

# **MAINVIEW<sup>®</sup> SRM Allocation StopX37/II User Guide and Reference**

**Version 7.2**

**June 20, 2003**



Copyright 2003 BMC Software, Inc., as an unpublished work.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 UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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>xi</b>
<b>Chapter 1</b>	<b>What Is StopX37/II?</b>
<b>Chapter 2</b>	<b>How to Use StopX37/II</b>
	Overview..... 2-2
	Functions Summary ..... 2-3
	Global Parameter Cross-Reference..... 2-4
	StopX37/II Processing ..... 2-5
	VSAM Considerations ..... 2-7
	SMS Considerations..... 2-14
	Data Striping Considerations ..... 2-15
	Filter and Rule List Sample ..... 2-17
	Processing Restrictions and Recommendations..... 2-19
	Tip for StopX37/II Stand-Alone Users ..... 2-23
<b>Chapter 3</b>	<b>Functions Reference</b>
	Overview..... 3-2
	NOCATLG2 - Prevent NOT CATLG 2 Errors ..... 3-3
	NOCATLG2 Considerations ..... 3-4
	Bypassing NO CATLG2 Processing ..... 3-5
	Considerations for NOCATLG2 for SMS-Managed Data Sets ..... 3-6
	Usage Notes ..... 3-9
	OPENEMPT - Initialize Empty Data Sets ..... 3-12
	Usage Notes ..... 3-13
	OPTBLKSZ - Set Optimum Blocksize..... 3-16
	Usage Notes ..... 3-18
	SPACCONV - Change Allocation Units to Blocks ..... 3-21
	Usage Notes ..... 3-23
	SPACPRIM - Reduce Primary Space Allocation ..... 3-25
	Usage Notes ..... 3-28
	SPACRLSE - Release Data Set Space at Close ..... 3-32
	Usage Notes ..... 3-33
	SPACSECA - Add Secondary Allocation Quantity ..... 3-35

---

Usage Notes .....	3-36
SPACSECB - Reduce Secondary Allocation to Best Fit. ....	3-39
Usage Notes .....	3-40
SPACSECI - Increase Secondary Allocation Quantity .....	3-42
Usage Notes .....	3-43
SPACSECR - Reduce Secondary Allocation to Largest Extent .....	3-47
Usage Notes .....	3-49
SPACSQTY - Set Primary and Secondary Space for Data Set .....	3-52
Usage Notes .....	3-56
SPACSWIR - Reduce Initial Allocation on Volume Addition .....	3-60
Usage Notes .....	3-62
SPACVOLA - Add Volume During Allocation. ....	3-64
Usage Notes .....	3-71
VIOALLOC - Control VIO Allocation .....	3-78
Usage Notes .....	3-79

## Appendix A

### Utilities

Overview .....	A-2
SMF Analysis Program (SMFX37A) .....	A-2
StopX37/II Analysis Program (SMFX37B) .....	A-7
DASD Utilization Report (X37UTILC) .....	A-19
Dynamic Multivolume Data Set Extent Consolidation (X37REORG) ..	A-25
Data Set Reorganization (REDUCEXT) .....	A-30

## Index

---

# List of Figures

Figure 2-1	Example Coding to Bypass Critical Data Sets . . . . .	2-11
Figure A-1	SMFX37A Sample Report . . . . .	A-6
Figure A-2	Abend Analysis Report Sample . . . . .	A-6
Figure A-3	Writing SMF records to a Generation Data Set . . . . .	A-7
Figure A-4	Sample JCL - SMFX37B . . . . .	A-14
Figure A-5	SMFX37B Detail Report . . . . .	A-15
Figure A-6	SMFX37B Detail Report . . . . .	A-15
Figure A-7	SMFX37B Detail Report . . . . .	A-16
Figure A-8	SMFX37B Detail Report . . . . .	A-16
Figure A-9	SMFX37B Detail Report . . . . .	A-17
Figure A-10	SMFX37B Detail Report . . . . .	A-17
Figure A-11	SMFX37B Detail Report . . . . .	A-18
Figure A-12	Sample JCL - X37UTILC . . . . .	A-22
Figure A-13	Sample DASD Utilization Report . . . . .	A-23
Figure A-14	Sample DASD Utilization Report . . . . .	A-24
Figure A-15	Sample DASD Utilization Report Summary . . . . .	A-24
Figure A-16	Sample JCL for X37REORG . . . . .	A-27
Figure A-17	DFDSS Control Card Input Sample . . . . .	A-29
Figure A-18	Sample JCL to Run REDUCEXT . . . . .	A-32



---

# List of Tables

Table 2-1	Application Groupings of StopX37/II Functions . . . . .	2-2
Table 2-2	Global Parameters Accessed by StopX37/II Functions and Utilities . . . . .	2-4
Table 2-3	Filter and Rule List Sample . . . . .	2-18
Table 2-4	SPACVOLA Restrictions and Recommendations . . . . .	2-20
Table 2-5	NOCATLG2 Restrictions and Recommendations . . . . .	2-20
Table 2-6	OPENEMPT Restrictions and Recommendations . . . . .	2-21
Table 2-7	OPTBLKSZ Restrictions and Recommendations . . . . .	2-21
Table 2-8	SPACCONV Restrictions and Recommendations . . . . .	2-21
Table 2-9	SPACPRIM Restrictions and Recommendations . . . . .	2-21
Table 2-10	SPACRLSE Restrictions and Recommendations . . . . .	2-21
Table 2-11	SPACSECA Restrictions and Recommendations . . . . .	2-22
Table 2-12	SPACSECB Restrictions and Recommendations . . . . .	2-22
Table 2-13	SPACSECI Restrictions and Recommendations . . . . .	2-22
Table 2-14	SPACSECR Restrictions and Recommendations . . . . .	2-22
Table 2-15	SPACSQTY Restrictions and Recommendations . . . . .	2-22
Table 2-16	VIOALLOC Restrictions and Recommendations . . . . .	2-22
Table 3-1	Global Parameters—NOCATLG2 . . . . .	3-6
Table 3-2	Global Parameters—OPTBLKSZ . . . . .	3-16



---

---

# About This Book

This book contains detailed information about MAINVIEW® Storage Resource Manager StopX37/II 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)

Throughout this book, references to OS/390 support also include support for MVS and z/OS.

## How This Book Is Organized

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

Chapter/Appendix	Description
Chapter 1, "What Is StopX37/II?"	provides a brief overview of StopX37/II and a summary of StopX37/II functions
Chapter 2, "How to Use StopX37/II"	provides user information to help you tailor StopX37/II to your needs
Chapter 3, "Functions Reference"	provides detailed explanations and examples of each StopX37/II function
Appendix A, "Utilities"	provides explanations and examples batch reports provided by StopX37/II

---

## 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. These publications are available on request from BMC Software.

Category	Document	Description
general	<i>MAINVIEW Products General Information</i>	provides an overview of the MAINVIEW environment and the products that it supports
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 StopX37/II
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 a reference of global parameters, filter list 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
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.
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.	<i>dtsbackup control_directory</i>
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.	<i>[table_name, column_name, field]</i>
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.	<i>{DBD_name   table_name}</i>
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> .	<i>{commit   cancel}</i>
An ellipsis indicates that you can repeat the previous item or items as many times as necessary.	<i>column_name . . .</i>



---

---

# Chapter 1 What Is StopX37/II?

MAINVIEW SRM StopX37/II is one of the products that make up the MAINVIEW SRM Allocation storage management product group, which also includes MAINVIEW SRM EasySMS and MAINVIEW SRM EasyPOOL.

The Allocation component provides

- enhancements to OS/390 space management, reducing the incidence of space-related processing problems
- control over data set allocation and enforcement of allocation and naming standards
- tools that aid in the conversion to DFSMS and provide enhancements to the DFSMS environment after implementation

Allocation functions operate at the operating system level to interceptabend conditions or standards violations, thus providing services without any JCL changes.

Some of the features and uses of StopX37/II are:

- Addition of volumes dynamically to prevent out-of-space conditions
- Prevention of errors caused by duplicate catalog entries
- Controlled reduction of primary space to prevent allocation failure
- Dynamic adjustment of primary and secondary extent sizes based on available space
- Release of unused data set space
- Secondary extent allocation services:

- 
- Addition of secondary allocation quantity if none specified
  - Reduction of secondary allocation to largest available extent
  - Increase allocation sizes for secondary extents after a specified extent count
- Allocation of space in blocks versus tracks/cylinders
  - Optimization of block sizes
  - Prevention of errors caused by uninitialized data sets
  - Redirect temporary data sets to VIO

---

---

## Chapter 2 How to Use StopX37/II

This chapter provides information to help you tailor StopX37/II to your needs.

Overview . . . . .	2-2
Functions Summary . . . . .	2-3
Global Parameter Cross-Reference . . . . .	2-4
StopX37/II Processing . . . . .	2-5
VSAM Considerations . . . . .	2-7
SMS Considerations . . . . .	2-14
Data Striping Considerations . . . . .	2-15
Filter and Rule List Sample . . . . .	2-17
Processing Restrictions and Recommendations . . . . .	2-19
Tip for StopX37/II Stand-Alone Users . . . . .	2-23

# Overview

MAINVIEW SRM StopX37/II consists of a set of functions that help you

- prevent space abends
- enhance data set allocation efficiency
- prevent NO CATLG2 or JCL errors
- add volumes during allocation
- add secondary allocation quantity
- reduce or set primary space allocation
- reduce or set secondary space allocation
- reduce initial allocation on volume add
- prevent errors caused by uninitialized data sets

Using StopX37/II involves activating and controlling StopX37/II functions. Refer to the *MAINVIEW SRM User Guide and Reference* for details.

Table 2-1 lists the functions grouped according to general applications. An alpha listing of StopX37/II functions is shown on “Functions Summary” on page 2-3 for easy reference. Table 2-2 on page 2-4 is a matrix of StopX37/II functions and the global parameters that support them.

**Table 2-1 Application Groupings of StopX37/II Functions (Part 1 of 2)**

<b>If you need to...</b>	<b>Consider using...</b>
Prevent space abends	“SPACPRIM - Reduce Primary Space Allocation” on page 3-25 “SPACSECA - Add Secondary Allocation Quantity” on page 3-35 “SPACSECB - Reduce Secondary Allocation to Best Fit” on page 3-39 “SPACSECI - Increase Secondary Allocation Quantity” on page 3-42 “SPACSECR - Reduce Secondary Allocation to Largest Extent” on page 3-47 “SPACSWIR - Reduce Initial Allocation on Volume Addition” on page 3-60 “SPACVOLA - Add Volume During Allocation” on page 3-64
Enhance data set allocation efficiency	“OPENEMPT - Initialize Empty Data Sets” on page 3-12 “OPTBLKSZ - Set Optimum Blocksize” on page 3-16 “SPACCONV - Change Allocation Units to Blocks” on page 3-21 “SPACRLSE - Release Data Set Space at Close” on page 3-32 “SPACSQTY - Set Primary and Secondary Space for Data Set” on page 3-52 “VIOALLOC - Control VIO Allocation” on page 3-78
Prevent NO CATLG2 or JCL errors	“NOCATLG2 - Prevent NOT CATLG 2 Errors” on page 3-3
Add volumes during allocation	“SPACVOLA - Add Volume During Allocation” on page 3-64
Add secondary allocation quantity	“SPACSECA - Add Secondary Allocation Quantity” on page 3-35

**Table 2-1 Application Groupings of StopX37/II Functions (Part 2 of 2)**

If you need to...	Consider using...
Reduce or set primary space allocation	"SPACPRIM - Reduce Primary Space Allocation" on page 3-25 "SPACSQTY - Set Primary and Secondary Space for Data Set" on page 3-52
Reduce or set secondary space allocation	"SPACSECA - Add Secondary Allocation Quantity" on page 3-35 "SPACSECB - Reduce Secondary Allocation to Best Fit" on page 3-39 "SPACSECI - Increase Secondary Allocation Quantity" on page 3-42 "SPACSECR - Reduce Secondary Allocation to Largest Extent" on page 3-47 "SPACSQTY - Set Primary and Secondary Space for Data Set" on page 3-52
Reduce initial allocation on volume add	"SPACSWIR - Reduce Initial Allocation on Volume Addition" on page 3-60
Prevent errors caused by uninitialized data sets	"OPENEMPT - Initialize Empty Data Sets" on page 3-12

## Functions Summary

Use the following list find the StopX37/II function you are looking for.

- NOCATLG2 - Prevent NOT CATLG 2 Errors . . . . . 3-3
- OPENEMPT - Initialize Empty Data Sets . . . . . 3-12
- OPTBLKSZ - Set Optimum Blocksize. . . . . 3-16
- SPACCONV - Change Allocation Units to Blocks . . . . . 3-21
- SPACPRIM - Reduce Primary Space Allocation . . . . . 3-25
- SPACRLSE - Release Data Set Space at Close . . . . . 3-32
- SPACSECA - Add Secondary Allocation Quantity . . . . . 3-35
- SPACSECB - Reduce Secondary Allocation to Best Fit . . . . . 3-39
- SPACSECI - Increase Secondary Allocation Quantity. . . . . 3-42
- SPACSECR - Reduce Secondary Allocation to Largest Extent . . . . . 3-47
- SPACSQTY - Set Primary and Secondary Space for Data Set. . . . . 3-52
- SPACSWIR - Reduce Initial Allocation on Volume Addition . . . . . 3-60
- SPACVOLA - Add Volume During Allocation . . . . . 3-64
- VIOALLOC - Control VIO Allocation. . . . . 3-78

StopX37/II functions do not operate for DFDSS jobs.

**Tip:** BMC Software has discovered that some SAS programs (RECFM=FS) interfere with or bypass standard storage utilities, including some StopX37/II functions. To avoid allocation issues or unpredictable SAS results, customers with SAS installed should add the following statements to the FLST for the SPACSECR and SPACVOLA functions:

```
SET MODE=INACTIVE
PGM=SAS/ RECFM=FS DSORG=PS
```

## Global Parameter Cross-Reference

Table 2-2 provides a matrix of StopX37/II functions and utilities and the global parameters that support them. Global parameters are defined in the *MAINVIEW SRM Reference Summary* and in an appendix in the *MAINVIEW SRM User Guide and Reference*.

**Table 2-2 Global Parameters Accessed by StopX37/II Functions and Utilities (Part 1 of 2)**

Global Parameter	NOCATLG2	OPENEMPT	OPTBLKSZ	SPACCONV	SPACPRIM	SPACRLSE	SPACSECA	SPACSECB	SPACSECI	SPACSECR	SPACSQTY	SPACSWIR	SPACVOLA	VIOALLOC	SMFX37B
DADSMEX													X		
DCTYPE												X	X		
DFREORGPRC													X		
MAXVOL													X		
MREDUCE										X					
MSGPREF	X													X	
NOCATPFX	X														
NOCATPRG	X														
NOCATSEC	X														
NOCATSMS	X														
NOCATVOL	X														
NOCATWHEN	X														
REQTYPE													X		
SCAT													X		
SIZEISPRIM	X													X	
SKIP												X	X		

Table 2-2 Global Parameters Accessed by StopX37/II Functions and Utilities (Part 2 of 2)

Global Parameter	NOCATLG2	OPENEMPT	OPTBLKSZ	SPACCONV	SPACPRIM	SPACRLSE	SPACSECA	SPACSECB	SPACSECI	SPACSECR	SPACSQTY	SPACSWIR	SPACVOLA	VIOALLOC	SMFX37B
SMFID															X
VSAM_ADJ4GB							X	X		X		X	X		
VSAMLIMWARN										X		X	X		
VSAMPRIM												X	X		
VSAMZSEC							X						X		
X37POOL											X		X	X	
X37RLS							X	X		X		X	X		

## StopX37/II Processing

StopX37/II is a systems software product that attempts recovery on jobs that are abending with B37, D37, or E37 return codes and many other out-of-space conditions. These abend codes are associated with disk space availability and management. These codes are commonly found, even in well run data processing centers. Automatic recovery from these abend conditions through the use of StopX37/II can remove a significant burden from your installation's support staff.

Once installed, StopX37/II interacts with a module within the End-of-Volume SVC, a part of the OS/390 operating system. Before the operating system invokes the module that produces these return codes, it encounters StopX37/II. StopX37/II employs user-defined filtering language to decide whether to attempt recovery before the job abends.

Sequential data sets can be multivolume. If a job uses all sixteen extents available to a sequential data set, StopX37/II can dynamically allocate and add another volume. The sequential data set then becomes a multivolume data set. In order to make a data set multivolume, the data set must already be cataloged. StopX37/II adds the new volumes to the data set's catalog entry.

Your installation may not want StopX37/II to attempt recovery for all B37, D37, E37, or other errors. The decision to attempt recovery for a particular job can be based on any of many criteria, including job name, job class, account number, data set name, RACF group, and so on.

*Filtering criteria* are specified using a full screen ISPF interface. The filtering criteria are used to select which jobs will be recovered. Once a job has been selected, a *rule list* is applied to the job to determine the specific parameters of the recovery action to be taken. The rule list is also specified using a full screen ISPF interface. The filter and rule lists together determine which jobs will be recovered and the recovery actions that will be taken.

A complete audit trail is provided by way of the JOBLOG/SYSLOG and SMF data sets. Batch reporting utilities are provided to scan SMF data for statistics about recovery attempts and actions taken.

StopX37/II functions offer several methods for tailoring the product to installation requirements. StopX37/II will not *take over* your system and attempt to recover all abend conditions. The product only attempts recovery for user-directed abend conditions.

Therefore, once StopX37/II has been installed and tested, you need to decide which types of jobs merit a recovery attempt. The decision to attempt recovery for a particular job can be based on any of many criteria, including job name, job class, account number, data set name, or RACF group.

This chapter describes the steps necessary to define to StopX37/II your installation's recovery criteria and the associated action to take when those criteria are met.

## VSAM Considerations

This section defines VSAM considerations for StopX37/II processing. The following topics are discussed:

VSAM Primary Space Reduction Facility . . . . .	2-8
Space Reduction Rules for Key-Sequenced Clusters . . . . .	2-9
Nonrecovery Conditions . . . . .	2-10
Bypassing Critical Data Sets . . . . .	2-11
Bypassing VSAM EOV Processing . . . . .	2-11
StopX37/II for VSAM Record-Level-Sharing (RLS) Data Sets . . . . .	2-12

StopX37/II intercepts *out-of-space* conditions and allocates additional volumes to VSAM files when secondary space is insufficient. VSAM data sets can use up to 123 extents on a volume. With such growth potential, it seems VSAM would have relatively few out-of-space conditions. Unfortunately, the VSAM design creates situations that fragment volumes and wastes space in VSAM pools. These out-of-space conditions are an unfortunate reality in most installations.

When users are allocating VSAM files, the DEFINE CLUSTER control statements are frequently *borrowed* from other jobs or users. In this environment, files are not evenly distributed within the VSAM volume pool. The StopX37/II option recovers the problems caused by the fragmentation or over allocation of a volume. The jobs that would have failed with out-of-space conditions run to completion.

StopX37/II intercepts several types of out-of-space conditions for VSAM data sets. If there is insufficient space available to satisfy the primary space request, StopX37/II can reduce the request for primary space. This option is controlled by the SPACPRIM parameter and works in the same way for VSAM data sets as it does for non-VSAM data sets - that is, the primary space request is reduced by the percentage specified in the SPACPRIM parameter until the data set is allocated or until the specified minimum is reached. Whenever a secondary extent must be added to a VSAM data set and there is insufficient space on the volume (or in the SMS storage group) for the new extent, StopX37/II can recover from the error in one of several ways: if no secondary space amount was specified, StopX37/II can provide a secondary space amount, or if a secondary space amount was specified, the

secondary space request can be reduced, or another volume can be added to the data set. If another volume is added to the data set, the amount of space requested will normally be the primary space amount. Optionally, StopX37/II can request the secondary space amount instead of the primary space amount when a VSAM data set is extended to a new volume. The secondary space amount can be obtained whether the new volume was added by StopX37/II or whether the volume was specified as a candidate volume when the data set was defined. A catalog entry has room for 123 extents. While non-VSAM data sets are limited to sixteen extents per volume for 59 volumes, VSAM files are limited to 123 extents - no matter how many volumes are used.

**Note:** If you would like to manage the distribution of VSAM files, call your BMC Software representative and ask about EasyPOOL.

### VSAM Primary Space Reduction Facility

StopX37/II includes a facility to prevent errors due to insufficient primary space availability when defining ICF VSAM data sets. These errors can occur due to volume fragmentation, excessive demands for space, or any other condition which prevents a step from obtaining the space requested when attempting to define an ICF VSAM cluster.

Insufficient-primary-space errors which occur when VSAM data sets are defined result in message IDC3009I, return code 68, reason code 20. This message indicates that the DADSM request to allocate space on a DASD volume failed because sufficient space to meet the request was not available in five extents or less. StopX37/II can intercept this error and reduce the primary space request so that the job can continue.

StopX37/II uses the SPACPRIM parameter specified in the rule list to determine how much to reduce the VSAM primary space request and to establish a threshold below which the primary space request will not be reduced.

For example, assume the following StopX37/II rule list statement has been specified:

```
SET SPACPRIM=( 50 , 10 )
```

In this case, the original amount of primary space requested is reduced 10 percent at a time, until the cluster is successfully defined, or until the amount of space requested is less than 50 percent of the original request (in which case the DEFINE fails).

## Space Reduction Rules for Key-Sequenced Clusters

For key-sequenced ICF VSAM clusters (KSDS), primary space reduction is performed according to the following rules. These rules apply to both base clusters and alternate index clusters:

- If space is specified as a parameter of CLUSTER or DATA, reduction is performed on the data component.
- If space is specified as a parameter of both DATA and INDEX, and the type of space requested is the same on both parameters (for example, both DATA and INDEX requests are in TRACKS), reduction is performed on the component which specifies the largest request for space.
- If space is specified as a parameter of both DATA and INDEX, and the type of space requested on both parameters is *different* (for example, the DATA request is in RECORDS and INDEX request is in TRACKS), reduction is performed on the data component.

For non-KSDS clusters (for example, RRDS, ESDS, and so on), primary space reduction is always performed on the data component, since there is no index component to be reduced. Keyrange clusters are not eligible for primary space reduction.

There are two (relatively rare) cases in which StopX37/II Primary Space Reduction may be unable to successfully prevent a primary space not-available error:

- The primary space request is only reduced for one component of a cluster, never for both components. Thus, primary space reduction may fail to recover the error if both the data and the index components must be reduced in order to successfully define the data set.
- Occasionally, primary space reduction cannot be performed because doing so would alter the attributes of the cluster (for example, the control area size) so that the cluster cannot be defined.

Both of these errors are unlikely to occur except when defining a key-sequenced ICF VSAM cluster with a very large index component.

## Nonrecovery Conditions

StopX37/II goes through extreme validation procedures to ensure that recovery attempts are successful. A few conditions do exist which cannot be recovered.

- No recovery attempts can be made on any data sets residing on a VIO type device. This is the only exception when attempting recovery by adding or modifying the secondary space allocation.
- The other nonrecoverable condition concerns volume addition. If the data set resides on an MSS device, StopX37/II cannot attempt a volume add.

Under normal operation, StopX37/II does not allow volume addition on any of the conditions listed below. These conditions can potentially cause problems. If your installation determines that any of the following conditions can be ignored for a given application, a SKIP statement can be used in the SMMSYSnn member to bypass the check for these conditions. The NOCHECK= operand in the rule list can also be used to bypass a volume add check.

**Note:** Caution should be exercised in overriding these checks. Complete testing and verification that multivolume data sets are usable by the application is recommended.

- The job is processing the data set with the EXCP access method (or otherwise processing at the hardware level).
- The job is using NOTE/POINT logic in processing the data set.
- A permanent data set was accessed without the use of the catalog.
- The same data set is referenced by another DD statement in the same job step.
- The same data set is used in another step having a volume refer back to another file name.
- The data set is allocated with the CONTIG JCL parameter.

Installations using SYNC SORT or DF/SORT Releases 7.1 and higher can bypass the volume add EXCP check for the SORTOUT file. Because these versions can handle multivolume SORTOUT data sets, StopX37/II can be allowed to add volumes to these files. This applies to the BetterGener feature provided by SYNC SORT as well as the ICEGENER feature of DFSORT.

**Note:** Do not add a SKIP statement for the SORTWK $nn$  data sets. None of the available commercial sorts support multivolume SORTWK data sets.

## Bypassing Critical Data Sets

When defining the filter/rule lists, an installer should consider situations in which StopX37/II should not attempt recovery. A good example would be to bypass recovery attempts on data sets beginning with SYS1 or SYS2. This reject process can be easily accomplished by coding filter lists that do not allow any recovery attempts.

**Figure 2-1** Example Coding to Bypass Critical Data Sets

---

```
SET MODE=INACT
INC DSN=SYS1/ [or DSN1=SYS1]
INC DSN=SYS2/ [or DSN1=SYS2]
```

---

The previous example does not allow recovery on SYS1 or SYS2 data sets.

## Bypassing VSAM EOV Processing

If you should want StopX37/II to ignore VSAM end-of-volume processing for specific jobs, you can create a table of jobnames to bypass. To do this:

1. Copy SMUSRMOD in *?prefix.BBSAMP* to your library.
2. Modify the usermod to include the jobnames.
3. Receive and apply the usermod, but *never* accept the usermod.

**Warning!** When any new maintenance is applied to PRS41410, modifications to the usermod will be lost. You will need to rework, re-receive, and apply the newer version of the usermod to regain your bypass capabilities.

The VSAM\_ADJ4GB=NO global parameter bypasses StopX37/II processing if a new extent will exceed the 4GB maximum limit on non-extended VSAM data sets. BMC Software recommends not using this global setting and permitting StopX37/II to use the default (VSAM\_ADJ4GB=YES), which allows StopX37/II to adjust requested allocation amounts to fit within the 4GB limit.

### **StopX37/II for VSAM Record-Level-Sharing (RLS) Data Sets**

To activate StopX37/II end-of-volume (EOV) functions for RLS data sets, use the following procedure to install a StopX37/II global user exit in your CICS regions that share RLS data sets:

#### **Important Notes**

- MAINVIEW SRM (SVOS) must be active on all MVS systems sharing RLS data sets.
- MAINVIEW SRM started tasks require RACF UPDATE authority on your RLS spheres (data sets) to perform the required QUIESCE during a SPACVOLA process.
- There must be an active StopX37/II CICS global exit (XFCVSDS) in all CICS regions sharing RLS data sets.
- When performing SPACVOLA on an RLS data set (see “PLT background” on page 2-13 in the installations steps that follow), StopX37/II might cause CICS to issue the DFHFC6023 message that a QUIESCE has been canceled.

#### **RLS Restrictions**

- StopX37/II processes RLS data sets only in a CICS environment.
- The VSAMLIMWARN= global parameter will not apply to VSAM RLS data sets.

