

BMC Software Subsystem User Guide

**BMC Software Consolidated Subsystem
BMC Software Primary Subsystem**

Version 1.1

June 15, 2001



Copyright © 1995–2001 BMC Software, Inc., as an unpublished work.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 is a registered trademark of International Business Machines Corp. All other registered trademarks or trademarks belong to their respective companies.

The BMC Software Consolidated Subsystem (BCSS) and BMC Software Primary Subsystem (BMCP) technologies hold U.S Patent Number 5,566,334.

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 UNPUBLISHED–RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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. Send any contract notices to Contractor/Manufacturer:

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

Contacting BMC Software

You can access the BMC Software Web site at <http://www.bmc.com>. From this Web site, you can obtain general information about the company, its products, special events, and career opportunities. For a complete list of all BMC Software offices and locations, go to <http://www.bmc.com/corporate/offices.html>.

USA 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 USA and Canada

Telephone (01) 713 918 8800

Fax (01) 713 918 8000

Customer Support

You can obtain technical support by using Response Online™ (comprehensive information from the Web) or Response On Demand™. To expedite your inquiry, please see “Before Contacting BMC Software,” below.

Response Online

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

Response On Demand

In the USA and Canada, if you need technical support and do not have access to the Web, call 800 537 1813. Outside the USA and Canada, please contact your local support center or your local sales office for assistance.

Before Contacting BMC Software

Before you contact BMC Software, have the following information available so that a technical support analyst 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 program temporary fix (PTF)
 - system hardware configuration
 - serial numbers
 - related software (database, application, and communication) including type, version, and service pack or PTF
- 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
 - messages from related software

Contents

About This Book	xi	
Chapter 1	Introduction	
	Overview	1-2
	Extended Compression Architecture	1-2
	BMCP	1-3
	BCSS	1-4
	Multiple BCSSs	1-4
	BCSS Commands Data Set	1-5
	MVS Command Support	1-5
	Subsystem Naming Conventions	1-6
	Subsystem Procedure Names	1-6
	Types of Procedure Libraries	1-6
	Registration Data Set DD Statements	1-7
	Engine Level Mask	1-8
Chapter 2	Subsystem Operations	
	BMCP Command Tasks	2-2
	Command Summary	2-2
	BCSS Operator Console Command Tasks	2-6
	Command Summary	2-6
	Engine Level Mask	2-12
	CAS Command Tasks	2-13
Chapter 3	Controlling the BMC Software Subsystems by Using Start-up Parameters	
	Types of Parameters	3-2
	Specifying BMC Software Subsystem Parameters	3-2
	BMC Software Primary Subsystem Parameters	3-2
	BMC Software Consolidated Subsystem Parameters	3-4

Chapter 4	REGISET Concepts	
	REGISET Overview	4-2
	REGISET Data Sets	4-2
	Registration Data Sets	4-3
	Primary Registration Data Sets	4-4
	Duplex Registration Data Sets	4-4
	REGISET Cache	4-5
	Registration Data Set Failures	4-5
	Registration Data Set Recovery	4-6
	Recovery Tasks	4-7
Chapter 5	Utilities and Programs	
	Extract Utility	5-2
	Installation Check Utility	5-9
	JCL Requirements	5-11
	Parameter Options	5-12
	REGISET Cleanup Utility	5-13
	Registration Data Set Records	5-18
	Fixed Section—Registration Header Record	5-19
	Data Set Control Record Block Section	5-20
	Program Control Record Block Section	5-24
	Statistical Section	5-27
Chapter 6	BMC Software Subsystem Messages	
	Parts of a Message	6-2
	Components that Generate Messages	6-2
	Description Format	6-3
	Message Levels	6-3
	Contacting BMC Software Product Support	6-4
	BMC10000–BMC10999	6-5
	BMC11000–BMC11999	6-16
Index		

Figures

Figure 5-3	PARM Diagram	5-8
Figure 5-6	Sections of REGISET Record	5-18
Figure 6-1	Parts of a BMC Software Subsystem Message	6-2

