

upgrading  
    SG-Control D-1

user fields  
    use of 1-6

USERF1 parameter  
    BUDGET command 6-10

USERF2 parameter  
    BUDGET command 6-10

USERF3 parameter  
    BUDGET command 6-10

utilities  
    conversion D-1  
    SGCMAINT 5-1  
    SGCRSYNC 5-1

---

## V

- value subparameter
  - BUDGET command 6-6
- VOLSER parameter
  - BUDDSN command 6-4
- VOLUME parameter
  - file control statement 5-4
- VOLUME selection statement 5-13
  - EXCLUDE parameter 5-14
  - INCLUDE parameter 5-14
- VSAM
  - operand of SORT command 4-15

## W

- WARN parameter
  - BUDGET command 6-11
- WARNING parameter
  - BUDGET command 6-12



# END USER LICENSE AGREEMENT NOTICE

**BY OPENING THE PACKAGE, INSTALLING, PRESSING "AGREE" OR "YES" OR USING THE PRODUCT, THE ENTITY OR INDIVIDUAL ENTERING INTO THIS AGREEMENT AGREES TO BE BOUND BY THE FOLLOWING TERMS. IF YOU DO NOT AGREE WITH ANY OF THESE TERMS, DO NOT INSTALL OR USE THE PRODUCT, PROMPTLY RETURN THE PRODUCT TO BMC OR YOUR BMC RESELLER, AND IF YOU ACQUIRED THE LICENSE WITHIN 30 DAYS OF THE DATE OF YOUR ORDER CONTACT BMC OR YOUR BMC RESELLER FOR A REFUND OF LICENSE FEES PAID. IF YOU REJECT THIS AGREEMENT, YOU WILL NOT ACQUIRE ANY LICENSE TO USE THE PRODUCT.**

This Agreement ("**Agreement**") is between the entity or individual entering into this Agreement ("You") and BMC Software Distribution, Inc., a Delaware corporation located at 2101 CityWest Blvd., Houston, Texas, 77042, USA or its affiliated local licensing entity ("BMC"). "You" includes you and your Affiliates. "Affiliate" is defined as an entity which controls, is controlled by or shares common control with a party. THIS AGREEMENT WILL APPLY TO THE PRODUCT, UNLESS (1) YOU AGREED TO A WEB BASED LICENSE AGREEMENT WITH BMC WHEN ORDERING THE PRODUCT, IN WHICH CASE THAT WEB BASED LICENSE AGREEMENT GOVERNS THE USE OF THE PRODUCT, OR (2) IF YOU DID NOT AGREE TO A WEB BASED LICENSE AGREEMENT WITH BMC WHEN ORDERING THE PRODUCT AND YOU HAVE A WRITTEN LICENSE AGREEMENT WITH BMC, THEN THAT WRITTEN AGREEMENT GOVERNS THE USE OF THE PRODUCT. THE ELECTRONIC AGREEMENT PROVIDED WITH THE PRODUCT AS PART OF THE INSTALLATION OF THE PRODUCT WILL NOT APPLY. In addition to the restrictions imposed under this Agreement, any other usage restrictions contained in the Product installation instructions or release notes shall apply to Your use of the Product.

**PRODUCT AND CAPACITY.** "**Software**" means the object code version of the computer programs provided, via delivery or electronic transmission, to You. Software includes computer files, enhancements, maintenance modifications, upgrades, updates, bug fixes, and error corrections.

"**Documentation**" means all written or graphical material provided by BMC in any medium, including any technical specifications, relating to the functionality or operation of the Software.

"**Product**" means the Software and Documentation.

"**License Capacity**" means the licensed capacity for the Software with the pricing and other license defining terms, including capacity restrictions, such as tier limit, total allowed users, gigabyte limit, quantity of Software, and/or other capacity limitations regarding the Software. For licenses based on the power of a computer, You agree to use BMC's current computer classification scheme, which is available at <http://www.bmc.com> or can be provided to You upon request.