## Installation Steps for StopX37/II VSAM RLS Data Sets

- Step 1** Add X37RLS=YES global parameter to your active SMMSYSxx member in *?prefix.BBPARM*.
- Step 2** Add required DFHCSD entries for StopX37/II in all CICS regions sharing RLS data sets. Refer to member STOPXCSD in *?prefix.BBSAMP* for sample entries.
- Step 3** In all CICS regions that use RLS data sets, add the following StopX37/II PLTPI program entry to your active DFHPLTxx (where xx matches PLTPI=xx in CICS startup)

```
DFHPLT TYPE=ENTRY , PROGRAM=STOPXPLT
```

The PLTPI entry can run second or third stage. If you should have more than one program enabling the XFCVSDS global exit, BMC Software recommends that the STOPXPLT entry be run first (see “PLT background”).

### PLT background

STOPXPLT enables program STOPXRLS for CICS global user exit (GLUE) XFCVSDS. STOPXRLS will check for a SPACVOLA request on the RLS data set being QUIESCED. If there is a SPACVOLA in progress, STOPXRLS will cancel the QUIESCE request after adding the volume in CICS. If STOPXRLS cancels a QUIESCE request, subsequent programs that are enabled for XFCVSDS exit might be affected.

When installed, the STOPXPLT program will always enable the XFCVSDS exit for STOPXRLS program. If StopX37/II (MAINVIEW SRM) is not active, or if a SPACVOLA is not in progress, the STOPXRLS exit will proceed with the normal QUIESCE request.

- Step 4** Add the MAINVIEW SRM load library (*?prefix.BBLINK*) to your DFHRPL in all CICS regions sharing RLS data sets.

**Note:** There are only two (2) StopX37/II modules required in the CICS DFHRPL. As an option, you may copy modules STOPXPLT and STOPXRLS from your *?prefix.BBLINK* library to an existing library in the DFHRPL. Refer to member SX37COPY in *?prefix.BBSAMP* for sample JCL to copy required modules.

- Step 5** Provide RACF (or equivalent MVS security product) UPDATE authority for all RLS spheres to your MAINVIEW SRM started tasks.

## SMS Considerations

StopX37/II can prevent out-of-space conditions in a System Managed Storage environment as well as in non-SMS systems. When StopX37/II attempts recovery for SMS-managed data sets, there are some minor differences in operation as compared to a non-SMS environment.

When StopX37/II attempts to perform a volume add for an SMS-managed data set, the `VOLSER`, `MNTYPE`, `USEPOOL`, and `ALTPOOL` statements specified in SPACVOLA rule list are ignored. The storage management subsystem decides to which volume a data set can be extended. Only volumes within the same SMS storage group are chosen for the volume add.

During an add of a new volume, if no volume in the SMS storage group has sufficient space available to accommodate the secondary request, and the SPACSWIR function is active for this recovery attempt, then the secondary space request is reduced by a rule list specified percentage until the allocation succeeds or until it is reduced to zero. This is somewhat different from the non-SMS environment, where the secondary space request is reduced to the largest available contiguous block of space on the volume. Because StopX37/II cannot choose the volume for an SMS-managed data set, there is no way to know which block of space is the largest contiguous one. Therefore, StopX37/II must reduce the secondary space by a set percentage.

**Note:** In this case there is probably very little space anywhere in the storage group and daily space management should be run to provide additional free space on some of the volumes.

In all other cases, StopX37/II works the same way for system managed data sets as it does for all other data sets.

## Data Striping Considerations

The use of sequential data striping was introduced with DFSMS version 1.1 and was originally limited to QSAM and BSAM files only. Since the introduction of DFSMS version 1.2, not only can QSAM and BSAM files be defined as extended format data sets, but VSAM KSDS files can also benefit from the data compression provided by this type of data set.

First, we need to discuss sequential data striping. This technique uses extended sequential data sets that DFSMS can allocate over multiple volumes, preferably on different channel paths and control units to improve performance. When striped data sets were first introduced, it was not possible to add additional volumes to a striped data set after it had been initially allocated. A change was introduced with DFSMS 1.2 that allowed single stripe, permanent data sets to extend to new volumes after initial allocation. The number of stripes can be controlled by the DFSMS Storage Group constructs you specify for the data set. Refer to the IBM Storage Management Library documentation for more details. The key point is you can have single stripe sequential data sets or multi stripe sequential data sets. Another benefit of extended sequential data sets is you are now allowed to have 123 extents per volume.

IBM now supports VSAM KSDS files as extended format data sets. However, you can only have single stripe VSAM data sets. This support was added so VSAM can use the data compression available for the extended format data sets. Also, temporary data sets can be allocated as extended format data sets, but they can not be extended to multiple volumes. If they are allocated as such, they will not be allocated as striped. Refer to APAR OW03815 for further details.

When extended format data sets were first introduced, they had to be allocated on volumes attached to the 3990-3 Storage Control Unit with the extended platform and connected to the host by way of ESCON channels. Now, DFSMS/MVS SAM/VSAM hardware compression requires extended format data sets, but IBM has made modifications to the Licensed Internal Code (LIC) of the control unit portion of several IBM direct access storage subsystems to support the extended format data sets on parallel channels. On these control units, ESCON attachment is no longer a requirement for DFSMS/MVS SAM/VSAM hardware compression. This is true for DFSMS version 1.2. DFSMS version 1.1 still has the ESCON requirement. Refer to IBMLINK item RTA000031564 for more details.

There are now several variations of the extended format (striped) data sets:

- SAM Single Stripe
- SAM Multi Stripe
- VSAM Single Stripe

Below is a summary of the StopX37/II functions and the processing performed on each type of extended format data set.

### **SAM - Primary Space Reduction (SPACPRIM)**

- **Single Stripe**—The primary allocation can be reduced to allow the allocation to remain on the current volume. This can be done even if it drives utilization above the volume's high-water mark.
- **Multi Stripe**—This works the same as for single stripe data sets. Keep in mind the allocation is divided among the number of stripes. When using ISPF option 3.4, the catalog entry shows the total allocations for all stripes. You can specify the data set name and volume for the specific allocations to that stripe. StopX37/II reduces the primary allocation *before* the stripe count is changed.

### **VSAM - Primary Space Reduction (SPACPRIM)**

- **Single Stripe**—StopX37/II reduces the primary allocation of the data set even if it forces the volume utilization above the high water mark.
- **Multi Stripe**—Multi stripe VSAM data sets *are not currently allowed*. This is an IBM limitation. It appears that if you try to allocate a VSAM file using a multi stripe Data Class, it will be allocated with only one stripe.

VSAM files still have the 123 extent per data set limit (regardless of the number of volumes).

### **SAM - Secondary Space Reduction**

- **Single Stripe**—With SPACSECR active for this recovery attempt, StopX37/II reduces the secondary as it did for non-extended files. If SPACSECR is not active for this recovery attempt, StopX37/II forces SPACVOLA processing.
- **Multi Stripe**—With SPACSECR active for this recovery attempt, StopX37/II reduces the secondary as it did for non-extended files. If SPACSECR is not active for this recovery attempt, StopX37/II issues a message and the job fails with an x37 abend.

### VSAM - Secondary Space Reduction

- **Single Stripe**—With SPACSECR active for this recovery attempt, StopX37/II reduces the secondary and continues. If SPACSECR is not active for this recovery attempt, StopX37/II attempts SPACVOLA processing.
- **Multi Stripe**—Multi stripe VSAM data sets *are not currently allowed*.

### SAM - Volume Add (SPACVOLA)

- **Single Stripe**—No change in functionality.
- **Multi Stripe**—These type of data sets *cannot* have volumes added with SPACVOLA processing.

### VSAM - Volume Add (SPACVOLA)

- **Single Stripe**—No change in functionality.
- **Multi Stripe**—Multi stripe VSAM data sets *are not currently allowed*.

### SAM - Add Secondary (SPACSECA)

- **Single Stripe**—No change in functionality.
- **Multi Stripe**—No change in functionality.

### VSAM - Add Secondary (SPACSECA)

- **Single Stripe**—No change in functionality.
- **Multi Stripe**—Multi stripe VSAM data sets *are not currently allowed*.

## Filter and Rule List Sample

The following sample filter list provides an excellent model for developing your installation's filter list. The module accomplishes the following:

- Bypasses recovery attempts for SYS1 or SYS2 data sets.
- Allows secondary space reduction for any VSAM data set with a reduction floor of 10%.

- Allows secondary space allocation for sequential data sets with an allocation amount of 50% of the primary.
- Allows volume adding for temporary data sets. Volume addition to any volume beginning with the same first four characters as the volume the data set is already on.
- Allows volume addition to skip EXCP checking for SORTOUT data sets.
- Allows SPACPRIM to reduce primary on temporary data sets. Reduces the primary by 10%, and does not allow the request to fall below 70% of the original primary request.
- Allows NOCATLG2 to recover NOT-CATLGD-2 conditions for all data sets other than those that begin with SYS1 or SYS2.

The following filter/rule lists and the SMMSYSX1 member can be found in *?prefix.BBSAMP*, where the member name corresponds to the filter/rule list name.

**Note:** If SYNCSORT or DF/SORT release 7.1 and above is not being used by your installation, remove the SKIP control statements so that recovery is bypassed for SORTOUT data sets.

**Table 2-3 Filter and Rule List Sample (Part 1 of 2)**

Member	Contents
SMMSYSX1	<pre> SET FUNC=EX ===== ***Member SMFUNCEX contains the function definitions for StopX37*** ===== SET SKIP=(CHECK=EXCP, DDNAME=SORTOUT)                     </pre>
SMFLSTX1	<pre> SET MODE=INACT INC DSN=SYS1/ INC DSN=SYS2/ SET MODE=ACT INC DSORG=PS                     </pre>
SMRLSTX1	<pre> SET SPACSECR=10 INC DSORG=PS                     </pre>
SMFLSTX2	<pre> SET MODE=INACT INC DSN=SYS1/ INC DSN=SYS2/ SET MODE=ACT INC DSORG=PS                     </pre>
SMRLSTX2	<pre> SET SPACSECA=50 INC DSORG=PS                     </pre>



**Table 2-4 SPACVOLA Restrictions and Recommendations**

<b>SPACVOLA (Add a volume)</b>			
<b>Standard Support</b>	<b>Data Sets Not Supported</b>	<b>Data Set Exclusions From SMFLSTxx Member</b>	<b>Potential Problems</b>
Physical sequential data sets (PS DSORG) ICF VSAM DFSMS-managed and non-DFSMS data sets	BDAM <sup>a</sup> Concatenated  With volume reference (VOL=REF) <sup>b</sup>  For which two DD statements exist in the same job step <sup>c</sup>  Open by another job With maximum number of volumes (59)  DFHSM program names starting with 'ARC' <sup>d</sup> PDS and PDSE (PO DSORG) Existing on all volumes in Storage Group or pool	SAS work files SORTWKxx data sets SYS1 and SYS2 Page data sets Uncataloged data sets	EXCP processed data sets (SORTOUT, and so on) DMS (Sterling Software) RECALLS Multi-stripe data sets (IBM does not allow a volume to be added to multistripe data sets) NOTE/POINT processed data sets OSAM (IMS) data sets OEM vendor software, such as pooling, recovery, compression, or migration products that uses its own internal control blocks to access VOLSERs.

<sup>a</sup>Custom zap available

<sup>b</sup>Except when SUPVOLRF is used

<sup>c</sup>Custom zap available for PGM=IDCAMS (DELETE, DEFINE, REPRO)

<sup>d</sup>Custom zap available

**Table 2-5 NOCATLG2 Restrictions and Recommendations**

<b>NOCATLG2 (Correct duplicate catalog entries)</b>	
<b>Standard Support</b>	<b>Recommended Exclusions</b>
DASD data sets ONLY Non-VSAM data sets ONLY DFSMS-managed and non-DFSMS data sets	DMS (Sterling Software) data set RECALLs

Table 2-6 OPENEMPT Restrictions and Recommendations

<b>OPENEMPT</b>
<b>Standard Support</b>
Physical sequential (PS) and direct (DA) DSORGs only DASD data sets ONLY and they must have: <ol style="list-style-type: none"> <li>1. A VOLSER assigned</li> <li>2. DSORG=PS</li> <li>3. NO concatenation</li> <li>4. First character of RECFM must be F, V, or U</li> </ol>

Table 2-7 OPTBLKSZ Restrictions and Recommendations

<b>OPTBLKSZ (Set optimum blocksize)</b>	
<b>Data Sets Supported</b>	<b>Data Sets Not Supported</b>
DASD and TAPE	With LRECL=0 ISAM VSAM Direct (DA)

Table 2-8 SPACCONV Restrictions and Recommendations

<b>SPACCONV (Change allocation units to blocks)</b>
<b>Data Set Exclusions</b>
Tape ISAM DFSMS-managed

Table 2-9 SPACPRIM Restrictions and Recommendations

<b>SPACPRIM (Reduce primary)</b>
<b>Data Set Exclusions</b>
Tape DFDSS jobs (should be excluded in the SMFLSTxx parmlib member)

Table 2-10 SPACRLSE Restrictions and Recommendations

<b>SPACRLSE (Release unused space)</b>
<b>Support Issues</b>
Only supports new allocations of physical sequential (PS) data sets Only DASD data sets are supported Does NOT work with data sets allocated through ISPF service 3.2 (data set utilities) Data sets that are opened and closed multiple times in the job or step SHOULD be excluded because each close causes remaining space in the current extent to be released SPACRLSE causes an exclusive ENQueue on DSN until step termination

Table 2-11 SPACSECA Restrictions and Recommendations

<b>SPACSECA (Adding secondary)</b>
<b>Data Set Exclusions</b>
PDS (PDS data sets are supported but may be excluded due to performance considerations) SYS1, SYS2 (LINKLIST load libs in particular can cause S106-0C) Tape data sets not supported

Table 2-12 SPACSECB Restrictions and Recommendations

<b>SPACSECB (Always reduce secondary to largest extent)</b>
<b>Recommended Exclusions</b>
VSAM KSDS indexes VSAM KSDSs with key ranges

Table 2-13 SPACSECI Restrictions and Recommendations

<b>SPACSECI (Increase secondary)</b>
<b>Standard Support</b>
Supports only physical sequential (PS) and partitioned data sets (PO) on DASD that are being extended with a secondary allocation

Table 2-14 SPACSECR Restrictions and Recommendations

<b>SPACSECR (Reduce secondary if abend pending)</b>
<b>Recommended Exclusions</b>
VSAM KSDS indexes VSAM KSDSs with key ranges

Table 2-15 SPACSQTY Restrictions and Recommendations

<b>SPACSQTY (Set primary and secondary space quantities)</b>
<b>Standard Support</b>
Supports only new allocations of DASD, non-VSAM data sets Can be used to assign a SPACE quantity for a tape data set so the file can be pooled to a DASD device. This is the same strategy employed in TMM for managing small tape data set mounts.

Table 2-16 VIOALLOC Restrictions and Recommendations

<b>VIOALLOC (Control VIO allocation)</b>
<b>Standard Support</b>
Supports only specified new allocations of temporary DASD data sets that are not referenced in later steps by a VOL=REF statement.

---

## Tip for StopX37/II Stand-Alone Users

If you are running StopX37/II *without* the MAINVIEW interface, you can use the X37PARMS CLIST to determine the parmlib data set members that contain the active StopX37/II parameter definitions. Operator commands to refresh the various functions and pools are also shown when the clist is invoked.

The X37PARMS member is in *?prefix.BBSAMP* on the installation tape and is invoked by issuing the one of the following TSO commands:

```
%X37PARMS
```

```
CALL `?BBLINK(COSSR358)`
```

Either command will generate an output list that will assist you in making changes to the parmlib members.

**Note:** Change *?prefix.BBLINK* to the library that contains the MAINVIEW SRM StopX37/II modules.



---

---

## Chapter 3 Functions Reference

The StopX37/II functions are described in detail in this chapter. Use the following list find the function you are looking for.

NOCATLG2 - Prevent NOT CATLG 2 Errors . . . . .	3-3
OPENEMPT - Initialize Empty Data Sets . . . . .	3-12
OPTBLKSZ - Set Optimum Blocksize. . . . .	3-16
SPACCONV - Change Allocation Units to Blocks . . . . .	3-21
SPACPRIM - Reduce Primary Space Allocation . . . . .	3-25
SPACRLSE - Release Data Set Space at Close . . . . .	3-32
SPACSECA - Add Secondary Allocation Quantity . . . . .	3-35
SPACSECB - Reduce Secondary Allocation to Best Fit . . . . .	3-39
SPACSECI - Increase Secondary Allocation Quantity. . . . .	3-42
SPACSECR - Reduce Secondary Allocation to Largest Extent . . . . .	3-47
SPACSQTY - Set Primary and Secondary Space for Data Set. . . . .	3-52
SPACSWIR - Reduce Initial Allocation on Volume Addition . . . . .	3-60
SPACVOLA - Add Volume During Allocation . . . . .	3-64
VIOALLOC - Control VIO Allocation. . . . .	3-78

## 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. Refer to “Filter and Rule List Sample” on page 2-17 for model to use when configuring your environment.

Functions are defined in SET statements. You can change parameters

- by editing the member directly
- through the Functions option on the Parmlib Members pop-up menu of the EZSRM Menu (see the *MAINVIEW SRM User Guide and Reference*.)

StopX37/II functions do not operate for DFDSS jobs.

## NOCATLG2 - Prevent NOT CATLG 2 Errors

NOCATLG2 intercepts conditions that lead to the issuing of message IEF377I with a reason of NOT CATLGD 2. NOCATLG2 must process SMS-managed data sets during allocation since a JCL error will occur if a job attempts to catalog already existing SMS data sets. However, for non-SMS data sets NOCATLG2 can process in either place. The NOCATWHEN rule list SET parameter dictates where NOCATLG2 processing will occur for non-SMS data sets.

When OS/390 attempts to catalog a data set during step termination and a catalog entry already exists for the data set, a message IEF377I with a reason of NOT CATLGD 2 is issued. The step completes with a normal condition code, but later steps that attempt to access the data set by way of the catalog use an invalid catalog entry. For SMS-managed data sets, a JCL error occurs when the step is started.

For non-SMS managed data sets, if an older version of the data set really does exist on the volume specified by the invalid catalog entry, the step uses the wrong data as input. If the data set does not exist on the volume, the step abends with a 213.

NOCATLG2 can recatalog the data set using the new volume. This facility works with both disk and tape data sets.

The NOCATLG2 function prevents data sets from being created but not cataloged when the same data set name exists on the catalog.

Without DFSMS, when a new data set is cataloged with a name that already exists in the catalog, OS/390 issues a job log message at step termination. The following is an example of such a message:

```
14.53.43 JOB03558  IEF377I  EMPCRMX  STEP10
EMPCRM.NOCAT.TEST NOT CATLGD 2
```

There is no condition code set to allow the job to be stopped by JCL. The new data set exists on the volume where it was allocated, but *it is not cataloged*. It can be accessed only with a specific volume serial number. The system catalog still refers to the original data set with that name. Subsequent jobs that refer to that data set by the catalog entry access the old data set. This leads to production problems, job reruns, and wasted DASD space.

Under DFSMS, this same situation causes a JCL error, again leading to lost time.

NOCATLG2 detects data sets which have names that already exist in the catalog and applies user-defined corrective action automatically in both DFSMS and non-DFSMS environments.

## NOCATLG2 Considerations

NOCATLG2 addresses certain limitations within the OS/390 operating system which arise when jobs attempt to allocate a new data set using the same name as a data set which already exists in the system catalog.

In a non-SMS environment, data sets are allocated at the start of a job step, but not cataloged until step termination. When OS/390 attempts to catalog a data set during step termination, no error occurs if a catalog entry already exists for the data set. Instead, message IEF285I NOT-CATLGD-2 is issued, and the step completes normally. The catalog entry is not changed. Subsequent steps which reference the data set via the catalog use the invalid catalog entry. This may result in corrupted data later on.

SMS-managed data sets are cataloged at the start of the job step as part of the allocation process. If an attempt is made to create an SMS-managed data set using the same name as a data set which already exists in the catalog, a JCL error occurs and the job is flushed with messages IGD17273I, IGD17001I, or IGD17101I.

NOCATLG2 addresses both of these types of errors. Several different options allow recovery from these conditions. The recovery options are specified in the NOCATLG2 operand of the StopX37/II rule list statement. Each type of action is described below.

Recovery Option	Action
FAIL	Indicates that the job should fail with a JCL error.
DELETE	Indicates that the old data set should be deleted. The old data set will also be uncataloged. If the old data set has not expired (that is, if its expiration date has not passed), it is deleted only if PURGE=YES is also specified. If PURGE=NO is specified or defaulted, the unexpired old data set is not deleted, and the catalog is not changed.
UNCATLG	Indicates that the old data set should only be uncataloged. Thereafter, the old data set can only be referred to with a volume serial number.

Recovery Option	Action
RENAME	Indicates that the old data set should be renamed and recataloged. The NOCATLG2 function generates a new name by adding a 3-byte, second-level qualifier (specified in the global parameter NOCATPFEX, default='BAB') after the original high-level qualifier and a time stamp as the third-level qualifier, followed by the remainder of the original data set name up to 44 bytes. Example: Original data set      SYS9.OLD.DATA Renamed data set    SYS9.BAB.Tmmss.OLD.DATA
CANCEL	Cancels the job only if NOCATWHEN=TERM.
NO	Allows the job step to complete with a normal return code. Subsequent job steps may abend or process invalid data. For SMS-managed data sets, the job fails immediately with a JCL error.
FLUSH	Generates a NOT CATLGD2 message for the data set; the remainder of the job steps are flushed.
OPER	Issues a message to the system console by way of WTOR, allowing the operator to reply with the desired option.

The SVOS address space must have RACF authority to scratch or rename any data sets that will be processed by the NOCATLG2=YES/SCRATCH/RENAME options. These options generate modify commands from the individual address spaces to the SVOS address space to actually perform the function.

An IGD17001I message appears for SMS data sets processed with NOCATLG2=RENAME/SCRATCH even though the SCRATCH/RENAME is successful.

## Bypassing NO CATLG2 Processing

If you should want StopX37/II to ignore NO CATLG 2 processing for specific jobs, you can create a table of job names to bypass. To do this:

- Step 1** Copy SMUSRMD3 in *?prefix.BBSAMP* to your library.
- Step 2** Shift the usermod to the left two columns.
- Step 3** Modify the usermod to include the job names to be bypassed.
- Step 4** Receive and apply the usermod, but never accept the usermod.

**Warning!** When any new maintenance is applied to the PRS41310 module, modifications to the usermod will be lost. You will need to rework, re-receive, and apply the newer version of the usermod to regain your bypass capabilities.

## Considerations for NOCATLG2 for SMS-Managed Data Sets

When NOCATLG2=DELETE, RENAME, or UNCATLG processing is performed for a new, SMS-managed data set, NOCATLG2 calls SVOS to perform the scratch, rename, or uncatalog of an existing SMS-managed data set with the same name. Because SVOS is used to perform these functions, the SVOS address space must be active if NOCATLG2 processing is to be performed. If NOCATSMS=NO is specified in the SMMSYS $nn$  global parameter member, the NOCATGL2 processing will not be performed for SMS-managed data sets.

### Global Parameters

The global parameters that affect the NOCATLG2 function are described in the table below.

**Table 3-1 Global Parameters—NOCATLG2**

Parameter	Purpose
MSGPREF	MAINVIEW SRM message identifier prefix
NOCATPFX	Second-level qualifier used when renaming a data set during NOCATLG2 processing
NOCATPRG	Allows data sets to be scratched before the expiration date during NOCATLG2 processing
NOCATSEC	Level of security performed before scratching or renaming a data set during NOCATLG2 processing (NONE, CREATE, READ, UPDATE, ALTER)
NOCATSMS	Allows SMS-managed data sets to be renamed, uncataloged, or scratched during NOCATLG2 processing
NOCATVOL	Allows a new data set to be allocated to the same volume it was previously cataloged on during NOCATLG2 processing
NOCATWHEN	Specifies when NOCATLG2 processing is to occur for non-SMS data sets (allocation or step termination)
SIZEISPRIM	Determines if the SIZE filter/rule list parameter includes only the size of the primary extent or the size of the primary and one secondary extent

### Rule List Parameters

SET Statement (in member SMRLST $xx$ )

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNT $xx$ member
NOCATLG2=xxxxxxx	Specifies the action to be taken on the old data set

Parameter	Description
NOCATWHEN= <i>TERM/ALLOC</i>	Specifies when NOCATLG2 processing will occur for a non-SMS data set
PURGE= <i>YES/NO</i>	Specifies action on an unexpired data set to be deleted

### Rule List Parameter Explanations

#### EVENTID=

**Purpose:** Specifies the identifier assigned to a user event in SMEVNT*xx*. 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 SMEVNT*xx*.

**Default:** None

#### NOCATLG2=

**Purpose:** Specifies the action to be taken when a NOT CATLGD2 condition occurs. This condition occurs when a data set has a disposition of (NEW,CATLG) and the same name already exists in the catalog. The NOCATLG2 function can cancel the job, or rename or delete or uncatalog the old data set. Also see the parameter PURGE.

**Syntax:** NOCATLG2=*xxxxxxx*

where *xxxxxxx* is a value from the following list:

FAIL	causes the job to fail
RENAME	renames the old data set
DELETE	deletes the old data set
UNCATLG	uncatalogs the old data set
CANCEL	<p>cancels the job</p> <p>If NOCATWHEN=ALLOC, the job is canceled before the current step executes. If NOCATWHEN=TERM, the job is canceled after the current step ends, that is, all following steps are flushed.</p>

NO	jobstep completes with a normal return code Subsequent jobsteps may abend or process invalid data. For SMS-managed data sets, the job fails immediately with a JCL error.
FLUSH	data set receives NOT CATLGD2 message; the remainder of the jobsteps are flushed
OPER	issues a message to the system console by way of WTOR, allowing the operator to reply with the desired option

Default: None

### **NOCATWHEN=**

Purpose: Specifies when NOCATLG2 processing will occur for a non-SMS data set.

Syntax: NOCATWHEN=*TERM/ALLOC*

**TERM** NOCATLG2 processing will occur during step termination.

**ALLOC** NOCATLG2 will occur during OS/390 data set allocation processing.

Default: NOCATWHEN=ALLOC

**Note:** When NOCATWHEN=ALLOC and no volser is specified, you will not be able to filter on parameter VOL=. To be able to filter on VOL=, you must specify NOCATWHEN=TERM.

### **PURGE=**

Purpose: Specifies whether the old data set should be purged if NOCATLG2=DELETE and the data set has an unexpired expiration date. PURGE=YES allows the unexpired data set to be deleted; PURGE=NO prevents the unexpired data set from being deleted. This is an optional parameter. If this parameter is not specified, the NOCATPRG global parameter is used.

Syntax: PURGE=*YES/NO*

Default: None. The NOCATPRG global parameter setting is used if PURGE is not set.

## Filter List Parameters Not Supported

The following filter list parameters are not supported by NOCATLG2:

- GDGVER
- VALUE

## Usage Notes

NOCATLG2 acts on new non-VSAM data sets when the new data set is to be cataloged with a name that already exists in the catalog. The actions described in “NOCATLG2 Considerations” on the previous page can be specified.

NOCATLG2 takes no action if the data set is

- dynamically allocated, and the action is RENAME or DELETE
- DFSMS-managed, and the action is UNCATLG

NOCATLG2 performs an UNCATLG action if the requested action is RENAME but the old data set does not have a volume ID (as when a data set has been deleted but not uncataloged).

For DFSMS-managed data sets, a RENAME on a migrated data set does *not* cause a recall. For non-DFSMS-managed data sets, either a RENAME or a DELETE on a migrated data set does *not* cause a recall.

Note that the NOCATLG2 function must be specified in the SMFUNCxx member to be available and must be set active to provide service.

## Messages

**Note:** Message SVM4105E will be issued stating that NOCATLG2 was unable to process this data set.

If ACTION=DELETE, PURGE=NO is specified, and the data set has an unexpired expiration date, the NOCATLG2 function is not able to correct the situation. The original data set remains cataloged, and the new data set receives a NOT CATLGD 2 error. An IEC614 error message is produced by OS/390, referring to a SCRATCH operation; NOCATLG2 generates a SCRATCH ERROR console message. *Note that this could result in an uncataloged data set on an DFSMS volume.*

If NOCATLG2=DELETE and NOCATWHEN=ALLOC are specified and there is a catalog entry that has no actual data set on the DASD volume referenced in the catalog entry, the following messages will be issued:

SVM4122W SCRATCH FOR DATASET ON VOLUME *xxxxxx* FAILED (RC= 8-104190401)

SVM4120I DATASET WILL BE RECATALOGED FROM VOLUME *xxxxxx* TO VOLUME *xxxxx*

This message is merely informing that an invalid catalog entry for the data set was found. NOCATLG2 attempted to delete the data set, but the data set did not exist on the volume in the catalog entry. NOCATLG2 continues and recatalogs the data set to the new volume.

Messages issued by NOCATLG2 include:

SVM4120I DATASET WILL BE RECATALOGED FROM VOLUME *xxxxxx* TO VOLUME *xxxxx*

SVM4122W SCRATCH FOR DATASET ON VOLUME *xxxxxx* FAILED (RC= 8-104190401)

SVM4125S ENTER NOCATLG2 OPTION FOR DSN=*dsn*,  
CVOL=*curvol*,NVOL=*newvol*

SVM4126S NOCATLG2 NOT ALLOWED DUE TO NOCATVOL=DIFF

SVM4127S UNCATALOGED DATASET EXISTS ON TARGET VOLUME  
*volume*

SVM4128S NOCATLG2=UNCATLG/RENAME FOR SMS DATASET, BUT  
NEW ALLOCATION FAILED

SVM4129S NOCATLG2=UNCATLG/DELETE/RENAME FOR SMS DATASET  
BUT NOCATSMS=NO

SVM4130W NOCATLG2=DELETE/RENAME FOR SMS DATASET BUT  
EXISTING DATASET IS NOT DASD

SVM4131S NOCATLG2=DELETE/RENAME BUT NEW VOLUME = OLD  
VOLUME. NOCATLG2 FAILED

SVM4132S NOCATLG2=UNCATLG BUT NEW VOLUME = OLD VOLUME  
UNCATLG2 FAILED. NOCATVOL=SAME WILL OVERRIDE

SVM4146W NOCATLG2=NO WAS SPECIFIED IN RLST. RECAT FAILED

## Example

This example shows you how to intercept all NOT CATLGD 2 error situations.

- For TSO users, fail the job.
- For test jobs in batch, delete the data set, even if it is unexpired.
- For production jobs, rename the old data set.
- For all other data sets, uncatalog the old data set

### SMFUNC14 member

```
SET NAME=NOCATLG2 ACTIVE=YES      The function is defined and
MSG=I SMF=I                       activated.
FLST=16 RLST=16
DESC='INTERCEPT NOT CATLGD 2'
```

### SMFLST16 member

```
SET MODE=ACT                       Select all data sets.
INC DSN=/'
```

### SMRLST16 member

```
SET NOCATLG2=FAIL                  Cause jobs to immediately fail for
INC XMODE=TSO                      TSO users.

SET NOCATLG2=DELETE PURGE=YES      Unconditionally delete data sets
INC XMODE=JOB                       in test batch jobs.
JOBCLASS=T

SET NOCATLG2=RENAME                Rename data sets for production
INC JOBCLASS=P                      jobs.

SET NOCATLG2=UNCATLG              For all other data sets, uncatalog
INC DSN=/'                          the old data set.
```

## OPENEMPT - Initialize Empty Data Sets

The OPENEMPT function initializes empty data sets at job step termination.

Data sets that are allocated but never opened are a potential hazard to subsequent jobsteps and jobs. OS/390 does not initialize a data set at allocation; a new data set is only initialized when it is opened. If this uninitialized data set is later opened by another program, one of two things may happen:

- The job receives an I/O error because an end-of-file mark has not been written
- The job reads a record left on the volume from a prior data set and possibly processes it as normal data

OPENEMPT can intercept and prevent most uninitialized data set problems by initializing empty data sets at job step termination.

### Rule List Parameters

SET Statement (in member SMRLST $_{xx}$ )

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNT $_{xx}$ member

### Rule List Parameter Explanations

#### EVENTID=

**Purpose:** Specifies the identifier assigned to a user event in SMEVNT $_{xx}$ . 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 SMEVNT $_{xx}$ .

**Default:** None

## Filter List Parameters Not Supported

The following filter list parameters are not supported for OPENEMPT:

BUFSP	CATALOG	CISIZE	CURDAY
CURSPACE	CURTIME	ERASE	EXTENT
GDGVER	IMBED	JOBSDAY	JOBSTIME
LABELTYP	OWNER	REPL	REUSE
VALUE			

## Usage Notes

OPENEMPT provides services for new non-VSAM data sets other than partitioned data sets. OPENEMPT obtains control at the termination of the jobstep and checks all data sets for an uninitialized condition. If found, the uninitialized data sets are opened and closed to initialize them.

An additional benefit of OPENEMPT is that the initialized data set can be processed by the SPACRLSE function to release allocated unused space. If the data set remained uninitialized, it would retain its full allocated space.

### Exclusions

OPENEMPT ignores the data set in the following circumstances:

- The step is terminating normally, and the normal termination disposition is delete: DISP=(NEW,DELETE)
- The step is terminating abnormally, and the abnormal termination disposition is delete: DISP=(NEW,*anything*,DELETE)
- The data set has no volumes assigned
- The data set is a PDS
- The data set is subsequent to the first data set in a concatenation of data sets (only the first data set is processed due to OS/390 architectural restrictions)
- The data set is allocated to VIO
- No DCB information specified. DCB information uses a referback (DCB=\*STEP.DD).

Note that the OPENEMPT function must be specified in the SMFUNCxx member to be available and must be set active to provide services.

### Messages

The messages issued by OPENEMPT are:

```
SVM0500I  job,step,dd,dsn INITIALIZED
```

OPENEMPT found the indicated data set had not been initialized. The data set was opened and closed by OPENEMPT to set the end-of-file mark.

```
SVM0501I  job,step,dd,dsn NOT INITIALIZED, REASON=RECFM
```

The indicated data set could not be initialized because record format information was not available; for example:

```
SVM0501I  EMPCRM,STEP10,DD2,EMPCRM.OPENEMPT.TEST NOT  
INITIALIZED,REASON=RECFM
```

---

```
//EMPCRMX  JOB ( 3500 ) , 'MORRIS' ,MSGCLASS=R , CLASS=F  
//STEP10   EXEC PGM=IEFBR14  
//DD2      DD  DSN=EMPCRM . OPENEMPT . TEST ,  
//          DISP=( NEW ,CATLG ,DELETE ) ,  
//          SPACE=( TRK , ( 1 , 1 ) ) ,UNIT=SYSDA
```

---

```
SVM0501I  job,step,dd,dsn NOT INITIALIZED, REASON=DDNAME
```

The indicated data set could not be initialized because the ddname was a duplicate of another; for example:

```
SVM0500I  EMPHRFX,STEP10,DD2,EMPHRF.OPENEMPT.TEST  
INITIALIZED
```

```
SVM0501I  EMPHRFX,STEP10,DD2,EMPHRF.OPENEMPT.TEST2 NOT  
INITIALIZED,REASON=DDNAME
```

---

```
//EMPHRFX  JOB ( 3500 ) , 'FITZGERALD' ,MSGCLASS=R , CLASS=F  
//STEP10   EXEC PGM=IEFBR14  
//DD2      DD  DSN=EMPHRF . OPENEMPT . TEST  
//          DISP=( NEW ,CATLG ,DELETE ) ,  
//          SPACE=( TRK , ( 1 , 1 ) ) ,UNIT=SYSDA ,  
//          DCB=( LRECL=80 ,BLKSIZE=16000 ,RECFM=FB )  
//DD2      DD  DSN=EMPHRF . OPENEMPT . TEST2 ,  
//          DISP=( NEW ,CATLG ,DELETE ) ,  
//          SPACE=( TRK , ( 1 , 1 ) ) ,UNIT=SYSDA ,  
//          DCB=( LRECL=80 ,BLKSIZE=16000 ,RECFM=FB )
```

---

SVM0501I *job,step,dd,dsn* NOT INITIALIZED, REASON=OPEN

The indicated data set could not be initialized because it was already open. This normally indicates the program terminated without closing the data set.

### Example

This example shows you how to use OPENEMPT on all data sets other than those in programs under development.

#### SMFUNC11 member

SET	NAME=OPENEMPT	ACTIVE=YES	The function is defined and activated. Note that there is no RLST parameter.
	MSG=I	SMF=I	
	FLST=03		
	DESC='OPEN EMPTY DATA SETS'		

#### SMVARS02 member

SET	VARIABLE=TESTPGMS	Set a variable to identify test jobs by program name. Any job that begins or ends with T, or that begins with DBT is considered to be a test job.
	VALUE=T*	
	VALUE=*T	
	VALUE=DBT*	
SET	VARIABLE=TESTJOBS	Set a variable to identify test jobs by job class.
	VALUE=T	
	VALUE=U	
	VALUE=B	

#### SMFLST03 member

SET	MODE=ACT	Select resources for OPENEMPT based on program name or on the job class field.
INC	PGM=&TESTPGMS	
INC	JOBCLASS=&TESTJOBS	

## OPTBLKSZ - Set Optimum Blocksize

The OPTBLKSZ function forces an optimum blocksize for new data set allocations on DASD and tape. With the use of optional parameters, the BLKSIZE may also be changed on OLD or INPUT data sets.

The blocksize of a data set has a direct effect on the storage efficiency of DASD devices. Generally, the larger the blocksize, the more records fit on a track. This not only achieves better utilization of disk space, it also reduces the number of I/Os required to read the data set. Although very small data sets may not benefit significantly from the denser record storage, the minimum allocation by OS/390 is a single track; therefore, it generally is more effective to allocate larger block sizes. The only disadvantage is that larger block sizes require larger buffers in memory. The following table shows the effect of different block sizes on DASD capacity.

Record Density, 3380 DASD, 80-byte records			
Blocksize	Records/track	Records/cylinder	Percent space used <sup>a</sup>
4240	530	7950	90.0
6320	553	8295	93.7
23440	586	8790	98.9

<sup>a</sup> Source: IBM 3380 Direct Access Storage: Reference Summary

OPTBLKSZ allows the data center to set and enforce standard data set block sizes. Using rule lists, different block sizes can be applied to different groups of data sets.

### Global Parameters

The global parameters that affect OPTBLKSZ are described below.

**Table 3-2 Global Parameters—OPTBLKSZ (Part 1 of 2)**

Parameter	Purpose
BLKINPUT	Changes block size for input data sets.
BLKOLDSR	Changes blocksize for output data sets opened with disposition of old or shared.
TRKCYL	Specifies the number of tracks per cylinder for the default device type. The value specified for 3380/3390/9345 devices should be fifteen. (Note that this specification is the same as the SCDS base configuration DEFINE under ISMF for DFSMS.) TRKCYL and TRKLEN are used by the DASDPOOL function to convert allocations in tracks or cylinders to megabytes for volume selection based on available space; for example, VOLSEL=BESTFIT. The information specified on these two parameters should reflect the devices that are most prevalent in your environment.

Table 3-2 Global Parameters—OPTBLKSZ (Part 2 of 2)

Parameter	Purpose
TRKLEN	<p>Specifies the number of bytes per track for the default device type. Valid values are:            3380 – 47,476            3390 – 56,664            9345 – 46,456</p> <p>Note that this specification is the same as the SCDS base configuration DEFINE under ISMF for DFSMS.)</p> <p>TRKCYL and TRKLEN are used by the DASDPOOL function to convert allocations in tracks or cylinders to megabytes for volume selection based on available space; for example, VOLSEL=BESTFIT. The information specified on these two parameters should reflect the devices that are most prevalent in your environment.</p>

### Rule List Parameters

SET Statement (in member SMRLST $xx$ )

Parameter	Description
BLKSIZE= $nnnnn$	A blocksize in the range 0–32,760
EVENTID= $xxxxx$	Identifies a user event defined in an SMEVNT $xx$ member
FORCE= <u>YES/NO</u>	Overrides program specified blocksize

### Rule List Parameter Explanations

#### BLKSIZE=

**Purpose:** Specifies a blocksize to be assigned to the new data set. A specification of zero directs the use of a system-assigned blocksize. This is a required parameter; no processing occurs if a BLKSIZE is not specified.

**Syntax:** BLKSIZE= $nnnnn$

where  $nnnnn$  is a number in the range 0–32,760.

**Default:** None

#### EVENTID=

**Purpose:** Specifies the identifier assigned to a user event in SMEVNT $xx$ . 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 SMEVNT*xx*.

Default: None

#### **FORCE=**

Purpose: Specifies whether any program specified blocksize should be overridden.

Syntax: FORCE=*YES/NO*

If YES, the value specified will be used to override a program specified blocksize; if NO, a program specified blocksize will NOT be overridden.

Default: FORCE=NO

#### **Parameters Not Supported**

The following filter list parameters are not supported for OPTBLKSZ:

- GDGVER
- VALUE

## **Usage Notes**

OPTBLKSZ provides services for both new and old DASD and tape data sets with record formats of undefined, variable, fixed, or fixed blocked. *It does not operate on ISAM, direct, or VSAM data sets.* It will override block sizes assigned in program DCBs if FORCE=YES is specified. The data set allocation *must* be specified with both record length and a blocked record format (FB, VB, FBA, and so on); without these two pieces of information, a block size is not assigned.

A BLKSIZE rule list parameter is required; if not specified in a rule, the data set selected for processing is ignored.

If the LRECL of the selected data set is zero, or larger than the rule-specified block size (SET BLKSIZE=), OPTBLKSZ takes no action. Otherwise, a new block size is assigned. For fixed and fixed blocked data sets, the rule-specified block size is adjusted to the next lower multiple of the LRECL; for variable and undefined data sets, the rule-specified block size is used as is.

Note that a blocksize of zero can be specified to request a system-assigned blocksize; in this case, OPTBLKSZ sets the BLKSIZE to 0. OS/390 System-Determined Blocksize (SDB) must be driven to reset the BLKSIZE to the optimum value. For SDB to reset the value, the data set must be opened in the same jobstep in which it is allocated.

## Messages

The message issued by OPTBLKSZ when a blocksize is changed is:

```
SVM3347I  job,step,dd,dsn BLKSIZE CHANGED nnnnn/nnnnn
```

The first *nnnnn* gives the original blocksize specified in the JCL; the second *nnnnn* gives the blocksize set by OPTBLKSZ.

The message issued by OPTBLKSZ when a specific blocksize is requested by program name and FORCE=YES is coded is:

```
SVM3351I  job,step,dd,dsn BLKSIZE nnnn IN PGM=name REMOVED
```

Note that the OPTBLKSZ function must be specified in the SMFUNCxx member to be available and must be set active to provide service.

## Example

This example shows you how to

- specify blocksize for all data sets except certain system-level PDSs
- run in simulate mode for data sets in jobs with BLKEXEMPT in the third job account field

### SMFUNC73 member

```
SET  NAME=OPTBLKSZ ACTIVE=YES
      MSG=I  SMF=N
      FLST=7D RLST=7E
      DESC='ASSIGN OPTIMUM BLOCKSIZE'
```

The function is defined and activated. No SMF records will be produced.

### SMFLST7D member

```
SET  MODE=SIM
      INC    JOBACCT3=BLKEXEMPT
```

The function is to run in simulate mode for all data sets in jobs with BLKEXEMPT in the third job accounting field.

```
SET  MODE=ACT
      EXC   HLQ=SYS* DSORG=PO
      RECFM=U
      INC   DSN=/'
```

The function is to run in active mode for all selected resources. All data sets are selected, except system-level undefined partitioned data sets.

### SMRLST7E member

SET	BLKSIZE=3861	Specify a small blocksize for
INC	DSORG=PO LLQ=JCL/	JCL PDSs, since they consist
		of many members with 50
		records or less.
SET	BLKSIZE=23440	Specify half-track blocking for
INC	MAXSIZE>50MB DEVTYPE=3380	large data sets on all 3380 and
INC	MAXSIZE>50MB DEVTYPE=3390	3390 devices.
SET	BLKSIZE=4629	For large data sets on all other
INC	MAXSIZE>50MB DEVTYPE=DASD	devices, the blocksize is set
		for half- or third-track blocking.
SET	BLKSIZE=11476	For all other data sets on 3380
INC	DEVTYPE=3380	devices, the blocksize is set to
		quarter-track blocking.
SET	BLKSIZE=13682	For all other data sets on 3390
INC	DEVTYPE=3390	devices, the blocksize is set to
		quarter-track blocking.
SET	BLKSIZE=25600	The blocksize for tape data
INC	DEVTYPE=TAPE	sets is set to 25KB.
SET	BLKSIZE=27999 FORCE=YES	For all programs that start with
INC	PROGRAM=PRO031/	PRO031, set the blocksize to
		27999 and override any
		program blocksize.

# SPACCONV - Change Allocation Units to Blocks

The SPACCONV function converts space allocation values from cylinders or tracks to blocks, for both primary and secondary allocation quantities.

Space allocation in blocks is advantageous because allocation in blocks is device independent; the system determines the number of tracks or cylinders necessary to hold the requested number of blocks, taking into account the device type. Allocation in units of cylinder or track is device-specific, because the size of the allocation unit varies with the device type.

SPACCONV is useful in allocating the correct amount of space when new device types are put into use, such as when converting from 3380 to 3390 devices. The old device type (for which the JCL space parameters are coded) is identified by the TRKLEN and TRKCYL parameters. SPACCONV automatically converts track and cylinder space requests into blocks, which, when applied by OS/390 during allocation to the new device type, allocates a proportionally larger or smaller total space.

## Rule List Parameters

SET Statement (in member SMRLST $xx$ )

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNT $xx$ member
ROUND=YES/NO	Round allocation in blocks to cylinder boundary
TRKCYL= $nn$	Specifies the tracks per cylinder
TRKLEN= $nnnn$	Specifies the bytes per track

## Rule List Parameter Explanations

### EVENTID=

**Purpose:** Specifies the identifier assigned to a user event in SMEVNT $xx$ . 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 SMEVNT $xx$ .

**Default:** None

**ROUND=**

**Purpose:** Specifies that the converted space allocation should be rounded up to a cylinder boundary.

This is an optional parameter. If not used, the SMRLSTxx member can be omitted for this function.

**Syntax:** ROUND=*YES/NO*

**Default:** None

**Note:** Both SPACCONV and SPACSQTY functions allow the ROUND parameter to be specified. If both are specified for the same data set, the value in SPACCONV will be the value assigned to the data set, because SPACCONV receives control after SPACSQTY.

**TRKCYL=**

**Purpose:** Specifies the number of tracks per cylinder of the *source volume* for the SPACCONV function. The value is used to calculate the proper size of an allocation on new devices for specifications based on devices no longer used.

**Syntax:** TRKCYL=*nn*

where *nn* is a number in the range of 1-99.

**Default:** None (if not specified, the TRKCYL value in SMMSYSxx is used)

**TRKLEN=**

**Purpose:** Specifies the number of bytes per track of the *source volume* for the SPACCONV function. The value is used to calculate the proper size of an allocation on new devices for specifications based on devices no longer used.

**Syntax:** TRKLEN=*nnnnn*

where *nnnnn* is a number in the range of 1-99999.

**Default:** None (if not specified, the TRKLEN value in SMMSYSxx is used)

## Filter List Parameters Not Supported

The following filter list parameters are not supported for SPACCONV:

- GDGVER
- VALUE

## Usage Notes

SPACCONV applies to new DASD data sets (non-VSAM and JCL-defined VSAM). *It does not apply to tape data sets or ISAM data sets.* The original space allocation must be in cylinders or tracks; if it is already in blocks, no action is taken. Both primary and secondary allocation quantities are converted.

Conversion of space allocation from cylinders or tracks to blocks is based on the bytes/track and tracks/cylinder values specified in the TRKLEN and TRKCYL rule list parameters. An average block size of 9000 is used (this yields 5 blocks per track on both 3380 and 3390 devices). The calculated number of blocks is always increased by one. If ROUND is specified, rounding and alignment to a cylinder boundary is performed by OS/390. (Although an average block size of 9000 is used to calculate space, block size specified for the data set is not modified.)

If TRKLEN and/or TRKCYL are not specified in the rule list, the values are taken from the TRKLEN and TRKCYL parameters in the SMMSYSxx parmlib member. However, these SMMSYSxx parameters are used to represent the *normal* devices for allocation; accepting these values as a default causes the SPACCONV function to allocate the full size specified in the JCL.

The TRKLEN and TRKCYL specifications can have different values on different rules. This is useful when upgrading from two obsolete device types to a new device type, such as from 3350s and 3380s to 3390s.

**Note:** The ISPF data set information display in functions 3.4 and 3.2 do not always give the correct values for secondary space in blocks, particularly when the data set is allocated without a blocksize. The information provided by IEHLIST is correct, however.

Note that the SPACCONV function must be specified in the SMFUNCxx member to be available and must be set active to provide services.

## Messages

Messages issued by SPACCONV include:

```
SVM3448I  job,step,dd,dsn SPACE CONVERTED TO prim/sec BLOCKS
SVM3318I  job,step,dd,dsn ROUND=y/n ASSIGNED
```

## Example

This example shows you how to specify space conversion for unit specifications of 3350 and 3380.

### SMFUNC33 member

```
SET  NAME=SPACCONV ACTIVE=YES  The function is defined and
     MSG=I  SMF=I               activated.
     FLST=33 RLST=33
     DESC='CONV SPACE TO BLOCKS'
```

### SMFLST33 member

```
SET  MODE=ACT                 The function is to run in active
     INC UNIT=3350             mode for all allocation requests to
     INC UNIT=3380             devices no longer used (3350s and
                               3380s).
```

### SMRLST33 member

```
SET  ROUND=YES                Convert space for allocations coded
     TRKLEN=47476 TRKCYL=15    for 3380 devices. Round allocations
     INC UNIT=3380             to an integral number of cylinders.

SET  TRKLEN=19069 TRKCYL=30    Convert space for allocations coded
     INC UNIT=3350             for 3350 devices.
```

# SPACPRIM - Reduce Primary Space Allocation

SPACPRIM stops *Space-Not-Available* conditions during step initialization. These conditions are associated with disk space availability and management. Space-Not-Available conditions arise when OS/390 is unable to satisfy space allocation for a new data set.

SPACPRIM interacts with modules in the OS/390 allocation function. When a user or job is requesting allocation of a new data set, the primary space request must be satisfied in no more than five extents. If the space is unavailable on the requested volume, SPACPRIM can reduce the primary space request and retry the allocation. This allows an initiator to start a job that would normally be failed due to lack of space.

SPACPRIM prevents Space-Not-Available conditions for all types of non-VSAM data sets, including PDSEs. SPACPRIM also prevents these errors from occurring for SMS-managed VSAM data sets which are allocated in JCL.

The SPACPRIM function attempts to satisfy a data set's primary allocation by reducing the requested space in user-specified increments.

If OS/390 cannot find enough free space to satisfy the primary allocation amount specified in the JCL, the job fails with a JCL error. However, it is possible that a smaller primary allocation for the data set can still allow successful processing because:

- The primary space in the JCL might be set at the maximum space necessary for conditions that only occur infrequently
- When the primary space is filled, the data set still has secondary allocations to handle expansion

SPACPRIM reduces the primary allocation in user-specified increments, down to a specified limit. If OS/390 cannot find the full primary allocation, SPACPRIM reduces the primary amount by a percentage specified by the user RLST and retries the allocation. The low allocation limit, below which SPACPRIM does not reduce the primary allocation, can be selectively applied to resources on the basis of data set name, job name, data or storage class, data set size or maximum possible size, and so on.

SPACPRIM can reduce the amount of primary space requested for a data set when there is insufficient space available on the volume to satisfy the request. This facility is controlled by the SPACPRIM function. The SPACPRIM function allows the user to specify:

- By what amount the primary space request is to be reduced

- The floor or limit to which SPACPRIM is allowed to reduce the primary space.

If SPACPRIM is used to reduce the primary space request for a partitioned data set or PDSE, only the primary space amount requested is changed. The number of directory blocks requested remains unchanged.

IGD17040I is issued for an SMS-managed data set even when SPACPRIM allows the allocation to succeed.

### SPACPRIM and NOCATLG2 under SMS

Note that recovery attempts are not necessarily done in the order the DD statements appear. SPACPRIM first attempts recovery on DD statements using specific requests (hardcoded volume requests). The nonspecific storage requests are the next candidates for recovery. And lastly, nonspecific public requests are recovered. The following job is an example of this condition.

---

```
//JOB ...
//STEP1 EXEC PGM=IEFBR14
//DD1 DD DSN=&&TEMP,DISP=(NEW,CATLG),
//DD2 DD DSN=EXAMPLE.DATASET2,DISP=(NEW,CATLG),
//      UNIT=SYSDA,SPACE=(CYL,(9999,10))
//DD3 DD DSN=EXAMPLE.DATASET1,DISP=(NEW,CATLG),
//      UNIT=SYSDA,SPACE=(CYL,(9999,10)),VOL=SER=TEMP99
//
```

---

In the example above, recovery would first be attempted on DD3, followed by DD2, then DD1. If recovery failed on DD1, messages concerning successful recovery for DD3 and DD2 would precede the recovery failed message for DD1.

### Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
SPACPRIM=( <i>nm</i> , <i>nm</i> )	Specifies the lower limit and decrement of space reduction, both as a percentage of the original primary allocation value.

## Rule List Parameter Explanations

### EVENTID=

**Purpose:** Specifies the identifier assigned to a user event in SMEVNT $xx$ . 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 SMEVNT $xx$ .

**Default:** None

### SPACPRIM=

**Purpose:** Specifies the lower limit and decrement of space reduction, both as a percentage of the original primary allocation value. The first value specifies a lower limit, below which SPACPRIM will not go. The second value is the percentage by which the primary allocation will be decremented. For example, if SPACPRIM=(50,15), the function will decrement the original primary value by 15% on each attempt to find a primary extent but will not decrement the primary size more than 50% of the original value.

**Syntax:** SPACPRIM=( $nn,nn$ )

where  $nn$  is a percentage in a range 0-90. The first value specified is the lower limit; the second value specified is the decrement amount.