**ACCEPTANCE.** The Product is deemed accepted by You, on the date that You received the Product from BMC.

**LICENSE.** Subject to the terms of this Agreement, as well as Your payment of applicable fees, BMC grants You a non-exclusive, non-transferable, perpetual (unless a term license is provided on an order) license for each copy of the Software, up to the License Capacity, to do the following:

- (a) install the Software on Your owned or leased hardware located at a facility owned or controlled by You in the country where You acquired the license;
- (b) operate the Software solely for processing Your own data in Your business operations; and
- (c) make one copy of the Software for backup and archival purposes only (collectively a "**License**").

If the Software is designed by BMC to permit you to modify such Software, then you agree to only use such modifications or new software programs for Your internal purposes or otherwise consistent with the License. BMC grants You a license to use the Documentation solely for Your internal use in Your operations.

**LICENSE UPGRADES.** You may expand the scope of the License Capacity only pursuant to a separate agreement with BMC for such expanded usage and Your payment of applicable fees. There is no additional warranty period or free support period for license upgrades.

**RESTRICTIONS:** You agree to **NOT:**

- (a) disassemble, reverse engineer, decompile or otherwise attempt to derive any Software from executable code;
- (b) distribute or provide the Software to any third party (including without limitation, use in a service bureau, outsourcing environment, or processing the data of third parties, or for rental, lease, or sublicense); or
- (c) provide a third party with the results of any functional evaluation or benchmarking or performance tests, without BMC's prior written approval, unless prohibited by local law.

**TRIAL LICENSE.** If, as part of the ordering process, the Product is provided on a trial basis, then these terms apply: (i) this license consists solely of a non-exclusive, non-transferable evaluation license to operate the Software for the period of time specified from BMC or, if not specified, a 30 day time period ("**Trial Period**") only for evaluating whether You desire to acquire a capacity-based license to the Product for a fee; and (ii) Your use of the Product is on an AS IS basis without any warranty, and **BMC, ITS AFFILIATES AND RESELLERS, AND LICENSORS DISCLAIM ANY AND ALL WARRANTIES (INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT) AND HAVE NO LIABILITY WHATSOEVER RESULTING FROM THE USE OF THIS PRODUCT UNDER THIS TRIAL LICENSE ("Trial License").** BMC may terminate for its convenience a Trial License upon notice to You. When the Trial Period ends, Your right to use this Product automatically expires. If You want to continue Your use of the Product beyond the Trial Period, contact BMC to acquire a capacity-based license to the Product for a fee.

**TERMINATION.** This Agreement shall immediately terminate if You breach any of its terms. Upon termination, for any reason, You must uninstall the Software, and either certify the destruction of the Product or return it to BMC.

**OWNERSHIP OF THE PRODUCT.** BMC or its Affiliates or licensors retain all right, title and interest to and in the BMC Product and all intellectual property, informational, industrial property and proprietary rights therein. BMC neither grants nor otherwise transfers any rights of ownership in the BMC Product to You. BMC Products are protected by applicable copyright, trade secret, and industrial and intellectual property laws. BMC reserves any rights not expressly granted to You herein.

**CONFIDENTIAL AND PROPRIETARY INFORMATION.** The BMC Products are and contain valuable confidential information of BMC ("**Confidential Information**"). Confidential Information means non-public technical and non-technical information relating to the BMC Products and Support, including, without limitation, trade secret and proprietary information, and the structure and organization of the Software. You may not disclose the Confidential Information to third parties. You agree to use all reasonable efforts to prevent the unauthorized use, copying, publication or dissemination of the Product.

**WARRANTY.** Except for a Trial License, BMC warrants that the Software will perform in substantial accordance with the Documentation for a period of one year from the date of the order. This warranty shall not apply to any problems caused by software or hardware not supplied by BMC or to any misuse of the Software.

**EXCLUSIVE REMEDY.** BMC's entire liability, and Your exclusive remedy, for any defect in the Software during the warranty period or breach of the warranty above shall be limited to the following: BMC shall use reasonable efforts to remedy defects covered by the warranty or replace the defective Software within a reasonable period of time, or if BMC cannot remedy or replace such defective copy of the Software, then BMC shall refund the amount paid by You for the License for that Software. BMC's obligations in this section are conditioned upon Your providing BMC prompt access to the affected Software and full cooperation in resolving the claim.