**Default:** None

## Filter List Parameters Not Supported

The following filter list parameters are not supported for SPACPRIM:

ENVIR	FILESEQ	GDGVER
LABELTYP	VALUE	VCOMPLLQ

## Usage Notes

SPACPRIM attempts to allocate the data set by reducing the primary space in decrements of *nn%* (of the original primary space), until the low limit specified in the SPACPRIM parameter is reached. The first SPACPRIM parameter specifies the minimum acceptable space as a percentage of the original primary allocation; the second parameter specifies the decrement percentage. SPACPRIM does not attempt to allocate the data set with primary space less than the low limit.

The following JCL specifies a primary allocation of 110 tracks:

```
//TRANOUT1 DD . . . , SPACE=(TRK,(110,50)), . . .
```

In the SET SPACPRIM statement, the minimum acceptable space is 70% of the primary allocation.

```
SET SPACPRIM=(70,10)
```

In this example, the minimum is 77 tracks (0.70 x 110=77).

Allocation Attempt	Results/Action
First	OS/390 cannot find 110 tracks. SPACPRIM reduces the primary allocation by 10% to 99 tracks (110-11=99). Since this is not less than the low limit of 70% (77 tracks), SPACPRIM returns control to OS/390 for allocation processing.
Second	OS/390 cannot find 99 tracks. SPACPRIM reduces the primary allocation by 10% to 88 tracks (99-11=88). Since this is not less than the low limit of 70% (77 tracks), SPACPRIM returns control to OS/390 for allocation processing.
Third	OS/390 cannot find 88 tracks. SPACPRIM reduces the primary allocation by 10% to 77 tracks (88-11=77). Since this is not less than the low limit of 70% (77 tracks), SPACPRIM returns control to OS/390 for allocation processing.
Fourth	OS/390 cannot find 77 tracks. SPACPRIM reduces the primary allocation by 10% to 66 tracks (77-11=66). Since this is less than the low limit of 70% (77 tracks), SPACPRIM writes a message indicating that the space allocation could not be satisfied within the defined limits. The job fails with a JCL error.

### Processing Considerations

SPACPRIM performs primary space reduction processing for data sets only after all eligible pools have been searched for the full required space. The SPACPRIM function then begins the reduced space search with the first pool and proceeds through all eligible pools at each 10% reduction level.

SPACPRIM will not reduce the primary allocation space below the specified minimum. In conjunction with the DASDPOOL function, SPACPRIM will search all volumes in *all* eligible pools for a volume with sufficient space; however, if no volumes are found with the minimum required space, the allocation will fail.

The SPACPRIM parameter may be set to zero to allow a successful allocation even when the successively reduced primary allocation cannot be found. If SPACPRIM=0 is specified, the data set will allocate with zero space; when the data set is subsequently written to, the SPACVOLA function will add a volume to the data set. VSAM data sets cannot be allocated with an extent less than one CA in size so StopX37/II will not attempt to allocate a zero space VSAM primary.

**Note:** The MNTYPE keyword is not a reliable filter criterion for this function, since no volume has been selected. Also, the VOL keyword will not be filled in for a non-specific allocation.

Note that the SPACPRIM function must be specified in the SMFUNCxx member to be available and must be set active to provide service.

In general, extent-size during allocation/volume-addition can be controlled by several StopX37/II functions, as follows:

SPACSQTY adds or overrides requested space on new allocations.

SPACPRIM reduces the size of the primary extent if the requested value cannot be found.

SPACSWIR reduces the size of the first allocation on an added volume if the requested value cannot be found.

SPACVOLA adds a volume when OS/390 encounters an end-of-volume condition:

- when at least one of the eligible volumes contain an extent large enough to satisfy the requested size.

*OR*

- when SPACSWIR is active, and none of the eligible volumes contain an extent large enough to satisfy the requested size.

During the SPACPRIM function, the VOL parameter will not contain a volume serial number for non-specific allocation requests.

**VOL parameter and non-specific requests**

- Non-specific allocations are DD statements that do not contain a VOL=SER parameter or dynamic allocations that do not contain a DALVLSER text unit (unit key x'0010').
- The DASDPOOL function can convert a specific request to non-specific if you use the SUPVOL=YES parameter.
- A volume serial number is not assigned for non-specific requests until allocation has successfully completed. The SPACPRIM function is entered because there is no volume with enough space to allow allocation to complete the request.
- Non-SMS allocation requests that include VOL=SER are specific.
- SMS-managed requests that include VOL=SER must also use a storage class with the guaranteed space attribute; otherwise, DFSMS will strip the VOL=SER parameter from the request.

**Messages**

SPACPRIM issues one of the following messages:

SVM4117I REDUCED PRIMARY SPACE FROM *nnnn type* TO *nnnn*

SVM4100I PRIMARY NOT REDUCED FOR KEYSRANGE CLUSTER *dsn*

SVM4101I UNABLE TO REDUCE PRIMARY RC = *return code*, RSN = *reason code*

SVM4102I UNABLE TO REDUCE PRIMARY, REASON=NO SPACE FPL

SVM4103I PRIMARY SPACE REQUEST WENT TO ZERO DSN=*dsn*

SVM4104I PRIMARY SPACE REQUEST WENT BELOW LIMIT DSN=*dsn*

SVM4114I PRIMARY SPACE REQUEST WENT BELOW LIMIT

SVM4115I PRIMARY SPACE REQUEST WENT TO ZERO

## Example

This example shows you how to

- allow primary space reduction on all data sets except those with a primary allocation size of less than 50 KB
- tailor the reduction limit to primary allocation size, maximum possible size, secondary allocation, and pool

SMFUNCF3 member

```
SET  NAME=SPACPRIM ACTIVE=YES      The function is defined and activated.
      MSG=I   SMF=I
      FLST=F6 RLST=F6
      DESC='REDUCE PRIMARY SPACE'
```

**SMFLSTF6** member

```
SET  MODE=ACT                      Select all data sets, except those with a primary allocation less
      EXC   SIZE<50KB              than 50 kilobytes.
      INC   DSN=/'
```

SMRLSTF6 member

```
SET  SPACPRIM=90                   For data sets without a secondary allocation, allow a space
      INC   SECSpace=0KB           reduction of only 10%.

SET  SPACPRIM=60                   For sequential data sets with a maximum size (primary plus 15
      INC   MAXSIZE>100MB         secondaries) greater than 100 megabytes, allow primary
      DSORG=PS                    space reduction down to 60% of the original space.

SET  SPACPRIM=70                   For VSAM data sets with a maximum size (primary plus 122
      INC   MAXSIZE>300MB         secondaries) greater than 300 megabytes, allow primary
      DSORG=VS                    space reduction down to 70% of the original space.

SET  SPACPRIM=(50,15)             For data sets with a primary allocation between 10 and 100
      INC   SIZE<100MB SIZE>10MB megabytes, allow a reduction to 50% of the original space, in
                                   decrements of 15%.

SET  SPACPRIM=80                   For data sets with a primary allocation between 100 and 300
      INC   SIZE<300MB           megabytes, allow a reduction to 80% of the original space.
      INC   SIZE>100MB

SET  SPACPRIM=90                   For data sets allocated in PROD12 or PROD13 pools, allow a
      INC                                   reduction only to 90%.
      POOL=(PROD12,PROD13)

SET  SPACPRIM=(40,20)             For all other data sets, allow a reduction down to 40% of the
      INC   DSN=/'                 original space in decrements of 20%.
```

## SPACRLSE - Release Data Set Space at Close

The SPACRLSE function causes unused space in physical-sequential data sets to be released when the data set is closed.

Inaccurate space estimates can allocate significantly more space in data sets than is actually needed. Excessive space requests are frequently made when the expected data set size is not known, particularly to avoid abending with a space related problem. While OS/390 provides a facility to release unused space from a new data set, it must be requested using the RLSE subparameter of the SPACE parameter in JCL. If RLSE is not specified in the JCL, the extra space in the data set is not freed until the data set is eventually deleted, or until the storage administrator manually searches for and releases that space.

Excessive space allocations unnecessarily consume significant DASD space, causing other jobs to fail for lack of space and requiring the purchase of extra DASD devices.

The SPACRLSE function allows automated release of unused space, regardless of JCL specifications.

### Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
RLSE=ALL/SEC	Specifies the release of space for all data sets, or only for those with a secondary allocation

### Rule List 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

### **RLSE=**

**Purpose:** Specifies whether space is released for all selected data sets, or only those selected data sets which have secondary allocations. This is a required parameter.

**Syntax:** `RLSE=ALL/SEC/NO`

ALL	Releases space for all data sets
SEC	Releases space for data sets with a secondary allocation
NO	Turns the release flag off

Default: None

### **Filter List Parameters Not Supported**

The following filter list parameters are not supported for SPACRLSE:

- GDGVER
- VALUE

## **Usage Notes**

SPACRLSE provides services for DASD physical sequential data sets. Unused space is released from the data set when the data set is closed. SPACRLSE does not release the space itself, but sets the RLSE JCL parameter for selected resources.

SPACRLSE works for both new allocations and existing data sets.

The RLSE parameter specifies the scope of the function. *ALL* specifies that all selected data sets should be flagged for space release. *SEC* specifies that only data sets with a secondary allocation should be flagged for release; data sets with only a primary allocation are ignored. This allows the primary allocation to remain at full size, which may be desirable if the data set is extended at a later time.

The RLSE parameter must be specified; SPACRLSE takes no action without a rule specifying the release scope for selected resources.

SPACRLSE does not modify the action of OS/390 in processing the release of unused data set space; therefore, OS/390 restrictions still apply. See the discussion of the RLSE subparameter of the SPACE parameter in the *MVS JCL Reference Manual* for your release of OS/390.

Note that the SPACRLSE function must be specified in the SMFUNCxx member to be available and must be set active to provide services.

### Messages

The message issued by SPACRLSE is:

```
SVM3349I  job,step,dd,dsn RELEASE PARAMETER ASSIGNED
```

### Example

This example shows you how to release space for all data sets except certain temporary work files that are frequently extended during the course of a job. For production jobs, do not release space in data sets with only a primary allocation.

#### SMFUNCD5 member

```
SET  NAME=SPACRLSE ACTIVE=YES      The function is defined and
      MSG=I SMF=I                   activated.
      FLST=D3 RLST=D3
      DESC='FREE UNUSED SPACE'
```

#### SMVARSD0 member

```
SET  VARIABLE=DLYWORK              Define a variable which identifies all
      VALUE=**.DLYWORK*             transitory work files that should not
      VALUE=**.TRAN*/              have free space deleted.
      VALUE=**.TEMP*
      VALUE=WORK/
```

#### SMFLSTD3 member

```
SET  MODE=ACT                      Select all data sets for SPACRLSE
      EXC DSN=&DLYWORK              processing, except those specified
      INC DSN=/                     in the variable &DLYWORK.
```

#### SMRLSTD3 member

```
SET  RLSE=SEC                      For production jobs, release space
      INC JOBCLASS=P              only if the data set has a secondary
                                  allocation.

SET  RLSE=ALL                      For all other data sets, release
      INC DSN=/                   space.
```

# SPACSECA - Add Secondary Allocation Quantity

The SPACSECA function provides a secondary allocation for data sets that were created with only a primary allocation.

A data set created with only a primary allocation is not extended by OS/390 when the primary allocation is full. The job abends with a D37 system code. SPACSECA adds a secondary allocation quantity to the data set, based on a user-defined percentage of the primary allocation quantity. The data set can then be extended with up to 15 extents for non-VSAM data sets, or 122 extents for VSAM data sets.

## Global Parameter

The global parameter that affects SPACSECA is listed below.

Parameter	Purpose
VSAMZSEC	Controls out-of space recoveries for VSAM files with zero secondary space coded.

## Rule List Parameters

SET Statement (in member SMRLST $xx$ )

Parameter	Description
EVENTID= $xxxxx$	Identifies a user event defined in an SMEVNT $xx$ member
NOCHECK= $xxxxxx$	Specifies the checks bypassed in volume add validation
SPACSECA= $nnnn$	Specifies the amount of secondary space to allocate

## Rule List Parameter Explanations

### EVENTID=

**Purpose:** Specifies the identifier assigned to a user event in SMEVNT $xx$ . 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 SMEVNT $xx$ .

Default: None

**NOCHECK=**

Purpose: Specifies the checks bypassed in space recovery validation.

Syntax: NOCHECK=CONTIG

Default: None

**SPACSECA=**

Purpose: Specifies the amount of secondary space to allocate. This is a percentage of the primary allocation amount, and may be greater than 100%. This is a required parameter.

Syntax: SPACSECA=*nnnn*

where *nnnn* is a percentage in the range 1-9999.

Default: None

**Filter List Parameters Not Supported**

The following filter list parameters are not supported for SPACSECA:

ENVIR	FILESEQ	GDGVER
LABELTYP	VALUE	VCOMPLLQ

## Usage Notes

SPACSECA adds a secondary allocation amount to data sets which were created without a secondary allocation and which are out of space on the primary extent.

SPACSECA provides services for both VSAM and non-VSAM data sets, new or existing.

SPACSECA calculates a secondary space amount based on the value specified in the SPACSECA action parameter. This parameter specifies a percentage of the primary allocation and may be greater than 100%. SPACSECA multiplies the primary amount by the SPACSECA value to compute the new secondary amount.

The minimum secondary allocation size used by SPACSECA for VSAM data sets is the control area size.

The SPACSQTY function also provides the capability of specifying a secondary allocation for a non-VSAM data set. SPACSQTY operates during a data set's allocation and also allows setting the primary allocation quantity.

The VSAMZSEC global parameter determines which function controls recoveries for VSAM data sets that do not have a secondary allocation amount coded. If VSAMZSEC=YES is specified, the SPACSECA determines whether the recovery will be performed. If VSAMZSEC=NO is specified, SPACVOLA determines whether the recovery will be performed.

Note that the SPACSECA function must be specified in the SMFUNCxx member to be available and must be set active to provide service.

### Messages

SPACSECA issues message SVM4855I when it adds a secondary allocation to a data set:

```
SVM4855I  SET SECONDARY SPACE USING nn% of primary
```

**FOR DATA COMPONENT** or **FOR INDEX COMPONENT** is appended to the message for VSAM data sets.

### Example

This example shows you how to add a secondary allocation of at least 50% of the primary for all data sets that do not have a secondary and are out of space

- for VSAM data sets with a primary allocation larger than 50 megabytes, use a secondary of 28%
- for sequential transaction file data sets, use a secondary of 250%

#### SMFUNC4B member

```
SET  NAME=SPACSECA ACTIVE=YES           The function is defined and activated.
     MSG=I SMF=I
     FLST=4B RLST=4C
     DESC='ADD SECONDARY
     ALLOCATION'
```

#### SMFLST4B member

```
SET  MODE=ACT                          Select all data sets.
     INC DSN=/'
```

#### SMRLST4C member

SET	SPACSECA=28 INC DSORG=VS SIZE>50MB	Add a secondary allocation of 28% of the primary to VSAM data sets larger than 50 megabytes without a secondary allocation that are out of space.
SET	SPACSECA=250 INC DSORG=PS LLQ=TRAN*	Add a secondary allocation of 250% of the primary to sequential transaction files.
SET	SPACSECA=50 INC DSN=/ 	Add a secondary allocation of 50% of the primary to all other data sets.

# SPACSECB - Reduce Secondary Allocation to Best Fit

The SPACSECB function reduces the size of the requested secondary allocation to the size of the largest single extent on the volume, even if the request could be satisfied within five extents.

OS/390 satisfies a secondary extent by allocating up to five individual extents to accumulate the required space. This has two undesirable effects:

- The space on the volume can become more fragmented, as a single large extent may be split to provide the second through fifth extents of the secondary allocation; and
- Five extents of the maximum allowed for a data set (16 for non-VSAM data sets) are used instead of 1, reducing the maximum possible expansion of the data set.

SPACSECB changes the secondary allocation amount to fit in the largest extent on the volume, before OS/390 tries to allocate the extent. This generally provides a net increase in total space allocation over OS/390 five-extent secondary processing.

## Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
SPACSECB=nnn	Specifies the lower limit percentage for space reduction

## Rule List 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 SMEVNT*xx*.

Default: None

### **SPACSECB=**

**Purpose:** Specifies the lower limit for the space reduction as a percentage of the original secondary allocation request. SPACSECB will not reduce the allocation request below this limit.

**Syntax:** SPACSECB=*nnn*

where *nnn* is a percentage in the range of 0-100.

Default: SPACSECB=0

### **Filter List Parameters Not Supported**

The following filter list parameters are not supported for SPACSECA:

ENVIR	FILESEQ	GDGVER
LABELTYP	VALUE	VCOMPLLQ

## **Usage Notes**

SPACSECB reduces the requested secondary allocation amount to the size of the largest available single extent on the volume (the best fit). If the secondary extent is less than the largest available single extent, normal allocation processing takes place without any action by SPACSECB. If the largest available single extent is smaller than the lower limit specified in the SPACSECB parameter, the reduction will not take place.

Both SPACSECB and SPACSECR provide similar processing; however, SPACSECB receives control before OS/390 attempts to allocate the secondary extent, while SPACSECR receives control after OS/390 attempts (and fails) to allocate the secondary extent. SPACSECB is more effective at preventing volume fragmentation, but will allocate a smaller single secondary extent than the OS/390 allocation of up to five extents. SPACSECR provides secondary space reduction only when OS/390 cannot satisfy the secondary allocation within five extents.

Because SPACSECB allocates the largest extent available on the volume, it should always be used with SPACRLSE, to ensure that unneeded space is released.

SPACSECB never increases the size of the secondary allocation to match the largest available extent.

SPACSECB provides services for both VSAM and non-VSAM data sets, new or existing. SPACSECB does not operate on VIO data sets, VSAM KSDS indexes, or VSAM KSDSs with key ranges.

The minimum secondary allocation selected by SPACSECB for VSAM data sets is the control area size.

Note that the SPACSECB function must be specified in the SMFUNCxx member to be available and must be set active to provide service.

### Messages

SPACSECB issues message SVM4865I when the function reduces a secondary allocation for a data set:

```
SVM4865I  CHANGED SPACE FROM nnnn units TO nnnn
```

### Example

Use best fit secondary reduction processing on pools and volumes that are prone to high fragmentation.

#### SMFUNC06 member

```
SET  NAME=SPACSECB ACTIVE=YES           The function is defined and
      MSG=I   SMF=I                       activated.
      FLST=10
      DESC='BEST FIT SECONDARY ALLOC'
```

#### SMFLST10 member

```
SET  MODE=ACT
      INC   POOL=(PRODA2,PUBL01)         Select certain pools and
      INC   POOL=WORK/                   volumes for best fit secondary
      INC   VOL=CICS03                    processing.
```

#### SMRLST10 member

```
SET  SPACSECB=0                           Set no lower limit for all jobs in
      INC   JOBCLASS=C                     class C.
```

## SPACSECI - Increase Secondary Allocation Quantity

The SPACSECI function increases the secondary space allocation quantity as the file is being extended.

The secondary allocation quantity is specified for a file to provide expansion space if the primary quantity is insufficient. Primary and secondary quantities are usually specified to handle normal conditions; this prevents over allocating space on the installation's disk drives. However, during periods of unusually heavy activity, the secondary allocation may not be sufficient to allow the file to hold the extra data. The job abends with a B37 or E37 system code.

SPACSECI increases the secondary allocation quantity of a data set at a user-defined point during secondary extent processing. The secondary quantity increase is progressive; each successive extent is larger than the preceding extent. This process adds increasingly larger secondary extents to a data set and can prevent job abends due to out-of-space conditions.

### Rule List Parameters

SET Statement (in member SMRLST $_{xx}$ )

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNT $_{xx}$ member
PCTI=nnnnn	Specifies the percentage value by which a secondary allocation is increased
SPACSECI=nn	Specifies the extent after which the secondary allocation quantity will be increased

### Rule List Parameter Explanations

#### EVENTID=

**Purpose:** Specifies the identifier assigned to a user event in SMEVNT $_{xx}$ . 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 SMEVNT $_{xx}$ .

Default: None

### **PCTI=**

Purpose: Specifies the percentage value by which a secondary allocation is increased.

Syntax: PCTI=*nnnnn*

where *nnnnn* is a number between 0 and 10000.

Default: None

### **SPACSECI=**

Purpose: Specifies the last extent to receive the original secondary space allocation quantity; after this extent, the secondary allocation is increased. This number is relative to the overall extent count; that is, the primary extent is number 1, the first secondary extent is number 2, and so on. SPACSECI=4 specifies that the increase in secondary allocation quantity will begin *after* the fourth extent of the data set. This is a required parameter.

Syntax: SPACSECI=*nn*

where *nn* is a number in the range 1-15.

Default: None

### **Filter List Parameters Not Supported**

The following filter list parameters are not supported for SPACSECI:

ENVIR	FILESEQ	GDGVER
LABELTYP	VALUE	VCOMPLLQ

## **Usage Notes**

SPACSECI applies to physical sequential and partitioned data sets that are being extended with a secondary allocation.

SPACSECI increases the secondary allocation amount by adding the original secondary allocation specification from the JCL to the previous value of the secondary allocation. The secondary allocation quantities, therefore, increment by the original secondary quantity.

The SPACSECI action parameter specifies where the increase in secondary allocation occurs. The parameter value is the extent *after* which the secondary amount will be increased; it is the last extent to receive the original secondary allocation quantity. The SPACSECI parameter is relative to the overall extent count; SPACSECI=1 specifies that processing will start after the primary extent, SPACSECI=14 specifies that processing will start after the fourteenth extent, and so on. If OS/390 satisfied the primary extent with multiple extents, those extents apply to the overall extent count.

In addition to increasing the secondary allocation amount for the next extent, the function also flags the data set for the release of unused space (as with the JCL parameter RLSE).

Note that the SPACSECI function must be specified in the SMFUNCxx member to be available and must be set active to provide service.

### PCTI Parameter

When specified with the SPACSECI function, the SPACSECI parameter value is ignored. The original secondary is increased by the PCTI value whenever the data set extends on the same volume. This does not apply to the first extent on the volume because SPACSECI does not get control after a volume add. The first extent will be equal to the original secondary; however, extents 2 - 16 will receive the adjusted value.

The NVOL selection parameter can be used to control the secondary increases after a volume add occurs.

### Example

SET SPACSECI=10 PCTI=200 INC DSN= / NVOL > 1	Increase the secondary allocation by 200 percent of the original secondary on all volumes except for the first.
---	---

### SPACSECI Processing Example

A data set is allocated with SPACE=(TRK,(1,1)); the primary extent is one track, the secondary extent is two tracks.

The SPACSECI rule list sets the extent trigger at the second extent:

```
SET SPACSECI=2
INC DSN= /
```

Secondary allocation quantity modification occurs *after* the second extent is allocated (on the third extent).

The output of SPACSECI shows the increase in the secondary allocation quantity.

SVM4000I ATTEMPTING SPACSECI FOR EMPCRM.SPACSECI.TEST2 1,  
SYSUT2, WORK01, 3

SVM4874I INCREASED SPACE FROM 1 TRKS TO 2

SVM4000I ATTEMPTING SPACSECI FOR EMPCRM.SPACSECI.TEST2 1,  
SYSUT2, WORK01, 4

SVM4874I INCREASED SPACE FROM 1 TRKS TO 3

SVM4000I ATTEMPTING SPACSECI FOR EMPCRM.SPACSECI.TEST2 1,  
SYSUT2, WORK01, 5

SVM4874I INCREASED SPACE FROM 1 TRKS TO 4

SVM4000I ATTEMPTING SPACSECI FOR EMPCRM.SPACSECI.TEST2 1,  
SYSUT2, WORK01, 6

SVM4874I INCREASED SPACE FROM 1 TRKS TO 5

SVM4000I ATTEMPTING SPACSECI FOR EMPCRM.SPACSECI.TEST2 1,  
SYSUT2, WORK01, 7

SVM4874I INCREASED SPACE FROM 1 TRKS TO 6

SVM4000I ATTEMPTING SPACSECI FOR EMPCRM.SPACSECI.TEST2 1,  
SYSUT2, WORK01, 8

SVM4874I INCREASED SPACE FROM 1 TRKS TO 7

SVM4000I ATTEMPTING SPACSECI FOR EMPCRM.SPACSECI.TEST2 1,  
SYSUT2, WORK01, 9

SVM4874I INCREASED SPACE FROM 1 TRKS TO 8

Note that each successive secondary allocation is the total of the preceding secondary quantity plus the original secondary quantity.

The new secondary extent size is temporary; it exists only for the duration of the current job step. The original secondary allocation quantity associated with the data set is not modified.

If the current secondary allocation quantity (before incrementing) is greater than or equal to the size of the single largest available extent on the volume, the secondary quantity will not be modified. If, after incrementing, the secondary quantity is larger than the single largest available extent on the volume, the size of the largest extent will be used as the secondary quantity.

The maximum expansion for a sequential data set with normal OS/390 secondary extent processing is 15 times the size of the secondary extent. With SPACSECI, the maximum expansion is 135 times the size of the secondary extent.

### Messages

SPACSECI issues message SVM4874I when it increases a secondary allocation:

```
SVM4874I  INCREASED SPACE FROM nnnn type TO nnnn
```

### Example

This example shows you how to enable secondary allocation increases for production data sets in end-of-quarter processing.

#### SMFUNC09 member

```
SET  NAME=SPACSECI ACTIVE=YES           The function is defined and
      MSG=I   SMF=I                       activated.
      FLST=03 RLST=S3
      DESC='INC SECONDARY ALLOCATION'
```

#### SMFLST03 member

```
SET  MODE=ACT
      INC   JOBACCT2=PROD LLQ=QE*        Select all production data sets
                                          whose last name qualifier
                                          begins with QE (quarter-end).
```

#### SMRLSTS3 member

```
SET  SPACSECI=8
      INC   DSN4=MSTR*                  Begin increasing the
                                          secondary allocation after the
                                          8th extent for master files.

SET  SPACSECI=5
      INC   DSN4=TRAN*                  Begin increasing the
                                          secondary allocation after the
                                          5th extent for transaction files.

SET  SPACSECI=10
      INC   DSN=/                        Begin increasing the
                                          secondary allocation after the
                                          10th extent for all other files
                                          selected by the filter list
                                          above.
```

## SPACSECR - Reduce Secondary Allocation to Largest Extent

The SPACSECR function reduces the size of the requested secondary allocation to the size of the largest single extent on the volume, after OS/390 has been unsuccessful at allocating the required space.

OS/390 satisfies a secondary extent by allocating up to five individual extents to accumulate the required space. If OS/390 cannot accumulate the total secondary allocation amount in five extents on the volume, the job abends.

SPACSECR decreases the secondary allocation amount to fit in the largest single extent on the volume, after OS/390 has failed to allocate the extent. Since OS/390 cancels the job on the allocation failure, SPACSECR provides a chance for the job to successfully complete.

### Global Parameters

The global parameters that affect the SPACSECR function are described in the following table.

Parameter	Purpose
MREDUCE	Determines if secondary space reduction can occur on multi-volume data sets allocated by JCL
VSAMLIMWARN	Specifies the percentage value to be used before issuing the 4GB-limit message

### Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
NOCHECK=xxxxxx	Specifies the checks bypassed in volume add validation
SPACSECR=( <i>nnn,nnn</i> )	Specifies a floor limit and percentage for space reduction

**Rule List 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

**NOCHECK=**

**Purpose:** Specifies the checks bypassed in space recovery validation.

**Syntax:** NOCHECK=CONTIG

**Default:** None

**SPACSECR=**

**Purpose:** Specifies the lower limit and the decrement of space reduction, both as a percentage of the original secondary allocation value. The first value specifies a lower limit, below which SPACSECR will not go. The second value is the percentage by which the secondary allocation will be decremented. The decrement value is only used for multi-stripe data sets; for all other data sets, the new secondary value is set to the value of the largest contiguous extent on the volume. For example, if SPACSECR=(50,10), the function will decrement the original secondary value by 10% on each attempt to find a secondary extent but will not decrement the secondary size more than 50% of the original value. The decrement percentage is used only for striped data sets with multiple stripes.

**Syntax:** SPACSECR=(*nnn*,*nnn*)

where the first *nnn* is the floor limit and the second *nnn* is a percentage from 0-100 by which reduction can take place until either it fits or the floor limit is reached. (A specification of 100 will not reduce the secondary size at all.)

Default: SPACSECR=(0,10)

### Filter List Parameters Not Supported

The following filter list parameters are not supported for SPACSECR:

ENVIR	FILESEQ	GDGVER
LABELTYP	VALUE	VCOMPLLQ

## Usage Notes

The first SPACSECR parameter, the floor value, specifies the minimum percentage of the original secondary request that will be allowed. SPACSECR will not attempt to allocate an extent that is below the specified percentage. For example, if SPACSECR=(40,10) the function will reject the allocation if the largest single extent on the volume is less than 40% of the data set's secondary extent.

The second SPACSECR subparameter specifies the decrement percentage that is applied only when a striped data set containing multiple stripes cannot find a requested secondary extent. In this case, the secondary extent will be decremented successively by the decrement value, starting with the full secondary value, until an extent is found or the floor value is reached. A percentage is used for reduction instead of best fit since the new secondary value must fit on all of the volumes the multi-stripe data set exists on.

If SPACSECR=100, no secondary reduction is applied.

SPACSECR processes both VSAM and non-VSAM data sets, new or existing. SPACSECR does not process VIO data sets, DB2® data sets, VSAM KSDS indexes, or VSAM KSDSs with key ranges.

The SPACSECR floor parameter sets a low limit to the space reduction, as a percentage of the original secondary extent size. The largest extent on the volume must be at least as large as the SPACSECR percentage of the original secondary allocation. If the largest extent is less than this, the allocation fails.

**Example**

SPACSECR = 55  
secondary = 100 tracks  
lower limit = 55% of 100 = 55 tracks

If the largest extent on the pack is 55 tracks or larger, that extent will be used. However, if the largest extent on the volume is less than 55 tracks, the allocation fails.

The minimum secondary allocation selected by SPACSECR for VSAM data sets is the control area size.

Both SPACSECR and SPACSECB provide similar processing; however, SPACSECB receives control before OS/390 attempts to allocate the secondary extent, while SPACSECR receives control after OS/390 attempts (and fails) to allocate the secondary extent. SPACSECB is more effective at preventing volume fragmentation, but allocates a smaller single secondary extent than the OS/390 allocation of up to five extents. SPACSECR provides secondary space reduction only when OS/390 cannot satisfy the secondary allocation within five extents.

**MREDUCE global parameter**

The MREDUCE global parameter affects SPACSECR processing for data sets that are allocated to multiple volumes through JCL. If a data set is allocated to multiple volumes using the VOL=SER or the UNIT JCL parameter and MREDUCE=NO is set, SPACSECR processing will not occur until the data set is on the last volume. MREDUCE=YES will allow SPACSECR to process on all volumes.

Note that the SPACSECR function must be specified in the SMFUNC $_{xx}$  member to be available and must be set active to provide service.

In general, extent-size during allocation/volume-addition can be controlled by several StopX37/II functions, as follows:

SPACSQTY adds or overrides requested space on new allocations.

SPACPRIM reduces the size of the primary extent if the requested value cannot be found.

SPACSWIR reduces the size of the first allocation on an added volume if the requested value cannot be found.

SPACVOLA adds a volume when OS/390 encounters an end-of-volume condition:

- when at least one of the eligible volumes contain an extent large enough to satisfy the requested size.

*OR*

- when SPACSWIR is active, and none of the eligible volumes contain an extent large enough to satisfy the requested size.

### Messages

SPACSECR issues message SVM4801I when the function successfully reduces a secondary allocation for a data set:

SVM4801I REDUCED SPACE FROM *orga units* TO *newa*

### Example

This example shows you how to reduce the allocation for production jobs if there is no volume with enough space.

#### SMFUNC09 member

```
SET  NAME=SPACSECR  ACTIVE=YES      The function is defined and activated.
      MSG=I  SMF=I
      FLST=04  RLST=04
      DESC='REDUCE SECONDARY
      ALLOC'
```

#### SMFLST04 member

```
SET  MODE=ACT
      INC    JOBACCT2=PROD*          Select only production jobs for reduced secondary
                                     processing.
```

#### SMRLST04 member

```
SET  SPACSECR=60
      INC    JOBCLASS=(A,F,S)       Set a lower limit for space reduction of 60% for all jobs in
                                     classes A, F, and S.

SET  SPACSECR=(30,20)
      INC    JOBCLASS=(K,U)        Set a lower limit for space reduction of 30% for all jobs in
                                     classes K and U and set a decrement of 20% for striped
                                     data sets.

SET  SPACSECR=0
      INC    JOBCLASS=C           Set no lower limit for space reduction for all jobs in class C.
```

## SPACSQTY - Set Primary and Secondary Space for Data Set

The SPACSQTY function sets primary and secondary space allocation values for non-VSAM and VSAM DASD data sets being created. Exceptions for VSAM allocations are noted in the parameters explanations.

Primary and secondary space allocation values have a significant role in effective use of DASD space. Allocations which are consistently too high, even for temporary data sets, utilize more space than is necessary, possibly preventing other jobs from allocating needed resources. Allocations which are too low require subsequent processing of some type, either automated or manual, to allocate more space to the data set.

SPACSQTY allows the installation to set requirements and limitations on data set size and to enforce those requirements rigorously. Application of predefined primary and secondary space allocations can be based on data set names, on device types, on assigned pools, on size or maximum possible size, or on any property allowed as a MAINVIEW SRM selection parameter.

### Global Parameter

The global parameter that affects SPACSQTY is described in the table below.

Parameter	Purpose
X37POOL	Specifies which volume is used by X37 to determine the POOL name in EOVS processing

### Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
ALCTYPE=xxx	Quantity unit of space allocation (TRK, CYL, KAV, MAV, UAV, BLK, KB, MB)
AVL=nnnnn	Average block or record length
CONTIG=YES/NO	Specifies whether a data set is allocated with contiguous space required
DIR=nnnn	Sets number of directory blocks for partitioned data sets
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
PQTY=nnnnn	Primary space allocation, in unit specified
REPLACE=YES/NO	Allows override to JCL-specified space allocations

Parameter	Description
ROUND=YES/NO	Rounds space allocated in blocks to nearest cylinder
SQTY=nnnnn	Secondary space allocation, in unit specified

### Rule List Parameter Explanations

#### ALCTYPE=

**Purpose** Specifies the unit of space allocation.

**Syntax:** ALCTYPE=*xxx*

where *xxx* is a valid allocation type from the following list:

KB	A block allocation where PQTY and SQTY are assumed to be kilobytes
MB	A block allocation where PQTY and SQTY are assumed to be megabytes
TRK	Tracks
CYL	Cylinders
KAV	AVGREC=K
MAV	AVGREC=M
UAV	AVGREC=U
BLK	Blocks

**Note:** KAV, MAV, and UAV are not valid parameters for VSAM allocations.

**Default:** KB

#### AVL=

**Purpose** Specifies the average block size for BLK allocations and record length for KAV, MAV, and UAV allocations.

**Syntax:** AVL=*nnnnn*

where *nnnnn* is a number in the range 1-32767

**Default:** 9000

**CONTIG=**

**Purpose:** Specifies whether a data set is allocated with contiguous space required.

**Syntax:** CONTIG=*YES/NO*

**Default:** None

**DIR=**

**Purpose** Specifies the number of directory blocks for partitioned data sets.

**Note:** This parameter can only be used to change the directory blocks. The user must request DSORG=PO or directory blocks in the original allocation.

**Syntax:** DIR=*nnnnn*

where *nnnn* is a number between 1 and 32767

**Default:** None

**EVENTID=**

**Purpose:** Specifies the identifier assigned to a user event in SMEVNT*xx*. 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 SMEVNT*xx*.

**Default:** None

**PQTY=**

**Purpose:** Specifies the primary space allocation quantity in the unit specification ALCTYPE for a new non-VSAM data set.

This is a required parameter, along with SQTY.

**Syntax:** PQTY=*nnnnn*KB/MB

where *nnnnn* is a number in the range 1–99999. An equivalent value can be expressed in megabytes with the suffix MB.

Default: None

### **REPLACE=**

**Purpose:** Specifies whether the space allocations already specified for the data set (in a JCL, for example) are overridden. If REPLACE=NO, and a primary or a secondary space allocation already exists for the new data set, then the SPACSQTY function does not replace either space allocation value. If REPLACE=YES, the SPACSQTY function sets the primary and secondary space allocations, regardless of any existing space parameters.

**Syntax:** REPLACE=YES/NO

**Default:** REPLACE=NO

**Note:** REPLACE only applies to PQTY and SQTY parameters. CONTIG and ROUND are not affected by REPLACE.

### **ROUND=**

**Purpose:** Specifies that the space allocation be rounded to the nearest cylinder for new data sets.

Note that this rounding process is performed by OS/390; MAINVIEW SRM merely switches on the appropriate bit to allow OS/390 to perform this service.

**Syntax:** ROUND=YES/NO

**Default:** None

**Note:** Both SPACCONV and SPACSQTY functions allow the ROUND parameter to be specified. If both are specified for the same data set, the value in SPACCONV will be the value assigned to the data set, because SPACCONV receives control after SPACSQTY.

**SQTY=**

**Purpose:** Specifies the secondary space allocation quantity in the units specified in ALCTYPE for a new non-VSAM data set.

This is a required parameter, along with PQTY.

**Syntax:** SQTY=*nnnnn*KB/MB

where *nnnnn* is a number in the range 1–99999. An equivalent value can be expressed in megabytes with the suffix MB.

**Default:** None

**Filter List Parameters Not Supported**

The following filter list parameters are not supported for SPACSQTY:

- GDGVER
- VALUE

## Usage Notes

SPACSQTY sets the data set's primary and secondary space allocation in the units specified in the ALCTYPE parameter.

**Note:** For compatibility with earlier releases of StopX37/II, PQTY and SQTY are assumed to be in kilobytes if ALCTYPE is not coded. The resulting allocation will be calculated in average blocks based on the AVL parameter.

SPACSQTY provides services for VSAM and non-VSAM DASD data sets being created; old data sets are not affected by SPACSQTY

Both PQTY and SQTY parameters must be specified; if either is omitted, the SPACSQTY function takes no action, regardless of other parameters.

The control parameters (ALCTYPE, AVL, CONTIG, DIR, ROUND) allow full control over space allocation specifications. ALCTYPE specifies the unit of space that is referred to by PQTY and SQTY; for example, PQTY=10 ALCTYPE=CYL requests 10 cylinders of space.

**REPLACE parameter**

The REPLACE parameter allows SPACSQTY to be applied in either a restrictive or a lenient manner. The REPLACE parameter specifies whether a previously-specified space allocation is to be overridden by the SPACSQTY PQTY and SQTY parameters.

- REPLACE=YES forces the use of the space allocation established in SPACSQTY, regardless of any space parameters already specified in the JCL or elsewhere.
- REPLACE=NO accepts the original space parameters. The space allocation set by SPACSQTY is applied only if there is no existing space allocation specification.

The SPACSECA function also allows specification of secondary allocation quantity. SPACSECA, however, is invoked when a data set is out-of-space on its primary extent.

**Note:** The ISPF data set information display in functions 3.4 and 3.2 do not always give the correct values for secondary space in blocks, particularly when the data set is allocated without a blocksize. The information provided by IEHLIST is correct, however. Note that the SPACSQTY function must be specified in the SMFUNCxx member to be available and must be set active to provide services.

In general, extent-size during allocation/volume-addition can be controlled by several StopX37/II functions, as follows:

SPACSQTY adds or overrides requested space on new allocations.

SPACPRIM reduces the size of the primary extent if the requested value cannot be found.

SPACSWIR reduces the size of the first allocation on an added volume if the requested value cannot be found.

SPACVOLA adds a volume when OS/390 encounters an end-of-volume condition:

- when at least one of the eligible volumes contain an extent large enough to satisfy the requested size

*OR*

- when SPACSWIR is active, and none of the eligible volumes contain an extent large enough to satisfy the requested size.

## Messages

SPACSQTY generates message SVM3446I when it provides new space allocation values for a data set:

```
SVM3446I  job,step,dd,dsn SPACE SET TO: TYPE=alloc_type,
          PRIMARY=prim_alloc, SECONDARY=sec_alloc, DIRECTORY
          BLOCKS=dir_blocks, AVL=avl_length.
```

Messages issued by SPACSQTY include:

```
SVM3317I  job,step,dd,dsn CONTIG y/n ASSIGNED
SVM3318I  job,step,dd,dsn ROUND y/n ASSIGNED
```

## Examples

The following example shows you how to force a primary and secondary space allocation limit on all new data sets being allocated on volumes starting with P and whose data set name ends with TEST, and with a normal termination disposition of DELETE.

### SMFUNCxx member

```
SET  NAME=SPACSQTY ACTIVE=YES
      MSG=I SMF=E
      FLST=JV RLST=JV
      DESC='SET PRIM AND SEC SPACE QTY'
```

The function is defined and activated.

### SMFLSTJV member

```
SET  MODE=ACT
      INC LLQ=TEST/VOL=P/
```

This filter list specification is active. It selects all data sets whose final qualifier begins with TEST, and which are being allocated to any volume whose name begins with P.

### SMRLSTJV member

```
SET  PQTY=100 SQTY=20KB REPLACE=YES
      ALCTYPE=KB
      INC DISP2=DELETE
```

All data sets selected by the FLST specification above and with a normal termination disposition of DELETE are given a primary and secondary space allocation as shown. Any space allocation parameters in the JCL are overridden.

The following example shows you how to impose space limitations determined by the fourth account parameter of the job step EXEC card for all data sets going on a public work device (POOL=PUBWRK), except for those jobs with a third job account parameter beginning with PR. Apply this limitation in simulation mode only, to determine impact.

**SMFUNCxx member**

```
SET  NAME=SPACSQTY ACTIVE=YES
      MSG=I SMF=I
      FLST=23 RLST=22
      DESC='SET PRIM AND SEC SPACE QTY'
```

The function is defined and activated.

**SMFLST23 member**

```
SET  MODE=SIM
      INC POOL=PUBWRK
      EXC JOBACCT3=PR/
```

This filter list specification is in simulation mode only. It selects all data sets in pool PUBWRK.

**SMRLST22 member**

```
SET  PQTY=1000 SQTY=500 REPLACE=YES ALCTYPE=KB
      INC STEPACCT3=LRG
SET  PQTY=500 SQTY=100 REPLACE=YES ALCTYPE=KB
      INC STEPACCT3=MOD
SET  PQTY=100 SQTY=25 REPLACE=YES ALCTYPE=KB
      INC STEPACCT3=SML
SET  PQTY=50 ALCTYPE=CYL CONTIG=YES REPLACE=YES
      INC DSN=ACCT/ DSORG=PS SIZE<5MB
```

Forces primary and secondary space allocation on data sets selected by the filter list above, depending on the value in the third account field of the step EXEC statement.

Allocate 50 cylinders for all PS accounting data sets that are allocated with less than 50 MB.

## SPACSWIR - Reduce Initial Allocation on Volume Addition

The SPACSWIR function allows the initial allocation on an added volume to be reduced, thereby allowing the volume addition to proceed even if there is not enough space to satisfy the requested size.

An alternate function, SPACSECR, allows secondary extent reduction, but can cause a data set to be extended on numerous small extents on the current volume. With SPACSWIR, when the requested size is not available on the current volume, secondary extents can be directed to an additional volume without first exhausting smaller extents on the current volume, thus reducing fragmentation.

SPACSWIR works in conjunction with SPACVOLA to decrease the initial requested allocation to the size of the largest contiguous extent on all of the available volumes, when a volume addition has been requested by OS/390.

### Global Parameters

The global parameters that affect SPACSWIR are described in the table below.

Parameter	Purpose
DCTYPE	Allows choice of one or more device characteristics to be maintained during volume addition (CACHE, SHARED, DUALCOPY, FASTWRITE)
SKIP	Specifies checks to be bypassed during volume addition
VSAMLIMWARN	Specifies the percentage value to be used before issuing the 4GB-limit message
VSAMPRIM	Use primary size of VSAM volume extensions

### Rule List Parameters

SET Statement (in member SMRLST<sub>xx</sub>)

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNT <sub>xx</sub> member
NOCHECK=xxxxxx	Specifies the checks bypassed in volume add validation
SPACSWIR=( <i>nnn,nnn</i> )	Specifies the lower limit and decrement of space reduction (0-100)

## Rule List Parameter Explanations

### EVENTID=

**Purpose:** Specifies the identifier assigned to a user event in SMEVNT $xx$ . 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 SMEVNT $xx$ .

**Default:** None

### NOCHECK=

**Purpose:** Specifies the checks bypassed in space recovery validation.

**Syntax:** NOCHECK=CONTIG

**Default:** None

### SPACSWIR=

**Purpose:** Specifies the lower limit and decrement of space reduction, both as a percentage of the original primary allocation value. The first value specifies a lower limit, below which SPACSWIR will not go. The second value is the percentage by which the primary allocation will be decremented. For example, if SPACSWIR=(50,10), the function will decrement the original primary value by 10% on each attempt to find a primary extent but will not decrement the primary size more than 50% of the original value. The decrement percentage is only used for SMS-managed data sets. For non-SMS managed data sets, the allocation uses the largest contiguous extent on the volume.

**Syntax:** SPACSWIR=( $nnn,nnn$ )

where the first  $nnn$  is the floor limit and the second  $nnn$  is a percentage from 0-100 by which reduction can take place until either it fits or the floor limit is reached. (A specification of 100 will not reduce the secondary size at all.)

**Default:** SPACSWIR=(0,10)

### Filter List Parameters Not Supported

The following filter list parameters are not supported for SPACSWIR:

ENVIR	FILESEQ	GDGVER
LABELTYP	VALUE	VCOMPLLQ

## Usage Notes

When a volume addition condition is recognized by OS/390, the SPACVOLA function will attempt to add a new volume. SPACVOLA will search all eligible volumes for an extent large enough for the requested extension. If a suitable extent is not found, the volume add will not occur. However, if SPACSWIR is active and SPACVOLA does not find a volume large enough to hold the new extent, the allocation will proceed to the volume with the largest contiguous extent. SPACVOLA/SPACSWIR will continue to extend the growing data set across multiple volumes, generally one extent on each new volume. For SMS-managed data sets the decrement percentage is used instead of the best fit method.

For non-VSAM data sets, SPACSECR specifications override any processing by SPACSWIR, that is, if both SPACSECR and SPACSWIR are set up to include the same non-VSAM data set, only SPACSECR will handle it. For VSAM data sets, because the initial extent on an added volume is a primary extent, SPACSWIR will operate for that initial extent, and SPACSECR will then process subsequent extents on that volume.

SPACSWIR is particularly effective in reducing fragmentation during secondary allocations for VSAM data sets, which can have up to 123 secondary extents. Where SPACVOLA/SPACSECR by themselves may allow the data set to use many small extents on the current volume, SPACVOLA/SPACSWIR adds a new volume for each extent that the current volume cannot support, selecting the volume with the largest contiguous extent.

SPACSWIR does not process DB2 databases.

Note that the VSAMPRIM global parameter (specified in SMMSYSxx) also affects extent size on volume additions.

In general, extent-size during allocation/volume-addition can be controlled by several StopX37/II functions, as follows:

SPACSQTY adds or overrides requested space on new allocations.

SPACPRIM reduces the size of the primary extent if the requested value cannot be found.

SPACSWIR reduces the size of the first allocation on an added volume if the requested value cannot be found.

SPACVOLA adds a volume when OS/390 encounters an end-of-volume condition:

- when at least one of the eligible volumes contain an extent large enough to satisfy the requested size

*OR*

- when SPACSWIR is active, and none of the eligible volumes contain an extent large enough to satisfy the requested size.

### Messages

SPACSWIR issues message SVM4801I when the function successfully reduces a secondary allocation for a data set:

SVM4801I REDUCED SPACE FROM *orga units* TO *newa*

### Example

Reduce the allocation for production jobs if there is no volume with enough space.

#### SMFUNC09 member

SET	NAME=SPACSWIR ACTIVE=YES	The function is defined and activated.
	MSG=I SMF=I	
	FLST=04 RLST=04	
	DESC='REDUCE ALLOC DURING SPACVOLA'	

#### SMFLST04 member

SET	MODE=ACT	Select only production jobs for reduced secondary processing.
INC	JOBACCT2=PROD*	

#### SMRLST04 member

SET	SPACSWIR=60	Set a lower limit for space reduction of 60% for all jobs in classes A, F, and S.
INC	JOBCLASS=(A,F,S)	
SET	SPACSWIR=30	Set a lower limit for space reduction of 30% for all jobs in classes K and U.
INC	JOBCLASS=(K,U)	
SET	SPACSWIR=0	Set no lower limit for space reduction for all jobs in class C.
INC	JOBCLASS=C	

## SPACVOLA - Add Volume During Allocation

The SPACVOLA function adds a volume to a data set when OS/390 is unable to find sufficient space on the current volume.

OS/390 allocates up to 16 extents on a volume for non-VSAM data sets and up to 123 extents for VSAM data sets. If this limit is exceeded and additional space is needed, or if the space needed is not available on any volumes specified for the data set in its original allocation, the expansion of the data set fails.

SPACVOLA adds a volume from the current pool or an alternate pool (both from pools specified in SMPOOLxx) when OS/390 has allocated 16 extents on a volume for a sequential data set, or when OS/390 could not find space on the current volume. SPACVOLA can prevent out-of-space conditions for data sets allocated on a single volume, and provides flexibility in secondary volume allocations for data sets allocated on multiple volumes.

### Global Parameters

The global parameters that affect SPACVOLA are described below.

Parameter	Purpose
DADSMEX	Determines if the DADSM preprocessing exit (IGGPREE00) is called
DCTYPE	Allows choice of one or more device characteristics to be maintained during volume addition (CACHE, SHARED, DUALCOPY, FASTWRITE)
DFREORGPRC	Defines default SPACVOLA reorganize procedure name
MAXVOL	Limits the number of volumes a data set is allowed to use
REQTYPE	Specifies if the MNTYPE statement is considered the request type instead of the mount type
SCAT	Forces immediate catalog update during volume add
SKIP	Specifies checks to be bypassed during volume addition
VSAMLIMWARN	Specifies the percentage value to be used before issuing the 4GB-limit message
VSAMPRIM	Use primary size for VSAM volume extensions
VSAMZSEC	Controls out-of-space recoveries for VSAM files with zero secondary space coded
X37POOL	Specifies which volume is used by X37 to determine the POOL name in EOVS processing

## Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
ALTPOOL=xxxxxxx	Specifies an alternate pool to search for space
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
MNTYPE=xxxxxxx	Mount type (PUBLIC, STORAGE, PRIVATE CURRENT, ALL)
NOCHECK=xxxxxxx	Specifies checks bypassed in volume add validation
OPER= YES/NO	Allow operator to provide volume when system cannot find space for a volume add. This option is not available for SMS-managed data sets.
PCTI=nnnnn	Specifies the percentage value by which a secondary allocation is increased
REORG= YES/NO	Indicates whether SPACVOLA processing should automatically start a started task to reorganize the file that was just made multivolume by SPACVOLA.
REORG_NSMS=(xx,pool)	SMRORGxx member name suffix in parmlib for DFDSS reorganize job control cards; name of pool to which MAINVIEW SRM is to reorg
REORG_PROC=xxxxxxxx	Reorganize procname for SPACVOLA to start
REORG_SMS=(xx,storclas)	SMRORGxx member name suffix in parmlib for DFDSS reorganize job control cards; name of DFSMS Storage Class to which MAINVIEW SRM is to reorganize
SPACVOLA=nn	Specifies the maximum volumes for a data set (1-59)
UNIT=xxxxxxxx	Unit generic name (1-8 characters)
USECPOOL= YES/NO	Specifies if the current pool is the first to be searched
VOLSER=(xxxxxx,xxxxxx,...)	Volume serial ID (1-6 characters or a mask of 1-5 asterisks)

## Rule List Parameter Explanations

### ALTPOOL=

**Purpose:** Specifies the name of a pool to be used as an alternate to the current pool. If USECPOOL=YES, the alternate pool is searched after the current pool is searched. If USECPOOL=NO, the alternate pool is the only pool searched. This is an optional parameter.

This parameter is ignored for DFSMS data sets.

**Syntax:** ALTPOOL=xxxxxxx

where xxxxxxxx is a pool name from those specified in SMPOOLxx. If the pool name is not defined, the refresh or start-up will fail.

Default: None

### 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

### MNTYPE=

**Purpose:** Specifies the mount type for additional volumes in secondary space allocation. MNTYPE=ALL considers all volumes in the pool, regardless of the volume's mount type. MNTYPE=CURRENT searches only for volumes with the same mount type as the current volume. Note that regardless of the MNTYPE specification, all volumes in an alternate pool are considered eligible.

**Syntax:** MNTYPE=xxxxxxx

where xxxxxxx is the mount status value from the following list:

ALL	All volumes in pool, regardless of mount type
CURRENT	Current volume
PRIVATE	Private volume
PUBLIC	Public volume
STORAGE	Storage volume

Default: MNTYPE=ALL

**NOCHECK=**

**Purpose:** Specifies the checks bypassed in space recovery validation.

**Syntax:** NOCHECK=(*xxxxxx*,...)

where *xxxxxx* is one or more values from the following list:

CONTIG	A data set is allocated with contiguous space required.
DC	A data set resides on a cached device. Under normal conditions, the volume add occurs only to packs that have the same device characteristics.
DSNAME	A data set is allocated to another DD statement within the same job step.
DISP	A permanent data set is being accessed without the use of a catalog.
ENQ	A permanent data set is allocated to a DD statement within another job.
EXCP	A data set is being processed with the EXCP access method (or otherwise processing at the hardware level).
NOTE	A data set is being processed with the NOTE macro.
POINT	A data set is being processed with the POINT macro.

**Default:** None

**OPER=**

**Purpose:** Allows the operator to provide a volume when the system cannot find space for a volume add during SPACVOLA. This option is not available for SMS-managed data sets.

**Syntax:** OPER=*YES/NO*

**Default:** OPER=NO

**PCTI=**

**Purpose:** Specifies the percentage value by which a secondary allocation is increased.

**Syntax:** PCTI=*nnnnn*

where *nnnnn* is a number between 0 and 10000.