**DISCLAIMER. EXCEPT FOR THE EXPRESS WARRANTIES ABOVE, THE PRODUCT IS PROVIDED "AS IS." BMC, ITS AFFILIATES AND LICENSORS SPECIFICALLY DISCLAIM ALL OTHER WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. BMC DOES NOT WARRANT THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR FREE, OR THAT ALL DEFECTS CAN BE CORRECTED.**

**DISCLAIMER OF DAMAGES. IN NO EVENT IS BMC, ITS AFFILIATES OR LICENSORS LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES RELATING TO OR ARISING OUT OF THIS AGREEMENT, SUPPORT, AND/OR THE PRODUCT (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, LOST COMPUTER USAGE TIME, AND DAMAGE OR LOSS OF USE OF DATA), EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND IRRESPECTIVE OF ANY NEGLIGENCE OF BMC OR WHETHER SUCH DAMAGES RESULT FROM A CLAIM ARISING UNDER TORT OR CONTRACT LAW.**

**LIMITS ON LIABILITY. BMC'S AGGREGATE LIABILITY FOR DAMAGES IS LIMITED TO THE AMOUNT PAID BY YOU FOR THE LICENSE TO THE PRODUCT.**

**SUPPORT.** If Your order includes support for the Software, then BMC agrees to provide support (24 hours a day/7 days a week) ("**Support**"). You will be automatically re-enrolled in Support on an annual basis unless BMC receives notice of termination from You as provided below. There is a free support period during the one year warranty period.

(a) **Support Terms.** BMC agrees to make commercially reasonable efforts to provide the following Support: (i) For malfunctions of supported versions of the Software, BMC provides bug fixes, patches or workarounds in order to cause that copy of the Software to operate in substantial conformity with its then-current operating specifications; and (ii) BMC provides new releases or versions, so long as such new releases or versions are furnished by BMC to all other enrolled Support customers without additional charge. BMC may refuse to provide Support for any versions or releases of the Software other than the most recent version or release of such Software made available by BMC. Either party may terminate Your enrollment in Support upon providing notice to the other at least 30 days prior to the next applicable Support anniversary date. If You re-enroll in Support, BMC may charge You a reinstatement fee of 1.5 times what You would have paid if You were enrolled in Support during that time period.

(b) **Fees.** The annual fee for Support is 20% of the Software's list price less the applicable discount or a flat capacity based annual fee. BMC may change its prices for the Software and/or Support upon at least 30 days notice prior to Your support anniversary date.

**VERIFICATION.** If requested by BMC, You agree to deliver to BMC periodic written reports, whether generated manually or electronically, detailing Your use of the Software in accordance with this Agreement, including, without limitation, the License Capacity. BMC may, at its expense, audit Your use of the Software to confirm Your compliance with the Agreement. If an audit reveals that You have underpaid fees, You agree to pay such underpaid fees. If the underpaid fees exceed 5% of the fees paid, then You agree to also pay BMC's reasonable costs of conducting the audit.

**EXPORT CONTROLS.** You agree not to import, export, re-export, or transfer, directly or indirectly, any part of the Product or any underlying information or technology except in full compliance with all United States, foreign and other applicable laws and regulations.

**GOVERNING LAW.** This Agreement is governed by the substantive laws in force, without regard to conflict of laws principles: (a) in the State of New York, if you acquired the License in the United States, Puerto Rico, or any country in Central or South America; (b) in the Province of Ontario, if you acquired the License in Canada (subsections (a) and (b) collectively referred to as the "**Americas Region**"); (c) in Singapore, if you acquired the License in Japan, South Korea, Peoples Republic of China, Special Administrative Region of Hong Kong, Republic of China, Philippines, Indonesia, Malaysia, Singapore, India, Australia, New Zealand, or Thailand (collectively, "**Asia Pacific Region**"); or (d) in the Netherlands, if you acquired the License in any other country not described above. The United Nations Convention on Contracts for the International Sale of Goods is specifically disclaimed in its entirety.