**Default:** None

**REORG=**

**Purpose:** Specifies whether SPACVOLA will start a started task to reorganize a data set that has just been made multivolume by SPACVOLA.

**Syntax:** REORG=*YES/NO*

**Default:** REORG=NO

**REORG\_NSMS=**

**Purpose:** Specifies the SMRORGxx suffix that contains the DFDSS control cards to be used by the DFDSS reorganize started task to reorganize a non-SMS multivolume data set. The SMRORGxx member must reside in the MAINVIEW SRM parmlib. Also specifies the MAINVIEW SRM pool name of the target pool to which the multivolume data set is to be reorganized.

**Syntax:** REORG\_NSMS=(*xx,poolname*)

where *xx* is the SMRORGxx suffix and *poolname* is the MAINVIEW SRM target pool for the reorganize.

**Default:** None

**REORG\_PROC=**

**Purpose:** Specifies the name of the procedure library member to be used as the started task JCL for the DFDSS reorganize job that is automatically started by SPACVOLA processing if REORG=YES is specified in the SPACVOLA RLST SET statement. This overrides the default reorg proc specified by the DFREORGPRC parameter in the SMMSYSnn member.

**Syntax:** REORG\_PROC=(*procname*)

where *procname* is the procedure library member to use as the started task JCL for the reorganize job.

**Default:** REORG\_PROC=REORGPRC

**REORG\_SMS=**

**Purpose:** Specifies the SMRORGxx suffix that contains the DFDSS control cards to be used by the DFDSS reorganize started task to reorganize an SMS multivolume data set. The SMRORGxx member must reside in the MAINVIEW SRM parmlib. Also specifies the DFSMS Storage Class to be used as the target Storage Class to which the multivolume data set is to be reorganized.

**Syntax:** REORG\_SMS=(*xx,storclas*)

where *xx* is the SMRORGxx suffix and *storclas* is the DFSMS target Storage Class for the reorganize.

**Default:** None

**SPACVOLA=**

**Purpose:** Specifies the maximum number of volumes for a single data set. SPACVOLA does not add a volume if it would cause the data set's total volume count to exceed this value. If a volume count is specified in the JCL for a job using multiple volumes, the SPACVOLA= value should be set higher than the volume count. Otherwise, SPACVOLA processing will not take effect. Alternatively, you can remove the volume count specification from the job's JCL. A value up to 59 can be specified. This is a required parameter.

**Syntax:** SPACVOLA=*nn*

where *nn* is a number in the range 1-59.

**Default:** None

**UNIT=**

**Purpose:** Contains or specifies the unit generic name (esoteric, generic, or pool name) to be used by the data set.

**Syntax:** UNIT=*xxxxxxxx*

where *xxxxxxxx* is a valid unit name (esoteric, generic, or pool name) 1-8 characters long.

**Default:** None

**USECPOOL=**

**Purpose:** Specifies whether the current pool or the alternate pool is searched to find an additional volume required by a secondary allocation. USECPOOL=YES specifies that the current pool is searched, followed by the alternate pool, if any. USECPOOL=NO specifies that the current pool is not searched; the search begins with the alternate pool, if any. If USECPOOL=NO and no alternate pool is specified, the secondary allocation attempt fails.

This parameter is ignored for DFSMS data sets.

**Syntax:** USECPOOL=*YES/NO*

**Default:** USECPOOL=YES if neither ALTPPOOL nor VOLSER specified, else USECPOOL=NO is default

**VOLSER=**

**Purpose:** Accepts a list of volumes or a special asterisk mask in which leading asterisks require the new volume name to match the existing volume name in the leading asterisk positions. For example, VOLSER=(\*\*\*) will add only volumes for which the first three characters match the existing volume.

There is a limitation to the SET VOLSER= masking facility:

- The / masking character does not work.
- Trailing asterisks work just like leading asterisks.
- Characters must match exactly.

**Syntax:** VOLSER=(*xxxxxx,xxxxxx,...*)

where *xxxxxx* is the 1-6 volume serial ID or 1-5 asterisks.

**Default:** None

**Filter List Parameters Not Supported**

The following filter list parameters are not supported for SPACVOLA:

ENVIR	FILESEQ	GDGVER
LABELTYP	VALUE	VCOMPLLQ

## Usage Notes

SPACVOLA attempts to add a volume to a data set when OS/390 has been unable to allocate space on the current volume. SPACVOLA applies to physical sequential data sets and to VSAM data sets, including DFSMS-managed data sets.

SPACVOLA does not operate on

- data sets with a volume reference (VOL=REF=)
- data sets for which two DD statements exist in the same job step
- concatenated data sets
- data sets being processed with NOTE/POINT
- data sets being processed with EXCP
- uncataloged data sets
- data sets that are open by another job
- data sets that already have the maximum number of volumes (59)
- VIO data sets

### **NOCHECK parameter**

The NOCHECK parameter can be used to override the aforementioned conditions that cause SPACVOLA to not process. See the syntax documentation for NOCHECK for a definition of those conditions. These checks can also be overridden using the global parameter SKIP.

### **Pools**

For non-DFSMS data sets, SPACVOLA works with either the pools defined in SMPOOL $xx$  or with a list of volumes defined in the VOLSER parameter of the SPACVOLA RLST. If pools are used, a data set must already belong to a pool or must have a pool assigned by the ALTPPOOL rule parameter. The pool to which a data set belongs (its current pool) is the first SMPOOL $xx$  pool definition that contains the data set's current volume. If the data set does not reside on a pool volume, the pool to add a volume from can be selected by the ALTPPOOL rule parameter, or the user can specify a list of volumes to choose from by coding the VOLSER rule parameter. If the VOLSER rule parameter is used, a list of specific or generic (using asterisks as generic place holders) volumes can be specified. Specifying generic volumes using asterisks on the volume name allows the user to force data sets to remain on like-named volumes.

Note, however, that EasyPOOL is not required; pool definitions in SMPOOL $xx$  are allowed even if the EasyPOOL component of MAINVIEW SRM is not purchased.

For DFSMS data sets, SPACVOLA searches the data set's storage group for volumes; for non-DFSMS data sets, SPACVOLA searches the current pool and any specified alternate pool for volumes (unless directed otherwise by its action parameters). The volume with the smallest extent that satisfies the request is selected. The defined USELIMIT for the pool is also considered.

### **SPACVOLA parameter**

The SPACVOLA parameter allows specification of the maximum number of volumes for a data set. The SPACVOLA function does not allocate an additional volume if it would cause the data set's volume count to exceed this number. (Note that SPACVOLA overrides any volume count limitation in the SMS data class.)

**Note:** If a volume count is specified in the JCL for a job using multiple volumes, the SPACVOLA= value should be set higher than the volume count. Otherwise, SPACVOLA processing will not take effect. Alternatively, you can remove the volume count specification from the job's JCL.

### **MNTYPE parameter**

The MNTYPE specifies the mount status (actually the use attribute) that a volume must have to be considered eligible. MNTYPE applies only to the current pool; the alternate pool is searched for volumes with any mount type (MNTYPE=ALL).

### **REORG parameters**

The REORG parameter determines if SPACVOLA processing will automatically start a started task to reorganize the data set that has just been made multivolume by SPACVOLA back to a single volume. If REORG=YES is specified, the REORG\_PROC parameter specifies the procedure library member name to use on the automatically generated OS/390 START command that SPACVOLA issues. The REORG\_NSMS and REORG\_SMS (for non-SMS and SMS reorgs respectively) parameters identify the control card member (SMRORGxx where xx is the first operand of the REORG\_NSMS and REORG\_SMS parameters) in the MAINVIEW SRM parmlib that contains the DFDSS control cards for the reorganize started task. For non-SMS reorganizes, the REORG\_NSMS parameter

specifies in its second operand the MAINVIEW SRM pool name to be used to build a volume list of volumes to which the reorganized data set can be copied. For SMS reorganizes, the REORG\_SMS parameter specifies in its second operand the SMS Storage Class to be used in the DFDSS control cards as the Storage Class to which the data set is to be copied during the reorganize. The reorganize started task will only be submitted once for each job, which causes the data set to be extended to multiple volumes even if the data set is extended more than once. See Appendix B (X37REORG Utility) for a complete explanation of setting up the started task and the control card members.

### **VSAMPRIM global parameter**

For VSAM data sets, the primary extent on the new volume added by SPACVOLA is for the secondary extent size defined for the VSAM data set. The VSAMPRIM global parameter of SMMSYSxx can be used to direct the use of the primary extent size for added volumes. If not specified in SMMSYS.xx, VSAMPRIM defaults to NO (primary size is not used); VSAMPRIM=YES must be specified to use the primary rather than the secondary extent size. Since VSAMPRIM is a global parameter, it applies to all VSAM data sets processed by SPACVOLA.

### **DCTYPE global parameter**

The DCTYPE global parameter controls which device characteristics are maintained across volumes during SPACVOLA processing. See the DCTYPE definition in the MAINVIEW SRM for OS/390 Implementation Guide for a complete description of the DCTYPE options.

### **MAXVOL global parameter**

The MAXVOL global parameter is used as the system default for the SPACVOLA RLST SET parameter. If SPACVOLA is not specified, the MAXVOL setting is used to determine the maximum number of volumes the data set can extend to.

### **SCAT global parameter**

The SCAT global parameter determines whether the data set is recataloged immediately after a volume is added or at step termination. SMS-managed data sets are always recataloged immediately.

**VSAMZSEC global parameter**

The VSAMZSEC global parameter controls recovery for VSAM data sets that do not have a secondary space allocation specified. If VSAMZSEC=NO is specified, SPACVOLA controls whether the recovery is processed. If VSAMZSEC=YES is specified, the SPACSECA function controls whether the recovery is processed.

**NOCHECK parameter**

The SPACVOLA function by default excludes data sets being processed with EXCP, NOTE/POINT, or CONTIG. The NOCHECK parameter can be used to cause SPACVOLA to process these data sets.

The SPACVOLA function must be specified in the SMFUNCxx member to be available and must be set active to provide service.

**Warning!** SPACVOLA (and OS/390) will allow a data set to extend across as many as 59 volumes. However, some products using DFP 3.3 CAMLST services will only process up to 20 volumes, due to a limitation in the CAMLST processing (see IBM manual *MVS/DFP 3.3 System Programming Reference*, section 4.3, “Retrieving Information from a Catalog”).

This is not a problem with any MAINVIEW SRM function. However, if you are using DFP 3.3 or earlier, and are using products that use CAMLST services to process multivolume data sets, you may wish to set the SPACVOLA parameter to less than 20 volumes.

Sort work data sets should be excluded from SPACVOLA processing. This can be done in the filter list by:

```
SET  MODE=INACT
INC  DD=SORTW*
```

In general, extent-size during allocation/volume-addition can be controlled by several StopX37/II functions, as follows:

SPACSQTY adds or overrides requested space on new allocations.

SPACPRIM reduces the size of the primary extent if the requested value cannot be found.

SPACSWIR reduces the size of the first allocation on an added volume if the requested value cannot be found.

SPACVOLA adds a volume when OS/390 encounters an end-of-volume condition

- when at least one of the eligible volumes contain an extent large enough to satisfy the requested size

*OR*

- when SPACSWIR is active, and none of the eligible volumes contain an extent large enough to satisfy the requested size.

### **PCTI Parameter**

When specified with the SPACVOLA function, the secondary is increased by the PCTI value only after a volume add. The new value for the secondary is permanent, therefore, subsequent extents on the new volume will be the same size as the first extent allocated after the volume add.

The NVOL selection parameter can be used to control the secondary increases after a volume add occurs.

### **Example**

```
SET SPACVOLA=4 PCTI=200
INC DSN=/ NVOL < 3
```

Allow up to five volumes for all data sets and increase the secondary allocation on the first volume add by the second volume add by 200 percent, but on the second volume add multiply the new secondary (not the original) by 200 percent thereby quadrupling the original secondary on the second volume add.

### **Messages**

SPACVOLA issues message SVM4400I when it adds a volume:

```
SVM4400I VOLUME ADDED - OLD VOLUME volume, NEW VOLUME
volume
```

SPACVOLA issues one of the following messages when it could not add a volume:

```
SVM4451I RECOVERY TERMINATED: DATA SET IS NOT BEING
PROCESSED SEQUENTIALLY
```

```
SVM4452I RECOVERY TERMINATED: DATA SET BEING PROCESSED BY
EXCP
```

```
SVM4453I RECOVERY TERMINATED: DATA SET BEING PROCESSED
USING NOTE/POINT LOGIC
```

SVM4454I RECOVERY TERMINATED: PERMANENT DATA SET REACHED WITHOUT CATALOG. DISP=(NEW/MOD/KEEP)

SVM4455I RECOVERY TERMINATED: PERMANENT DATA SET REACHED WITHOUT CATALOG. DISP=(SHR/OLD/MOD)

SVM4456I RECOVERY TERMINATED: DATA SET IS ON MASS STORAGE VOLUME

SVM4460I RECOVERY TERMINATED: TWO DDNAMES REFERENCING SAME DATA SET

SVM4500I RECOVERY TERMINATED: NO VOLUMES AVAILABLE FOR SWITCH

SVM4519I RECOVERY TERMINATED: DADSM EXIT REJECTED REQUEST

SVM4601I RECOVERY TERMINATED: DATA SET EXISTS ON MAXIMUM VOLUMES

### Example

This example shows you how to

- enable but restrict volume additions to production pools for data sets in production pools
- allow data sets in test pools to draw from volumes in work pools
- disallow data sets in work pools to allocate outside the work pool

#### SMPOOLA2 member

SET	POOLNAME=PRODSTD USELIMIT=90 VOL=PROD0/	Pool PRODSTD includes all volumes whose names begin with PROD0. A 90% capacity limit is put on all volumes in this pool.
SET	POOLNAME=PRODMAX USELIMIT=80 VOL=(PROD21,PROD22,PROD23,PROD24)	Pool PRODMAX includes a number of high-capacity 3390 volumes designated for very large data sets.
SET	POOLNAME=TEST00 EXC VOL=TESTC/ INC VOL=TEST/	Pool TEST00 includes all volumes whose names begin with TEST, except those beginning with TESTC.
SET	POOLNAME=WORK VOL=WORK/ VOL=UTIL/	Pool WORK includes all work and utility volumes.

#### SMFUNCA4 member

SET	NAME=SPACVOLA ACTIVE=YES MSG=I SMF=I FLST=B4 RLST=C4 DESC='ADD VOLUMES TO ALLOCATION'	The function is defined and activated.
-----	--	--

**SMFLSTB4** member

SET MODE=ACT  
INC DSN=/

Select all data sets for processing.

**SMRLSTC4** member

SET SPACVOLA=5 ALTPOOL=PRODMAX  
USECPool=NO  
INC POOL=(PRODSTD,PRODMAX)

Allow up to five volumes for a data set currently in pool PRODSTD. Expand into pool PRODMAX before expanding within PRODSTD.

SET SPACVOLA=10 ALTPOOL=PRODSTD  
INC POOL=(PRODMAX,PRODSTD)

Allow up to 10 volumes for a data set currently in pool PRODMAX. Allow expansion into pool PRODSTD if space cannot be found in PRODMAX.

SET SPACVOLA=2 ALTPOOL=WORK  
INC POOL=(TEST00,WORK)

Allow two volumes for a data set currently in pool TEST00. Allow expansion into pool WORK if space cannot be found in pool TEST00.

SET SPACVOLA=4 MNTYPE=CURRENT  
INC POOL=WORK

Allow up to four volumes for a data set currently in pool WORK. Restrict additional volumes to the same mount type as the current volume.

SET SPACVOLA=4 REORG=YES  
REORG\_PROC=X37REORG  
REORG\_NSMS=(NS,POOL1)  
REORG\_SMS=(SM,STC1)  
INC POOL=WORK

Allow up to four volumes for any data set currently in pool WORK. By default, add volumes in the current pool, which is WORK. When a volume is added, start a started task with procedure library member X37REORG to consolidate the data set back to a single volume using a DFDSS copy job. For SMS data sets, copy the multivolume data set to the STC1 Storage Class and use MAINVIEW SRM parmlib member SMRORGSM to input the DFDSS control cards. For non-SMS data sets, copy the multivolume data sets to a volume in MAINVIEW SRM pool POOL1 and use MAINVIEW SRM parmlib member SMRORGNS to input the DFDSS control cards.

## VIOALLOC - Control VIO Allocation

The VIOALLOC function changes the allocation of temporary data sets to virtual storage (VIO).

The OS/390 VIO facility provides a significant advantage over normal DASD I/O. VIO resides in the paging space, and I/O operations are performed at the speed of main storage access.

VIOALLOC changes the allocation of selected temporary data sets to (or from) VIO, with no JCL modifications.

### Global Parameters

The global parameters that affect VIOALLOC are described below.

Parameter	Purpose
MSGPREF	MAINVIEW SRM message identifier prefix
SIZEISPRIM	Determines if the SIZE filter/rule list parameter includes only the size of the primary extent or the size of the primary and one secondary extent
X37POOL	Specifies which volume is used by X37 to determine the POOL name in EOVS processing

### Rule List Parameters

SET Statement (in member SMRLSTxx)

Parameter	Description
EVENTID=xxxxx	Identifies a user event defined in an SMEVNTxx member
VIO=YES/NO	Specifies whether the data set is allocated in VIO space

### Rule List 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 SMEVNT*xx*.

Default: None

### **VIO=**

Purpose: Specifies whether the new data set is allocated in VIO. This is a required parameter; VIO=YES or VIO=NO must be specified for any processing to occur.

Syntax: VIO=*YES/NO*

Default: None

### **Filter List Parameters Not Supported**

The following filter list parameters are not supported for VIOALLOC:

- GDGVER
- VALUE

## **Usage Notes**

VIOALLOC applies to new temporary data sets. VIOALLOC changes the allocation of selected data sets to VIO from DASD if the VIO parameter is YES, or to DASD from VIO if the VIO parameter is NO.

VIOALLOC operates only on temporary data sets (DSN=&&*xxxxx*).

MAINVIEW SRM makes the data set eligible for VIO allocation, but OS/390 actually processes the request. Thus, OS/390 considerations apply to data sets switched to VIO. VIO allocation is discussed in “Data Set Resources - Allocation” in the IBM *MVS JCL User's Guide*.

VIOALLOC issues two messages.

- When a data set is directed to VIO:

SVM3439I *job,step,dd,dsn* MADE VIO ELIGIBLE

- When a data set is prohibited from VIO:

SVM3440I *job,step,dd,dsn* VIO ELIGIBILITY REMOVED

If a data set is already allocated VIO, and VIO=YES is specified, no action is taken, and a message is not issued. Similarly, if a data set is not allocated VIO, and VIO=NO is specified, no action is taken, and a message is not issued.

Note that the VIOALLOC function must be specified in the SMFUNCxx member to be available and must be set active to provide service.

### Example

This example shows you how to specify VIO usage for all data sets with a maximum allocation of less than five megabytes and ensure sort work data sets are not allocated to VIO.

#### SMFUNC48 member

```
SET  NAME=VIOALLOC ACTIVE=YES      The function is defined and
      MSG=I SMF=I                  activated.
      FLST=21 RLST=54
      DESC='CONTROL VIO ALLOCATION'
```

#### SMFLST21 member

```
SET  MODE=ACT                      Select all data sets except those
      EXC DDNAME=SORT/              with sort-related ddnames.
      INC DSN=/'
```

#### SMRLST54 member

```
SET  VIO=YES                        Allow VIO for data sets whose
      INC MAXSIZE<5MB               maximum size is less than five
                                   megabytes.
```

---

---

# Appendix A Utilities

This appendix provides explanations and examples batch reports provided by StopX37/II.

SMF Analysis Program (SMFX37A) . . . . .	A-2
StopX37/II Analysis Program (SMFX37B) . . . . .	A-7
DASD Utilization Report (X37UTILC) . . . . .	A-19
Dynamic Multivolume Data Set Extent Consolidation (X37REORG) . . . . .	A-25
Data Set Reorganization (REDUCEXT) . . . . .	A-30

## Overview

The following batch utility programs are supplied with StopX37/II:

- The SMFX37A program reads SMF step and job termination records and generates a report of the B37/D37/E37 abends that have occurred. This program can be used to quantify the amount of time lost due to disk space problems.
- The SMFX37B program reads and formats the SMF records produced by StopX37/II. These records contain detailed information about the jobs that were intercepted for a recovery attempt. This report provides information about the successful attempts, as well as the jobs that StopX37/II failed to recover. The SMFX37B program can be used to determine why a job was allowed to abend.
- The X37UTILC program provides a report that details the percentage of used and unused DASD space and the cost of DASD space. This report can be used to determine the costs and savings associated with a particular level of DASD utilization.
- X37REORG is a dynamically started DFDSS copy utility that reorganizes a multivolume data set back to a single volume.
- REDUCEXT is a user-written utility that reorganizes multivolume data sets onto a single volume.

## SMF Analysis Program (SMFX37A)

The SMF Analysis Program (SMFX37A) analyzes SMF data and produces a report on jobs that have failed with B37/D37/E37 abends. The report includes the job name, program name, date, and time of each job that failed. The report also includes the CPU and elapsed time lost with the failing job. The CPU and elapsed times are for the entire job, not just the step that abended.

This program can be used to evaluate the amount of time lost due to X37 abends and can also be beneficial in determining criteria needed to tailor the StopX37/II selection language. After determining your selection criteria, run this report for several past reporting periods and evaluate which jobs might have been saved and which jobs would have been bypassed.

## Program Logic

This program reads SMF type=30 records, looking for any jobs ending with B37, D37, or E37 abend codes. After end-of-file on the SMF input data set, the saved SMF records are printed and totaled.

## Execution JCL

This section describes the execution JCL used by the SMFX37A program.

<b>Statement</b>	<b>Usage</b>
JOB	This job needs a region of at least 512KB to function properly.
EXEC	Specifies the program name for the SMFX37A program (PGM=SMFX37A). This statement can include optional PARM information used to override the default parameters (refer "PARM Information" on page 85).
STEPLIB DD	Defines the load libraries required for executing the SMFX37A program. The MAINVIEW SRM load library is required.
SMFFILE DD	Defines an input sequential data set. The SMFX37A program scans this file for SMF job termination records contained in type 30 records only. You may need to examine the SMF parameters to determine which records are available. This information can be found in SYS1.PARMLIB(SMFPRMnn). The DCB parameters are: DCB=(RECFM=VBS,BFTEK=A).
SMFREPT DD	Defines a sequential output data set used for listing information. The DCB characteristics are: DCB=(LRECL=133,RECFM=FBM).
SYSIN DD	Defines the input control statement stream when PARM=SYSIN is specified. This data set should be an 80 character sequential data set or PDS member. PARM=SYSIN is mutually exclusive with other input parameters on the EXEC card.
SYSPRINT DD	Defines a sequential output data set used to display calling parameters. The DCB characteristics are: DCB=(LRECL=133,RECFM=FBM).

## Sample JCL

The sample JCL can be found in the JCLX37AA member of *?prefix.BBSAMP*. The source for this program can be found in the SMFX37AA member of that data set.

The SMFFILE DD must be a sequential data set. If your SMF data sets are VSAM files, you must extract the SMF data using the IBM SMF conversion utility (IFASMFDP).

## Parameters

During initialization, the SMFX37A program processes control information passed to the routine by the PARM field on the EXEC statement. These parameters are used to specify installation dependent options. A list of the parameters and their defaults follows.

### COST=

**Function:** The COST parameter specifies the average dollar cost per hour for the personnel time used to fix a job that resulted in a B37/D37/E37 abend.

**Default:** 50

**Format:** Any numeric value.

### CPU=

**Function:** The CPU parameter specifies the average dollar cost per hour for the CPU time lost by a job that resulted in a B37/D37/E37 abend.

**Default:** 250

**Format:** Any numeric value.

### CURSIGN=

**Function:** The CURSIGN= parameter is used to provide a currency symbol other than the US Dollar sign. Some national currency symbols are assigned to the EBCDIC code X'5B' and therefore print correctly (that is, the UK and Irish Pound sign, the Italian Lire, and the Japanese Yen). Other currencies may be specified with this parameter.

**Default:** \$

**Format:** CURSIGN= 2 characters (that is, DM, FF, BF, DF, SF, PS, KR, FM)

**PEOPLE=**

**Function:** The PEOPLE parameter specifies the average amount of time it will take someone to fix a job that resulted in a B37/D37/E37 abend. The time specified is in minutes.

**Default:** 30

**Format:** Any numeric value.

**SYSIN=**

**Function:** The SYSIN parameter indicates that parameters are being provided via the SYSIN DD statement instead of the JCL PARM field. When SYSIN is coded, it should be the only parameter coded on the EXEC PARM= parameter.

Parameters coded in the SYSIN DD statement file may start in any column. One or more parameters may be coded per card image, but no parameter may span a card boundary. Cards that begin with an asterisk (\*) in column 1 are treated as comments. All parm options and SYSIN card images are printed on the SMFX37A report for reference.

**Format:** //STEP1 EXEC PGM=SMFX37A , PARM=SYSIN  
//SYSIN DD \*

Following is a sample of how to code parms using SYSIN:

COST=25 , CURSIGN=DM

**Sample Output**

Following is a sample report produced by the SMFX37A program.

**Figure A-1 SMFX37A Sample Report**

REPORT SMFX37A		ABEND ANALYSIS			DETAIL REPORT		
VERSION: 7.2.0							
JOB NAME	PROGRAMMER NAME	PROGRAM	DATE	TIME	ABEND	CPU TIME	ELAPSE
TIME							
PRO1ED01	PAYROLL EDIT	PRO734	01/12/2000	14:12:33	B37	0:01:23	0:32:22
OSTWE33	OPERATIONS	FILECOPY	01/12/2000	16:00:02	B37	0:05:12	1:03:46
OUT27DE	E.CHANDLER	ACCRUPT	01/13/2000	2:11:54	B37	0:00:43	0:10:34
TOT12PR	ACCOUNTING	RC123ED	01/13/2000	3:51:02	D37	0:04:22	0:23:05
CDE1RRR	SYSTEMS	IEBCOPY	01/13/2000	9:21:43	D37	0:00:03	0:01:27
OUT88UPT	ACCOUNTING	ACR12UPT	01/13/2000	11:53:01	B37	0:23:19	3:12:33
COPY1	R.WARD	EDRM	01/13/2000	12:42:11	B37	0:02:42	0:35:59
JHDD23	R.BELL	FRD7	01/13/2000	14:24:29	B37	0:07:11	0:52:01
LNS26DEV	S.DEVITO	IEBGENER	01/13/2000	14:44:41	D37	0:01:51	0:41:31
OSTWE33	OPERATIONS	FILECOPY	01/13/2000	15:03:20	B37	0:01:02	0:27:58
TESTA	C.LINWOOD	SAR02ED	01/13/2000	23:02:22	B37	0:03:22	1:27:19
SMFDUMP	OPERATIONS	IEBGENER	01/14/2000	1:51:11	B37	0:04:53	0:35:47
GLU34RP	ACCOUNTING	GL94REP	01/14/2000	3:33:54	B37	0:13:33	2:28:22

**Figure A-2 Abend Analysis Report Sample**

REPORT SMFX37A		ABEND ANALYSIS		SUMMARY REPORT	
REPORTING PERIOD FROM : 03/01/2002 TO : 03/01/2002					
	NUMBER OF ABENDS	CPU TIME LOST	ELAPSE TIME LOST		
B37 ABENDS	10	1:03:20	11:11:41		
D37 ABENDS	3	0:06:16	1:06:03		
E37 ABENDS	0	0:00:00	0:00:00		
** TOTAL	13	1:09:36	12:17:44		
-	BASED ON USER SUPPLIED CRITERIA				
	PERSONNEL TIME PER ABEND	15	MINUTES		
	PERSONNEL COST PER HOUR	\$25.00	PER HOUR		
	CPU COST PER HOUR	\$475.00	PER HOUR		
	THE PERIOD REPRESENTS	.85	PERCENT OF A YEAR		
-	SAVING FOR THIS PERIOD				
	PERSONNEL TIME	3:15:00			
	PERSONNEL COST	\$162.50			
	COMPUTER TIME COST	\$550.21			
-	PROJECTED ANNUAL SAVING				
	PERSONNEL TIME	395:25:00			
	PERSONNEL COST	\$9,885.42			
	COMPUTER TIME COST	\$66,460.41			

## StopX37/II Analysis Program (SMFX37B)

The StopX37/II Analysis Program (SMFX37B) can be used to monitor StopX37/II and its optional components. For information to be available, the base product and the desired product option(s) must be started (refer to the *MAINVIEW SRM User Guide and Reference* accompanying this product for information on starting StopX37/II and the other optional components).

The SMFX37B program analyzes the SMF file for StopX37/II SMF records. The report produced contains the job name, program name, job start time and date, data set name, and action taken. There may be multiple entries for the same job, since StopX37/II may have made several recovery attempts while saving the job.

The user is encouraged to make this program a part of the daily SMF processing routine. If StopX37/II has volume switched a permanent data set, the SMFX37B program can inform your installation's DASD manager which data sets are multivolume.

In large installations, where the amount of SMF data generated makes it difficult to run the SMFX37B program, it is possible to change the SMF dump program to write StopX37/II records to a separate data set. The following job stream selects StopX37/II SMF records and writes them to a generation data set residing on disk.

**Figure A-3 Writing SMF records to a Generation Data Set**

---

```
//SMFDUMP JOB ...
//DUMP EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=A
//DUMPIN DD DSN=SYS1.MANX,DISP=SHR
//DUMPOUT DD DSN=SYS1.DUMP(+1),DISP=(NEW,CATLG),UNIT=TAPE
//X37OUT DD DSN=SYS1.X37OUT(+1),DISP=(NEW,CATLG),UNIT=SYSDA
//SYSIN DD *
    INDD(DUMPIN,OPTIONS(ALL))
    OUTDD(DUMPOUT,TYPE(000:255))
    OUTDD(X37OUT,TYPE(201))
/*
```

---

**Execution JCL**

This section describes the execution JCL used by the SMFX37B program.

<b>Statement</b>	<b>Usage</b>
JOB	Initiates the job. This job needs a region of 512K to function properly.
EXEC	Specifies the program name for the SMFX37B program (PGM=SMFX37B). This statement can include optional PARM information used to override the default parameters (refer to “PARM Information” on page 88).
SYSIN DD	Defines the input control statement stream when PARM=SYSIN is specified. This data set should be an 80 character sequential data set or PDS member. PARM=SYSIN is mutually exclusive with other input parameters on the EXEC card.
STEPLIB DD	Defines an input partitioned data set. This DD statement allocates to the StopX37/II load library.
SMFFILE DD	Defines an input sequential data set. The SMFX37B program scans this file for the StopX37/II SMF records. These records are written whenever StopX37/II traps a potential X37 abend, or for example, when StopX37/II/RECAT encounters a potential NOT-CATLGD-2 condition (the product and/or option must be started for the records to be written). The DCB information is: DCB=RECFM=VBS.
SYSPRINT DD	Defines a sequential output data set used for listing information obtained from the StopX37/II SMF records. The DCB characteristics are: DCB=(LRECL=133,RECFM=FBA).
PERMDSN DD	Defines a sequential output data set used for listing the permanent data sets that have gone multivolume. The SMFX37B program tests to ensure that these data sets still exist on the volume(s) added by StopX37/II. The DCB characteristics are: DCB=(LRECL=80,RECFM=FB).

## Parameters

During initialization, the SMFX37B program processes the PARM information passed to the routine via the PARM field on the EXEC statement. These parameters are used to pass program and installation dependent options to the SMFX37B program. A list of the parameters and their defaults follows:

### COST=

**Function:** The COST= parameter specifies the average dollar cost per hour for the personnel time used to fix a job that was taking an abend for which StopX37/II or another option would attempt recovery.

**Default:** 50

**Format:** Any numeric value.

### CPU=

**Function:** The CPU= parameter specifies the average dollar cost per hour for the CPU time lost by a job that was taking an abend for which StopX37/II or another option would attempt recovery.

**Default:** 250

**Format:** Any numeric value.

### DUMP=

**Function:** The DUMP= parameter specifies whether the SMFX37B report program writes the entire StopX37/II SMF record to the SYSPRINT file in dump format. The DUMP=YES option forces the record to be dumped after writing the usual report lines.

**Default:** NO

**Format:** YES or NO.

### LINES=

**Function:** The LINES= parameter specifies the number of lines per page used by the SMFX37B report.

**Default:** 54

**Format:** Any numeric value between 10 and 99.

### LIST=

**Function:** The LIST= parameter specifies which entries should be printed. StopX37/II writes records to the SMF file whether or not a recovery attempt is made. LIST=RECOV shows only jobs that StopX37/II attempted to recover, LIST=BYPASS shows jobs where StopX37/II bypassed the recovery attempt, and LIST=BOTH shows all jobs examined by StopX37/II. LIST=NEITHER produces a report that contains only the summary information. Three asterisks (\*\*\*) are printed to the left of the reported recovery attempts.

**Default:** BOTH

**Format:** BOTH, NEITHER, RECOV, or BYPASS

### PEOPLE=

**Function:** The PEOPLE= parameter specifies the average amount of time it would take someone to fix a job that was taking an abend for which StopX37/II or another option would attempt recovery. The time specified is in units of minutes.

**Default:** 30

**Format:** Any numeric value.

### RECID=

**Function:** The RECID= parameter specifies the SMF record number assigned by the SMFID parameter in the SMMSYSxx parmlib member.

**Default:** 201

**Format:** Any number between 129 and 255.

### STEP=

**Function:** The STEP= parameter specifies whether the amount of time lost is based on the job or step execution time. If most of your abending jobs are step restartable, then STEP=YES should be specified.

**Default:** NO

**Format:** YES or NO.

**SUM=**

Function: The SUM= parameter specifies whether the SMFX37B report program appends the report summary to the end of the SYSPRINT file.

Default: YES

Format: YES or NO.

**Additional Run-Time Options**

The following run time options may be specified for the SMFX37B report program. These options are specified on the PARM field of the EXEC PGM JCL statement, or may be provided by way of the SYSIN DD statement (see the SYSIN parm option below).

**CURSIGN=**

Function: The CURSIGN= parameter is used to provide a currency symbol other than the US Dollar sign. Some national currency symbols are assigned to the EBCDIC code X'5B' and therefore print correctly (that is, the UK and Irish Pound sign, the Italian Lire, and the Japanese Yen). Other currencies may be specified with this parameter.

Default: \$

Format: CURSIGN= 2 characters (that is, DM, FF, BF, DF, SF, PS, KR, FM)

**RTYPE=**

Function: The RTYPE= parameter allows you to select what type of recoveries are to be included in the report. If more than one recovery type is provided, they must be separated by commas and enclosed in parentheses. Using this parameter, it is possible to quantify the actual and potential savings for individual recovery types.

Default: All recovery types are included.

Format: (SPACSECA | SPACSECB | SPACSECI | SPACSECR | SPACVOLA | SPACSWIR | SPACPRIM | NOCATLG2)

### **TSO=**

**Function:** The TSO= parameter allows TSO sessions to be included (YES) or excluded (NO) from reporting. Typically, TSO sessions do not actually terminate due to file related errors not recovered by StopX37/II.

**Default:** YES

**Format:** TSO=YES or TSO=NO

### **VSAM=**

**Function:** The VSAM= parameter allows recoveries for VSAM files to be included (YES) or excluded (NO). Note: The STOP-X37 VSAM option must be installed and activated to receive SMF records for VSAM recoveries. VSAM=ONLY will print a report with information only about VSAM recoveries. The report may include any detail or summary information desired (see the LIST and SUM options) and will quantify the actual and potential savings from this option.

**Default:** YES

**Format:** VSAM=YES or VSAM=NO or VSAM=ONLY

### **XSTC=**

**Function:** The XSTC= parameter provides the names of started tasks that should be excluded from the report. These are typically long running tasks that do not actually abend after file related errors. If more than one started task name is provided, they should be separated by commas and enclosed in parentheses.

**Default:** None.

**Format:** XSTC=NETVIEW or XSTC=(HSM,NETVIEW)

**SYSIN=**

**Function:** The SYSIN parameter indicates that parameters are being provided via the SYSIN DD statement instead of the JCL PARM field. When SYSIN is coded, it should be the only parameter coded on the EXEC PARM= parameter.

Parameters coded in the SYSIN DD statement file may start in any column. One or more parameters may be coded per card image, but no parameter may span a card boundary. Cards that begin with an asterisk (\*) in column 1 are treated as comments. All parm options and SYSIN card images are printed on the SMFX37B report for reference.

**Format:** //STEP1 EXEC PGM=SMFX37B , PARM=SYSIN  
//SYSIN DD \*

Following is a sample of how to code parms using SYSIN:

RECID=201 , TSO=NO , XSTC= ( HSM , XWTR , NETVIEW )

**Sample JCL**

The following JCL can be found in the JCLX37B member of *?prefix.BBSAMP*. The source for this program can be found in the SMFX37B member of *?prefix.BBSAMP*. The JCLX37B job stream follows. The SORTIN DD must be a sequential data set. If your SMF data sets are VSAM files, you must extract the SMF data using the IBM-supplied SMF conversion utility (IFASMFDP). Fields underlined and in bold type may need to be modified before being submitted.

**Figure A-4      Sample JCL - SMFX37B**

---

```
//SMFX37B JOB ...
//*
//* THIS STEP EXTRACTS AND SORTS StopX37/II SMF RECORDS BY JOBID.
//* NOTE: IF RECID NOT = 201, MODIFY THE X'C9' ON THE INCLUDE
//* CARD TO THE APPROPRIATE RECORD ID.
//*
//SORT EXEC PGM=SORT
//SYSOUT DD SYSOUT=A
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(3,3))
//SORTIN DD DSN=smf.library,DISP=SHR
//SORTOUT DD DSN=&&TEMP,DISP=(NEW,PASS,DELETE),
// SPACE=(CYL,(3,3)),UNIT=SYSDA,
// DCB=(RECFM=VBS,LRECL=32752,BLKSIZE=32760,DSORG=PS)
//SYSIN DD *
INCLUDE COND=(6,1,BI,EQ,X'C9')
SORT FIELDS=(19,16,CH,A,11,4,PD,A,7,4,FI,A)
//*
//* PRODUCE StopX37/II ANALYSIS REPORT
//*
//X37 EXEC PGM=SMFX37B,
// PARM='RECID=201,LIST=BOTH'
//STEPLIB DD DSN=xxxxxxxx.BBLINK,DISP=SHR
//SMFFILE DD DSN=&&TEMP,DISP=(OLD,DELETE)
//SYSPRINT DD SYSOUT=A
//PERMDSN DD SYSOUT=A
```

---

**Sample Output**

The following represents sample detail records and summary reports produced by the SMFX37B program.

**Figure A-5 SMFX37B Detail Report**

```
EXEC PARM: RECID=253,LIST=BOTH
REPORT SMFX37B STOP-X37/II ANALYSIS DETAIL REPORT
SAMPLE EXCERPT
VERSION: 7.2.1
RECOVERY SUMMARY FOR JERRY2VA: JOB04562 READER TIME: 11:56:32 READER DATE: 03/25/00
PGM NAME: X37VER DDNAME: SYSUT2 VOLSER: HSM002 DSORG: PS DATA SET:
JERRY.TEST.SPACVOLA.NSMSCYL1
*** TYPE OF ATTEMPT: SPACSECI
*** SYSLOG MESSAGE: SMS4874I INCREASED SPACE FROM 1 CYLS TO 2
*** TIME OF DAY: 11:57:16 CPU TIME SAVED: 0:00:03 ELAPSE TIME SAVED:
0:00:34 NUMBER OF EXTENTS: 10
*** TYPE OF ATTEMPT: SPACSECI
*** SYSLOG MESSAGE: SMS4874I INCREASED SPACE FROM 1 CYLS TO 3
*** TIME OF DAY: 11:57:19 CPU TIME SAVED: 0:00:03 ELAPSE TIME SAVED:
0:00:37 NUMBER OF EXTENTS: 11
*** TYPE OF ATTEMPT: SPACSECI
*** SYSLOG MESSAGE: SMS4874I INCREASED SPACE FROM 1 CYLS TO 4
*** TIME OF DAY: 11:57:21 CPU TIME SAVED: 0:00:04 ELAPSE TIME SAVED:
0:00:39 NUMBER OF EXTENTS: 12
```

**Figure A-6 SMFX37B Detail Report**

REPORT SMFX37B	STOP-X37/II ANALYSIS	DETAIL REPORT
SAMPLE EXCERPT		
VERSION: 7.2.1		
PGM NAME: X37VER DDNAME: SYSUT2 VOLSER: HSM002 DSORG: PS DATA SET:		
JERRY.TEST.SPACVOLA.NSMSBLK1		
0:08:23	***	TIME OF DAY: 11:05:04 CPU TIME LOST : 0:00:39 ELAPSE TIME LOST :
	***	TYPE OF ATTEMPT: SPACSECI
	***	SYSLOG MESSAGE: SMS4874I INCREASED SPACE FROM 2 BLKS TO 20
	***	TIME OF DAY: 11:59:17 CPU TIME SAVED: 0:00:11 ELAPSE TIME SAVED:
0:02:35		NUMBER OF EXTENTS: 10
	***	TYPE OF ATTEMPT: SPACSECI
	***	SYSLOG MESSAGE: SMS4874I INCREASED SPACE FROM 2 BLKS TO 22
	***	TIME OF DAY: 11:59:18 CPU TIME SAVED: 0:00:12 ELAPSE TIME SAVED:
0:02:37		NUMBER OF EXTENTS: 11
	***	TYPE OF ATTEMPT: SPACSECI
	***	SYSLOG MESSAGE: SMS4874I INCREASED SPACE FROM 2 BLKS TO 24
	***	TIME OF DAY: 11:59:20 CPU TIME SAVED: 0:00:12 ELAPSE TIME SAVED:
0:02:38		NUMBER OF EXTENTS: 12

**Figure A-7 SMFX37B Detail Report**

```

REPORT    SMFX37B                STOP-X37/II ANALYSIS        DETAIL REPORT
SAMPLE EXCERPT
VERSION:  7.2.1
          PGM NAME: X37VER      DDNAME: SYSUT2      VOLSER: SMS002  DSORG: PS  DATA SET:
JERRY.SMS.TEST.SPACVOLA.TSTAVR1
          ***   TIME OF DAY: 12:05:39  CPU TIME LOST :  0:00:42  ELAPSE TIME LOST :
0:08:08
          ***   TYPE OF ATTEMPT: SPACSWIR
          ***   SYSLOG MESSAGE: SMS4118 REDUCED SECONDARY SPACE FROM 45 TRKS TO 15
          PGM NAME: X37VER      DDNAME: SYSUT2      VOLSER: SMS004  DSORG: PS  DATA SET:
JERRY.SMS.TEST.SPACVOLA.TSTAVR1
          ***   TIME OF DAY: 12:05:40  CPU TIME LOST :  0:00:42  ELAPSE TIME LOST :
0:08:58
          ***   TYPE OF ATTEMPT: SPACSECR
          ***   SYSLOG MESSAGE: SMS4802I NO VOL IN STORAGE GROUP WITH ENOUGH SPACE
          PGM NAME: X37VER      DDNAME: SYSUT2      VOLSER: HSM002  DSORG: PS  DATA SET:
JERRY.SMS.TEST.SPACVOLA.NSMSCYL2
          ***   TIME OF DAY: 12:05:45  CPU TIME SAVED:  0:00:37  ELAPSE TIME SAVED:
0:09:03
          ***   TYPE OF ATTEMPT: SPACSECA
          ***   SYSLOG MESSAGE: SMS4855I SET SECONDARY SPACE USING 200% OF PRIMARY ( 1
CYLS) : 2 CYLS
          PGM NAME: X37VER      DDNAME: SYSUT2      VOLSER: HSM002  DSORG: PS  DATA SET:
JERRY.SMS.TEST.SPACVOLA.NSMSTRK2
    
```

**Figure A-8 SMFX37B Detail Report**

```

REPORT    SMFX37B                STOP-X37/II ANALYSIS        DETAIL REPORT
SAMPLE EXCERPT
VERSION:  7.2.1
          RECOVERY SUMMARY FOR JOB: DATAGEN  READER TIME: 12:50:00  READER DATE: 04/05/00
          PGM NAME: IEBDGD      DDNAME: OUTSCR     VOLSER: WORKO2  DSORG: PS  DATA SET:
ICF1.NEWFILE.TESTDAT
          ***   TYPE OF ATTEMPT: SPACSECR
          ***   SYSLOG MESSAGE: SMS54118I REDUCED SPACE FROM 900 CYLS TO 408
          ***   TIME OF DAY: 12:56:00  CPU TIME SAVED:  0:00:00  ELAPSE TIME SAVED:
0:00:06
          RECOVERY SUMMARY FOR JOB: GLEDGRN  READER TIME: 16:48:31  READER DATE: 04/05/96
          PGM NAME: IEFBR14     DDNAME: DD1      VOLSER: WORKO3  DSORG: PS  DATA SET:
PROD.EMP.GLEDG.APRL
          ***   TYPE OF ATTEMPT: SPACPRIM
          ***   SYSLOG MESSAGE: SMS4117I REDUCED PRIMARY SPACE FROM 100000 BLKS TO 30000
          ***   TIME OF DAY: 16:48:40  CPU TIME SAVED:  0:00:01  ELAPSE TIME SAVED:
0:00:04
          RECOVERY SUMMARY FOR JOB: DL4DD92N READER TIME: 12:00:33  READER DATE: 04/05/96
          PGM NAME: IDCAMS      DDNAME: DL4DD92N  VOLSER: WORKO2  DSORG: VS  DATA SET:
TEMP.VSM.DATA
          ***   TYPE OF ATTEMPT: SPACVOLA
          ***   SYSLOG MESSAGE: SMS4400I VOLUME ADDED - OLDVOLUME WORKO2, NEW VOLUME
SCR151
    
```

Figure A-9 SMFX37B Detail Report

```

REPORT    SMFX37B                STOP-X37/II ANALYSIS        DETAIL REPORT
SAMPLE EXCERPT
VERSION:  7.2.1
      RECOVERY SUMMARY FOR JOB: STIH      READER TIME: 07:49:47  READER DATE: 04/05/00
      PGM NAME: IKJEFT01  DDNAME: ISPO7505  VOLSER: TS0004  DSORG: PS  DATA SET:
STIH.SPFTEMP0.CNTL
      ***      TYPE OF ATTEMPT: NOCATLG2
      ***      SYSLOG MESSAGE: SMS4126S RECATALOG NOT ALLOWED DUE TO NOCATVOL=DIFF
      ***      TIME OF DAY: 14:29:46  CPU TIME SAVED:  0:00:04  ELAPSE TIME SAVED:
0:12:26
      RECOVERY SUMMARY FOR JOB: ENQALD1  READER TIME: 15:48:10  READER DATE: 04/05/96
      PGM NAME: X37VER      DDNAME: SYSUT2      VOLSER: WORK03  DSORG: PS  DATA SET:
EAB.E49ALQRT.ND4
      ***      TYPE OF ATTEMPT: SPACSECR
      ***      SYSLOG MESSAGE: SMS4118I REDUCED SPACE FROM 25000 TRKS TO 90
      ***      TIME OF DAY: 15:48:21  CPU TIME SAVED:  0:00:00  ELAPSE TIME SAVED:
    
```

Figure A-10 SMFX37B Detail Report

```

SAMPLE
REPORT    SMFX37B                ABEND ANALYSIS        SUMMARY REPORT
      REPORTING PERIOD FROM : 03/25/00  TO : 04/05/00
      BASED ON USER SUPPLIED CRITERIA
      PERSONNEL TIME PER ABEND                30  MINUTES
      PERSONNEL COST PER HOUR                 $50.00  PER HOUR
      CPU COST PER HOUR                       $250.00  PER HOUR
      SUCCESSFUL RECOVERY ATTEMPTS
      NUMBER OF TOTAL SUCC. RECOVERY ATTEMPTS                200
      NUMBER OF ATTEMPTS FOR SPACSECR                       96
      NUMBER OF ATTEMPTS FOR SPACSECA                       19
      NUMBER OF ATTEMPTS FOR SPACVOLA                        8
      NUMBER OF ATTEMPTS FOR SPACPRIM                       10
      NUMBER OF ATTEMPTS FOR NOCATLG2                       25
      NUMBER OF ATTEMPTS FOR VSAM REDUCE                     0
      NUMBER OF ATTEMPTS FOR 2NDARY INCREASE                 40
      NUMBER OF ATTEMPTS FOR 2NDARY BEST FIT                 1
      NUMBER OF ATTEMPTS FOR SWITCH REDUCE                   1
      TOTAL ELAPSE TIME SAVED                13:24:20
      PROJECTED ELAPSE TIME SAVED                348  HRS/YEAR
      TOTAL CPU TIME SAVED                     1:20:00
      PROJECTED CPU TIME SAVED                 34:40:06  PER YEAR
      TOTAL CPU COST SAVED                     $333.00
    
```

Figure A-11 SMFX37B Detail Report

SAMPLE REPORT	SMFX37B	ABEND ANALYSIS	SUMMARY REPORT
BYPASSED RECOVERY ATTEMPTS			
		NUMBER OF TOT BYPASSED RECOVERY ATTEMPTS	60
		NUMBER OF ATTEMPTS FOR SPACSECR	6
		NUMBER OF ATTEMPTS FOR SPACSECA	5
		NUMBER OF ATTEMPTS FOR SPACVOLA	31
		NUMBER OF ATTEMPTS FOR SPACPRIM	12
		NUMBER OF ATTEMPTS FOR NOCATLG2	3
		NUMBER OF ATTEMPTS FOR VSAM REDUCE	0
		NUMBER OF ATTEMPTS FOR 2NDARY INCREASE	2
		NUMBER OF ATTEMPTS FOR 2NDARY BEST FIT	0
		NUMBER OF ATTEMPTS FOR SWITCH REDUCE	1
		TOTAL ELAPSE TIME LOST	07:03:45
		PROJECTED ELAPSE TIME LOST	183 HRS/YEAR
		TOTAL CPU TIME LOST	0:15:19
		PROJECTED CPU TIME LOST	6:38:16 PER YEAR
		TOTAL CPU COST LOST	\$63.82
		PROJECTED CPU COST LOST	\$1,668.27 PER YEAR
		TOTAL PERSONNEL TIME LOST	30:00:00
		PROJECTED PERSONNEL TIME LOST	780 HRS/YEAR
		TOTAL PERSONNEL COST LOST	\$1,500.00

# DASD Utilization Report (X37UTILC)

The X37UTILC program provides a report on current DASD utilization levels and the cost of unused (wasted) DASD space. X37UTILC also indicates which DASD volumes are sources of potential space errors due to an insufficient number of free DSCBs or a high fragmentation index and identifies volumes that are under-utilized. Parameters can be specified to inform X37UTILC which volumes to examine, the cost per megabyte of disk storage, warning thresholds, and so on.

X37UTILC can be executed from an unauthorized library.

## Control Statements

The following control statements can be specified in the SYSIN data set to control the operation of X37UTILC. If no SYSIN DD statement is present, or if SYSIN is empty, X37UTILC examines all online DASD volumes and produces a report on DASD utilization.

### **VOL=**

**Function:** Identifies the DASD volumes to be examined by X37UTILC. Up to 20 volumes or volume groups can be specified. If more than one volume or volume group is specified, the list must be enclosed in parentheses. A group of volumes can be specified by placing an asterisk (\*) as the last character (for example, VOL=TSO\* identifies all volumes beginning with the characters TSO).

**Default:** VOL=\* (all online volumes)

**Format:** 1-6 characters

### **COST=**

**Function:** The cost of DASD storage in cents per megabyte per day. This value is used to calculate the value of used and unused DASD storage. Values from 1 to 999 can be specified.

**Default:** 10 (\$0.10 per megabyte per day)

**Format:** 1-3 numeric characters

**CURSIGN=**

Function: Currency symbol to be printed in the output report. The currency symbol can be one or two alphameric characters. If two characters are specified, the currency symbol is printed following the value (for example., 1000DM).

Default: \$ (US dollar)

Format: 1-2 characters

**LINES=**

Function: The number of lines per page to be printed on the report. Values from 5 to 999 can be specified.

Default: 50

Format: 1-3 numeric characters

**MAXFRAG=**

Function: The maximum value of the fragmentation index allowed before X37UTILC identifies the volume as a potential source of space errors. Fragmentation index is a measure of the number of small, unused free space areas on a volume. A large value for the fragmentation index is often associated with poor device utilization and potential space errors.

Default: 100

Format: 1-3 numeric characters

**MINDSCBS=**

Function: The minimum number of free (format-0) DSCBs that must be present on a volume before X37UTILC identifies the volume as a potential source of space errors. When there are few free DSCBs on a volume, new allocations to the volume may fail.

Default: 25

Format: 1-3 numeric characters

**MINUTIL=**

**Function:** The minimum percentage of space that must be occupied on a volume before X37UTILC identifies the volume as under-utilized. Volumes with less than this percentage of used space are wasting resources, and it may be desirable to increase the utilization of these volumes.

**Default:** 75

**Format:** 1-3 numeric characters

**DETAIL=**

**Function:** Specifies whether X37UTILC is to print a detailed report showing volume utilization for each volume processed. If DETAIL=NO is specified, only a summary report is printed.

**Default:** DETAIL=YES

**Format:** YES or NO

**Execution JCL**

This section describes the execution JCL used by the X37UTILC program.

<b>Statement</b>	<b>Usage</b>
JOB	Jobcard.
STEPLIB DD	The STEPLIB statement identifies the load library in which X37UTILC resides. Normally, this is the load library. The STEPLIB DD is not required if the load library is in the system linklist.
SYSPRINT DD	This statement identifies the output data set for the X37UTILC report. This DD statement is required.
SYSUT2 DD	This statement identifies an optional output data set for unformatted report data produced by X37UTILC. This data can be used as input to other programs which can sort or perform statistical operations. The DCB characteristics for this data set are: DSORG=PS,RECFM=FBA,LRECL=133. This DD statement is optional.
SYSIN DD	This statement identifies the input control statement data set for X37UTILC. This statement is optional. If not specified, X37UTILC examines all volumes. The DCB characteristics for this data set are: DSORG=PS,RECFM=FB,LRECL=80.