**ARBITRATION. ANY DISPUTE BETWEEN YOU AND BMC ARISING OUT OF THIS AGREEMENT OR THE BREACH OR ALLEGED BREACH, SHALL BE DETERMINED BY BINDING ARBITRATION CONDUCTED IN ENGLISH. IF THE DISPUTE IS INITIATED IN THE AMERICAS REGION, THE ARBITRATION SHALL BE HELD IN NEW YORK, U.S.A., UNDER THE CURRENT COMMERCIAL OR INTERNATIONAL, AS APPLICABLE, RULES OF THE AMERICAN ARBITRATION ASSOCIATION. IF THE DISPUTE IS INITIATED IN A COUNTRY IN THE ASIA PACIFIC REGION, THE ARBITRATION SHALL BE HELD IN SINGAPORE, SINGAPORE UNDER THE CURRENT UNCITRAL ARBITRATION RULES. IF THE DISPUTE IS INITIATED IN A COUNTRY OUTSIDE OF THE AMERICAS REGION OR ASIA PACIFIC REGION, THE ARBITRATION SHALL BE HELD IN AMSTERDAM, NETHERLANDS UNDER THE CURRENT UNCITRAL ARBITRATION RULES. THE COSTS OF THE ARBITRATION SHALL BE BORNE EQUALLY PENDING THE ARBITRATOR'S AWARD. THE AWARD RENDERED SHALL BE FINAL AND BINDING UPON THE PARTIES AND SHALL NOT BE SUBJECT TO APPEAL TO ANY COURT, AND MAY BE ENFORCED IN ANY COURT OF COMPETENT JURISDICTION. NOTHING IN THIS AGREEMENT SHALL BE DEEMED AS PREVENTING EITHER PARTY FROM SEEKING INJUNCTIVE RELIEF FROM ANY COURT HAVING JURISDICTION OVER THE PARTIES AND THE SUBJECT MATTER OF THE DISPUTE AS NECESSARY TO PROTECT EITHER PARTY'S CONFIDENTIAL INFORMATION, OWNERSHIP, OR ANY OTHER**

**PROPRIETARY RIGHTS. ALL ARBITRATION PROCEEDINGS SHALL BE CONDUCTED IN CONFIDENCE, AND THE PARTY PREVAILING IN ARBITRATION SHALL BE ENTITLED TO RECOVER ITS REASONABLE ATTORNEYS' FEES AND NECESSARY COSTS INCURRED RELATED THERETO FROM THE OTHER PARTY.**

**U.S. GOVERNMENT RESTRICTED RIGHTS.** The Software under this Agreement is "commercial computer software" as that term is described in 48 C.F.R. 252.227-7014(a)(1). If acquired by or on behalf of a civilian agency, the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of this Agreement as specified in 48 C.F.R. 12.212 (Computer Software) and 12.211 (Technical Data) of the Federal Acquisition Regulations ("**FAR**") and its successors. If acquired by or on behalf of any agency within the Department of Defense ("**DOD**"), the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of this Agreement as specified in 48 C.F.R. 227.7202 of the DOD FAR Supplement and its successors.

**MISCELLANEOUS TERMS.** You agree to pay BMC all amounts owed no later than 30 days from the date of the applicable invoice, unless otherwise provided on the order for the License to the Products. You will pay, or reimburse BMC, for taxes of any kind, including sales, use, duty, tariffs, customs, withholding, property, value-added (VAT), and other similar federal, state or local taxes (other than taxes based on BMC's net income) imposed in connection with the Product and/or the Support. This Agreement constitutes the entire agreement between You and BMC and supersedes any prior or contemporaneous negotiations or agreements, whether oral, written or displayed electronically, concerning the Product and related subject matter. No modification or waiver of any provision hereof will be effective unless made in a writing signed by both BMC and You. You may not assign or transfer this Agreement or a License to a third party without BMC's prior written consent. Should any provision of this Agreement be invalid or unenforceable, the remainder of the provisions will remain in effect. The parties have agreed that this Agreement and the documents related thereto be drawn up in the English language. Les parties exigent que la présente convention ainsi que les documents qui s'y rattachent soient rédigés en anglais.

SW EULA Int 030102



# Notes



\*ItemNumber\*