## Sample JCL

The following JCL can be modified to run X37UTILC. This JCL is in member JCLUTILC in *?prefix.BBSAMP*.

**Figure A-12 Sample JCL - X37UTILC**

---

```
//JOBNAME JOB (ACCOUNT)
//*****
// RUN X37UTILC
//*****
//STEP10 EXEC PGM=X37UTILC
//STEPLIB DD DSN=EMP.BBLINK,DISP=SH <== MAINVIEW SRM LOADLIB
//SYSPRINT DD SYSOUT=*
//SYSUT2 DD DSN=raw.data,DISP= <== VOLUME DATA
//SYSIN DD *
VOL=WORK*
VOL=(EMP001,EMP002)
/*
```

---

## Sample Output

The following control statements produce a report on DASD utilization for the following volumes: all CICS volumes and two user application volumes. The cost of DASD storage is estimated to be 12 cents per megabytes per day. At least 75 free DSCBs are required on each volume, and all volumes should be at least 85% utilized.

```
VOL=(CICS*,USER01,USER02),COST=12,MINDSCBS=75,MINUTIL=85
```

The resultant report follows.

**Figure A-13 Sample DASD Utilization Report**

```
-----  
StopX37/II RELEASE 7.2.1 DASD UTILIZATION REPORT          COPYRIGHT 2000 BMC SOFTWARE  
96.253 PAGE 001  
  
SYSIN:  
VOL=(CICS*,USER01,USER02),COST=12,MINDSCBS=75,MINUTIL=85  
  
OPTIONS IN EFFECT:  
VOLSER=CICS*  
    USER01  
    USER02  
COST=012  
LINES=054  
CURSIGN=$  
MINDSCBS=075  
MAXFRAG=100  
MINUTIL=085  
  
LEGEND FOR NOTES:
```

Figure A-14 Sample DASD Utilization Report

```

-----
-----
StopX37/II  RELEASE 7.2.1 DASD UTILIZATION REPORT          COPYRIGHT 2000 BMC SOFTWARE
96.253 PAGE 002

```

VOLSER	FRAG INDEX	PERCENT		MEGABYTES		VALUE (\$ )		NOTES
		USED	UNUSED	USED	UNUSED	USED	UNUSED	
CICS42	74	89	11	542	65	65.04	7.80	C
CICSR2	180	95	5	580	27	69.60	3.24	B,C
CICSAT	399	99	1	601	6	72.12	.72	B,C
CICS04	231	41	59	250	357	30.00	42.84	A,B,C
CICS02	145	45	55	279	328	33.48	39.36	A,B,C
CICS03	297	44	56	272	335	32.64	40.20	A,B
CICS01	225	37	63	230	377	27.60	45.24	A,B,C
CICS05	152	36	64	220	387	26.40	46.44	A,B,C
USER01	263	73	27	448	159	53.76	19.08	A,B
USER02	185	75	25	458	149	54.96	17.88	A,B

Figure A-15 Sample DASD Utilization Report Summary

```

StopX37/II  RELEASE 7.2.1 DASD UTILIZATION REPORT          COPYRIGHT 2003 BMC SOFTWARE
96.253 PAGE 003

```

```

*****
*** SUMMARY DATA ***
*****

```

VOLSER	FRAG INDEX	PERCENT		MEGABYTES		VALUE (\$ )	
		USED	UNUSED	USED	UNUSED	USED	UNUSED
TOTAL	215	63	36	3880	2190	465.60	262.80

10 VOLUMES WERE PROCESSED BY X37UTILC.  
7 VOLUMES HAD FEWER THAN 75 FREE DSCBS.  
9 VOLUMES HAD A FRAGMENTATION INDEX GREATER THAN 100.  
THESE VOLUMES MAY BE SUBJECT TO SPACE ABENDS.

7 VOLUMES WERE LESS THAN 85% UTILIZED AND ARE WASTING SPACE.  
OUT OF A TOTAL CAPACITY OF 6070 MEGABYTES OF DISK  
SPACE, 2190 MEGABYTES ARE UNUSED. THIS REPRESENTS  
A WASTE OF 36% ,OR \$ 262.80 PER DAY.

OVER ONE YEAR, THE TOTAL AMOUNT OF STORAGE  
WASTED WOULD BE WORTH 365 TIMES THIS AMOUNT, OR  
\$ 95,922.00.

IF THE OVERALL UTILIZATION WAS INCREASED TO  
85% , THEN \$ 153.48 COULD BE SAVED  
DAILY. OVER ONE YEAR, THE TOTAL AMOUNT SAVED  
WOULD BE \$ 56,020.20.

# Dynamic Multivolume Data Set Extent Consolidation (X37REORG)

The X37REORG utility works in conjunction with the StopX37/II SPACVOLA function and DFDSS to dynamically copy a data set that has been extended to multiple volumes by SPACVOLA processing back to a single volume. When SPACVOLA dynamically extends a data set to an additional volume, the REORG SPACVOLA RLST parameter is interrogated to see if the user specified REORG=YES. If REORG=YES was specified, SPACVOLA processing issues an OS/390 START command to start the started task indicated by the REORG\_PROC SPACVOLA RLST parameter.

This started task must consist of two steps. The first step is the X37REORG step, which performs dynamic variable substitution on the DFDSS control cards found in the MAINVIEW SRM parmlib member SMRORGxx (where xx is the member name suffix found in the first operand of the REORG\_NSMS or the REORG\_SMS SPACVOLA RLST parameter for non-SMS and SMS reorganizations respectively). The dynamic variable substitution replaces X37REORG variable names it finds in the DFDSS control card input with data specific to the reorganize job that is being processed. For example, SPACVOLA processing passes the data set name that needs to be reorganized as a parm on the OS/390 START command and the X37REORG facility replaces any occurrences of the variable %DS in the DFDSS control cards with the data set name that was passed.

The second step in the started task is the DFDSS step, which uses the updated control cards as input to copy the data set back to a single volume. The user has complete control over the DFDSS control cards and can use any DFDSS functions that are available for copying the data set. For example, it is up to the user's discretion if the data set is to be renamed during the copy operation or if the old copy of the data set is to be deleted. Any DFDSS functions can be used by this started task.

## Execution JCL

This section describes the execution JCL used by the X37REORG started task.

### Statement

### Usage

PROC

This statement must have the MEM, DS, IV, POOL, and STC keywords specified as started task parms.

EXEC	Specifies the program name for the X37REORG program (PGM=X37REORG). This statement must also include a PARM field which will pass the MEM, DS, IV, POOL, and STC parms to the X37REORG program.
STEPLIB DD	Defines the MAINVIEW SRM load library in which the X37REORG program resides.
SYSPRINT DD	Defines the output file for messages from the X37REORG program.
REORGDSN DD	Specifies the data set to be reorganized by including a DSN=&DS JCL keyword where the &DS is replaced by the passed started task keyword DS.
PROPARM	Specifies the MAINVIEW SRM parmlib, which contains the SMRORGxx member, which contains the DFDSS control cards. The parmlib data set specification must include the &MEM variable in the data set member specification. This started task JCL variable will be overridden with the correct SMRORGxx member name, which is passed by SPACVOLA on the OS/390 START command.
OUTPARM	Specifies the data set that will contain the updated DFDSS control cards, which are output by the X37REORG program. Each control card in the PROPARM DD data set is output to the OUTPARM DD after any X37REORG variables have been replaced. This data set is typically a temporary data set and is always used as the SYSIN DD data set for the subsequent DFDSS step.

### Sample JCL

The following JCL can be found in the X37REDSS member of *?prefix.BBSAMP*.

**Figure A-16 Sample JCL for X37REORG**

```

*****
* //X37REDSS  PROC MEM=,DS=,IV=,POOL=,STC=
* //X37REORG EXEC PGM=X37REORG,PARM='MEM=&MEM,IV=&IV,POOL=&POOL,STC=&STC,
* //          DS=&DS'
* //STEPLIB DD DSN=LLLLLLLLL,DISP=SHR <== REPLACE LLLLLLLL WITH
* //*          <== INSTALLATION MAINVIEW SRM LOADLIB
* //REORGDSN DD DSN=&DS,DISP=OLD
* //SYSPRINT DD SYSOUT=X
* //PROPARM DD DSN=PPPPPPPP(&MEM),DISP=SHR <== REPLACE PPPPPPPP WITH
* //*          <== INSTALLATION MAINVIEW SRM
* //*          <== PARMLIB
* //OUTPARM DD DSN=&&PARMIN,DISP=(,PASS),SPACE=(CYL,(1,1)),UNIT=SYSDA
* //*
* //*
* //X37RECPY EXEC PGM=ADRDSSU,COND=(4,LT,X37REORG),REGION=4M
* //SYSPRINT DD SYSOUT=X
* //SYSIN DD DSN=&&PARMIN,DISP=(OLD,DELETE)
*****

```

**Tip:** The REORGDSN DD disposition should be specified as OLD so the X37REORG job will not start until the job that caused the data set to extend to multiple volumes finishes.

The DFDSS SYSIN DD should always specify the same data set that is referenced by the OUTPARM DD in the X37REORG step.

The JCL variables MEM, DS, IV, POOL, and STC must be specified exactly as shown in the sample JCL including in the DSN keyword of the REORGDSN DD statement and in the member name of the PROPARM DD. These variables are automatically passed by RESOLVE SRM SPACVOLA processing as parameters on the OS/390 START command, which starts this started task.

## Parameters

The following parameters are passed on the OS/390 START command that starts the X37REORG job. The user is not responsible for setting any of these parameters. They are set and passed automatically by SPACVOLA processing. These parameters are used both for JCL keyword substitution in the X37REORG started task JCL, as well as for keyword substitution by the X37REORG program in the DFDSS control cards.

<b>Parameter</b>	<b>Usage</b>
DS=	This is the name of the data set that is to be reorganized. This parm is used in the started task JCL as the data set name on the REORGDSN DD. It is also used by the X37REORG program to substitute for all occurrences of the %DS variable found in the DFDSS control cards.
IV=	This is the input volume on which the data set resided prior to SPACVOLA adding a new volume for the data set. This value is substituted by the X37REORG program in the DFDSS control cards for any occurrence of the %IV variable.
MEM=	This is the member name of the member in the MAINVIEW SRM parmlib that contains the DFDSS control cards that are to be processed for variable substitution by the X37REORG program and then used as input to the DFDSS step of the reorganize task. This parm is used in the started task JCL as the member name on the PROPARM DD.
POOL=	This is the name of the MAINVIEW SRM pool that is used to generate a list of volumes that are eligible as target volumes for the reorganize job. The list of volumes in the specified MAINVIEW SRM pool are substituted by the X37REORG program in the DFDSS control cards for any occurrence of the %POOL variable.
STC=	This is the name of the DFSMS Storage Class that is used as the target storage class for the reorganize job (SMS-managed data sets only). This value is substituted by the X37REORG program in the DFDSS control cards for any occurrence of the %STC variable.

### **DFDSS Control Card Variables**

The following variables can be placed in the DFDSS control cards and will be dynamically substituted by the X37REORG program. The control cards must be in an SMRORGxx member of the MAINVIEW SRM parmlib and the MAINVIEW SRM parmlib and the SMRORGxx member must be referenced by the PROPARM DD in the X37REORG started task JCL. The real-time values for these variables are passed to the X37REORG started task as parms on the OS/390 START command and then passed to the X37REORG step in the EXEC JCL statement PARM field.

<b>Variable</b>	<b>Usage</b>
%DS	This is the name of the data set that is to be reorganized. This value is substituted by the X37REORG program in the DFDSS control cards for any occurrence of the %DS variable.

%IV	This is the input volume on which the data set resided prior to SPACVOLA adding a new volume for the data set. This value is substituted by the X37REORG program in the DFDSS control cards for any occurrence of the %IV variable.
%POOL	This is the name of the MAINVIEW SRM pool that is used to generate a list of volumes that are eligible as target volumes for the reorganize job. The list of volumes in the specified MAINVIEW SRM pool are substituted by the X37REORG program in the DFDSS control cards for any occurrence of the %POOL variable.
%STC	This is the name of the DFSMS Storage Class that is used as the target storage class for the reorganize job (SMS-managed data sets only). This value is substituted by the X37REORG program in the DFDSS control cards for any occurrence of the %STC variable.

### Control Card Input

The following DFDSS control card sample can be found in the SMRORGNS member of the installation data set *?prefix.BBLINK*. It is important to remember that the user may code these control cards using any combination of DFDSS control cards and X37REORG variables as long as the resultant control card after X37REORG variable substitution is a valid DFDSS control card. For example, in the OUTDYNAM keyword of the following example the user could have combined the %POOL variable with some hardcoded volume serials if that would produce the desired resultant volume list.

**Figure A-17 DFDSS Control Card Input Sample**

```

*****
COPY DATASET (
  INCLUDE ( %DS
  ) )
LOGINDYNAM ( %IV )
OUTDYNAM ( %POOL )
DELETE
ALLMULTI
CATALOG
ALLDATA ( * )
ALLEXCP
*****

```

## Data Set Reorganization (REDUCEXT)

As a part of the StopX37/II recovery process, occasionally it is necessary to add a second or subsequent volume to save the job from abending. This is referred to as a volume switch. Often, you would like a simple method for reorganizing the multi-volume data sets back to a single volume.

The included code is one way to accomplish this using IBM's IDCAMS DCOLLECT and DFDSS control cards. The code was written by a BMC Software customer and is provided to you as a service aid.

The goal is to automate the process of consolidating extents using a single job. This job examines your DASD farm and consolidates appropriate files with multiple extents. IDCAMS DCOLLECT is used to determine which files are candidates for extent reduction and then DF/DSS is called to do the actual reduction. DCOLLECT is used because of the speed it provides in collecting the required data. The Assembler program, called REDUCEXT, was written for a DFSMS environment, but is applicable for non-DFSMS data sets as well.

A different approach is used for SMS and non-SMS managed data sets. Only one DFDSS copy is issued for SMS managed data sets. This consolidates extents and moves the data set to a different DFSMS managed volume, appropriate for the data set. For non-SMS managed data sets, the data set is first moved to a work volume (specified by the WK=parameter), where the extents are consolidated. By default, this is a temporary location, and a second copy is issued to return the data set to the originating volume. If there is no reason to return the data set to the original volume, the ONEWAY parameter can be specified. Use of the work volume is required for non-SMS data sets, but can be used for an SMS managed data set also. This might be needed if there is only one volume in the storage group and no other storage group can be made available.

By default, DFDSS will not process system data sets with a high level qualifier of SYS1 for extent reduction. If you choose to include these data sets, a parameter of SYS1 can be coded to allow this. Using the SIM parameter will allow you to get a list of the candidate data sets for extent reduction prior to executing. A similar capability is offered with the NORUN parameter, which generates COPY requests for the DFDSS NORUN mode. This could be used to identify data sets that are ineligible for processing because of outstanding ENQEs.

## Sample JCL

The REDUCEXT program is included in *?prefix.BBSAMP*. The program must be assembled and linked prior to use. Sample JCL for running the program is included in the program comments. This user-written program is distributed by BMC Software as a service to their customers.

## Parameters

REDUCEXT accepts the following parameters to maximize flexibility:

Parameter	Usage
VL=	Indicates the volumes to be included for processing. An asterisk (*) can be used for pattern masking. An asterisk indicates that any characters after the asterisk can be ignored.
SG=	Indicates the storage groups to be included for processing.  <b>Note:</b> If neither VL nor SG is coded, the default is to process all volumes.
NX=	Indicates the number of extents used before a data set is a candidate for reorganization. The default is 16, the number of extents OS/390 allows per volume, which indicates the data set is multivolume. If reorganization is desired for less extents, a lesser number can be specified. Any number over 16 will not be accepted. Numbers 1 through 9 should be prefixed with zero and entered as 01, 02, and so on.  <b>Note:</b> Any non-VSAM data set with 16 extents or more will be eligible for extent reduction, independent of whether StopX37/II was responsible for the added volume.
WK=	For non-SMS managed data sets, you must specify the name of a work volume. This volume may also be used for SMS managed volumes, but is not required.
SIM	Displays the DFDSS control cards that are generated but no actions are executed. This can be used when there is a need to determine which data sets will be processed prior to actual execution.
NORUN	DFDSS will be called in NORUN mode. This can be used when there is a need to determine which data sets will be processed prior to actual execution.
ONEWAY	Non-SMS data sets will not be copied back to the originating volume, but will reside on the volume specified on the WK= parameter.

**SYS1** Specifying this parameter allows processing of SYS1 data sets. By default, SYS1 data sets will be ignored.

**Note:** This program was originally published in Issue 70 (July 1992) of Xephon's MVS Update, a monthly journal for OS/390 technicians, and is used with their permission. For further information on MVS Update, contact Xephon through their North American Office in Oviedo, Florida at (407) 366-8751.

**Figure A-18 Sample JCL to Run REDUCEXT**

---

```
//          JOB   job info
//STEP1    EXEC  PGM=REDUCEXT,
//          PARM=('NX=16',other parms....)
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  UNIT=VIO,SPACE=(CYL,(10,1))
//DCOLLECT DD  UNIT=VIO,SPACE=(CYL,(10,1))
//          DCB= (LRECL=264,RECFM=VB,DSORG=PS)
//DISPLAY  DD  SYSOUT=*
```

---

---

# Index

## A

- adding volume for secondary allocation 3-64
- ALCTYPE 3-52, 3-53
- ALTPOOL 3-65
- AVL 3-52, 3-53

## B

- BLKSIZE 3-17
- blocksize 3-16
  - optimizing for data sets 3-16

## C

- change allocation units to blocks 3-21
- CICS 2-12
- clusters
  - key-sequenced, space reduction rules 2-9
- CONTIG 3-52, 3-54
- control VIO allocation 3-78
- conversion to blocks 3-21

## D

- DASD Utilization Report
  - X37UTILC A-19
- Data Set Reorganization
  - REDUCEXT A-30
- data sets
  - allocating with optimal blocksize 3-16

- limiting size 3-52
- non-VSAM
  - reducing secondary space of 3-39
  - reducing secondary space to largest available extent 3-47
- release space at close 3-32
- secondary space value for allocation 3-35, 3-42
- sequential, additional volumes 3-64
- size 3-32, 3-52
- temporary 1-2, 3-52, 3-78
  - VIO allocation of 3-78

- data striping
  - considerations 2-15
- device 3-21, 3-52
- DF/SORT 2-11, 2-18
- DFSMS enhancements
  - OPTBLKSZ, allocate with optimal data set blocksize 3-16
  - SPACVOLA, add volume for secondary allocation 3-64
- DIR 3-52, 3-54
- Dynamic Multivolume Data Set Extent Consolidation
  - X37REORG A-25

## E

- end-of-volume 2-12
- ENVIR 3-27, 3-36, 3-40, 3-43, 3-49, 3-62, 3-70
- EOV 2-12

---

EVENTID 3-6, 3-7, 3-12, 3-17, 3-21, 3-26,  
3-27, 3-32, 3-35, 3-39, 3-42, 3-47, 3-48, 3-52,  
3-60, 3-65, 3-66, 3-78  
EXCP 2-10, 3-74

## F

FILESEQ 3-27, 3-36, 3-40, 3-43, 3-49, 3-62,  
3-70  
FORCE 3-17, 3-18  
functions  
    DFSMS enhancements  
        OPTBLKSZ 3-16  
    space management  
        OPTBLKSZ 3-16  
functions, DFSMS enhancements  
    SPACVOLA 3-64  
functions, space management  
    SPACCONV 3-21  
    SPACPRIM 3-25  
    SPACRLSE 3-32  
    SPACSECA 3-35  
    SPACSECB 3-39  
    SPACSECI 3-42  
    SPACSECR 3-47  
    SPACSQTY 3-52  
    SPACSWIR 3-60  
    SPACVOLA 3-64  
    VIOALLOC 3-78

## G

GDGVER 3-27, 3-36, 3-40, 3-43, 3-49, 3-62,  
3-70  
generic name 3-69

## I

initialize empty data sets 3-12  
ISAM 3-18

## L

LABELTYP 3-27, 3-36, 3-40, 3-43, 3-49, 3-62,  
3-70

## M

mass storage restrictions 2-10  
maximum volumes 3-65  
MNTYPE 3-65, 3-66, 3-72  
mount type 3-66, 3-72  
    parameter 3-65  
MSS restrictions 2-10

## N

NO CATLG2  
    bypassing processing 3-5  
NOCATLG2 3-1, 3-6, 3-7  
    considerations 3-4  
    preventing NOT CATLG2 errors 3-3  
    SMS-managed data sets 3-6  
NOCATWHEN 3-7, 3-8  
NOCHECK 3-35, 3-36, 3-47, 3-48, 3-60, 3-61,  
3-65, 3-67  
nonspecific public requests 3-26  
nonspecific storage requests 3-26  
NOTE=POINT 3-74

## O

OPENEMPT 3-12  
OPER 3-65, 3-67  
OPTBLKSZ 3-16  
optimum blocksize 3-16

## P

parmlib data set members 2-23  
PCTI 3-42, 3-43, 3-65, 3-67  
PLT  
    background 2-13  
POINT 3-74  
PQTY 3-52, 3-54  
preventing NOT CATLG 2 errors 3-3  
primary space allocation reduction 3-25  
PURGE 3-7, 3-8

---

## R

Record-Level-Sharing 2-12  
REDUCEXT A-2

- data set reorganization A-30
- parameters A-31
- sample JCL A-31

reducing primary space allocation 3-25  
reducing secondary allocation to best fit 3-39  
reducing secondary allocation to largest extent 3-47  
releasing data set space at close 3-32  
REORG 3-65, 3-68  
REORG\_NSMS 3-65, 3-68  
REORG\_PROC 3-65, 3-68, 3-69  
REORG\_SMS 3-65  
REPLACE 3-52, 3-55  
RLS 2-12

- installation steps 2-13

RLSE 3-32, 3-33  
ROUND 3-21, 3-22, 3-53, 3-55

## S

SAS

- compatibility with SPACSECR 2-4
- compatibility with SPACVOLA 2-4
- compatibility with StopX37/II 2-4

secondary allocation quantity

- add 3-35
- increase 3-42

secondary space

- increasing value for non-VSAM data sets 3-42
- reduction to best fit size 3-39
- supplying value for data set allocation 3-35

sequential data sets

- supplying additional volumes 3-64

setting optimum blocksize 3-16  
setting primary and secondary space for data set 3-52  
SMF analysis program

- SMFX37A A-2

SMFX37A A-2

- execution JCL A-3
- parameters A-4
- program logic A-2

sample JCL A-3  
sample output A-6  
SMF analysis program A-2  
SMFX37B A-2

- additional run-time options A-11
- analysis program A-7
- execution JCL A-8
- parameters A-9
- sample JCL A-13
- sample output A-15

SMS

- considerations 2-14

SORT

- limitations 2-11

SPACCONV 3-21  
space management

- OPTBLKSZ, allocate with optimal blocksize 3-16
- SPACCONV, changing allocation units to blocks 3-21
- SPACPRIM, reducing primary space allocation 3-25
- SPACRLSE, releasing data set space automatically 3-32
- SPACSECA, supplying secondary space value for data set allocation 3-35
- SPACSECB, reducing second space to best fit 3-39
- SPACSECI, increasing secondary space value 3-42
- SPACSECR, reducing secondary space to largest available extent 3-47
- SPACSQTY, setting primary and secondary space for data sets 3-52
- SPACSWIR, reducing initial allocation on volume add 3-60
- SPACVOLA, adding volume for secondary allocation 3-64
- VIOALLAC, allocating temporary data sets on VIO 3-78

space reduction

- rules for key-sequenced clusters 2-9

SPACPRIM 3-25, 3-26, 3-27  
SPACRLSE 3-32  
SPACSECA 3-35, 3-36  
SPACSECB 3-39, 3-40  
SPACSECI 3-42, 3-43  
SPACSECR 3-47, 3-48

---

- compatibility with SAS 2-4
- SPACSQTY 3-52
- SPACSWIR 3-60, 3-61
- SPACVOLA 3-29, 3-61, 3-64, 3-65, 3-69
  - compatibility with SAS 2-4
- specific storage requests 3-26
- SQTY 3-53, 3-56
- StopX37/II 1-1
  - bypassing critical data sets 2-11
  - compatibility with SAS 2-4
  - data striping considerations 2-15
  - features 1-1
  - filter and rule list sample 2-17
  - functions summary 2-3
  - nonrecovery conditions 2-10
  - processing recommendations 2-19
  - processing restrictions 2-19
  - SMS considerations 2-14
  - stand-alone usage 2-23
  - support 2-19
  - tailoring 2-6
  - VSAM considerations 2-7
- StopX37/II Analysis Program
  - SMFX37B A-7
- StopX37/II functions
  - NOCATLG2 3-3
  - SPACCONV 3-21
  - SPACSECA 3-35
  - SPACSQTY 3-52
  - VIOALLOC 3-78
- storage requests
  - nonspecific 3-26
  - specific 3-26
- SYNCSORT 2-11, 2-18

## T

- tailoring
  - StopX37/II 2-6
- temporary data sets 1-2, 3-52, 3-78
- TRKCYL 3-21, 3-22
- TRKLEN 3-21, 3-22
- TSO 3-11

## U

- UNCATLG 3-4
- UNIT 3-65, 3-69
- USECPOOL 3-65, 3-70

## V

- VALUE 3-27, 3-36, 3-40, 3-43, 3-49, 3-62, 3-70
- VCOMPLLQ 3-27, 3-36, 3-40, 3-43, 3-49, 3-62, 3-70
- VIO
  - allocation of temporary data sets 3-78
  - parameter 3-78, 3-79
  - restrictions 2-10
- VIOALLOC 3-78
- VOLSER 3-65, 3-70
- volume selection 3-16, 3-17
- volumes
  - count limitation 3-72
  - overriding default checks when adding 2-10
  - sequential data sets 3-64
- VSAM 3-9, 3-13, 3-18, 3-73
  - primary space reduction facility 2-8
- VSAM RLS
  - installation steps 2-13
- VSAMZSEC 3-35

## X

- X37PARMS 2-23
- X37REORG A-2
  - control card input A-29
  - DFDSS control card variables A-28
  - dynamic multivolume data set extent
    - consolidation A-25
  - execution JCL A-25
  - parameters A-27
  - sample JCL A-26
- X37RLS= 2-13
- X37UTILC A-2
  - control statements A-19
  - DASD utilization A-19
  - execution JCL A-21
  - sample JCL A-22
  - sample output A-22

---

XFCVSDS exit 2-12



# 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 by 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.**

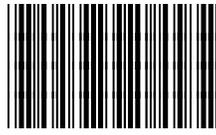
**U.S. GOVERNMENT RESTRICTED RIGHTS.** The Software under this Agreement is "commercial computer software" as that term is described in 48 C.F.R. 252.227-7014(a)(1). If acquired by or on behalf of a civilian agency, the U.S. Government acquires this commercial computer software and/or commercial computer 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



\*27839\*