Tables

Table 2-1	BMC Primary Subsystem (BMCP) Operator Commands Summary	2-2
Table 2-2	BMC Consolidated Subsystem (BCSS) General Operator Commands Summary	2-6
Table 2-3	Compression Address Space (CAS) Operator Commands Summary . . .	2-13
Table 5-1	Installation Check Utility Parameter Options	5-12
Table 5-2	REGISET Record Fixed Section	5-19
Table 5-3	REGISET Record DCRB Section	5-20
Table 5-4	REGISET Record PCRB Section	5-24
Table 5-5	REGISET Record Statistical Section	5-27
Table 6-1	Components that Generate BMC Software Subsystem Messages	6-2
Table 6-2	BMC Software Subsystem Message Levels	6-3

About This Book

This book contains detailed information about the BMC Software Subsystem and provides step-by-step installation instructions.

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

- your database management system (DBMS)
- Multiple Virtual Storage (MVS) systems, job control language (JCL), and the Interactive System Productivity Facility (ISPF)
- your client and host operating systems

For example, you should know how to respond to ISPF panels and how to perform common actions in a window environment (such as choosing menu items and resizing windows).

How This Book Is Organized

This book is organized as follows:

Chapter/Appendix	Description
Chapter 1, "Introduction"	explains what the BMC Software Subsystem is, what it does, and how it works
Chapter 2, "Subsystem Operations"	describes the BMC Software Primary Subsystem (BMCP) and the BMC Software Consolidated Subsystem (BCSS), the functions performed by these subsystems, and the commands used in each subsystem
Chapter 3, "Controlling the BMC Software Subsystems by Using Start-up Parameters"	describes the startup parameters and when to use them
Chapter 4, "REGISET Concepts"	provides essential information about the collection of registration data sets—the REGISET—that products use as a repository for control information
Chapter 5, "Utilities and Programs"	describes BMC Software Subsystem utilities and programs
Chapter 6, "BMC Software Subsystem Messages"	contains general information about the messages issued by the BMCP and the BCSS

In addition, an index appears at the end of the book.

Related Documentation

BMC Software products are supported by several types of documentation:

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

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.

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>.

Online Help

The BMC Software subsystem product includes online Help. In the BMC Software subsystem ISPF interface, you can access Help by pressing **F1** from any ISPF panel.

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 syntax statements.

General Conventions

This book uses the following general conventions:

Item	Example
information that you are instructed to type	Type SEARCH DB in the designated field. Type search db in the designated field. (Unix)
specific (standard) keyboard key names	Press Enter .
field names, text on a panel	Type the appropriate entry in the Command field.
directories, file names, Web addresses	The BMC Software home page is at www.bmc.com .
nonspecific key names, option names	Use the HELP function key. KEEPDICTIONARY option
MVS calls, commands, control statements, keywords, parameters, reserved words	Use the SEARCH command to find a particular object. The product generates the SQL TABLE statement next.
Unix commands, command options, database names	Use the sbacktrack program to create a backup script.
code examples, syntax statements, system messages, screen text	//STEPLIB DD The table <i>table_name</i> is not available.
emphasized words, new terms, variables	The instructions that you give to the software are called <i>commands</i> . In this message, the variable <i>file_name</i> represents the file that caused the error.

This book uses the following types of special text:

Note: Notes contain important information that you should consider.

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

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

Syntax Statements

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

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

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

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

Chapter 1 Introduction

This chapter explains what the BMC Software Subsystem is, what it does, and how it works. It also explains system requirements and compatibility with other products. The following sections are included:

Overview	1-2
Extended Compression Architecture	1-2
BMCP	1-3
BCSS	1-4
MVS Command Support	1-5
Subsystem Naming Conventions	1-6
Subsystem Procedure Names	1-6
Types of Procedure Libraries	1-6
Registration Data Set DD Statements	1-7
Engine Level Mask	1-8

Overview

The BMC Software Subsystem implements resource management facilities. A major advantage of using subsystem architecture is that it reduces virtual storage requirements in user address spaces. BMC Software Subsystem uses two BMC Software subsystems—BMC Software Primary Subsystem (BMCP) and BMC Software Consolidated Subsystem (BCSS).

Extended Compression Architecture

BMC Software Subsystem employs BMC Software's exclusive Extended Compression Architecture[®] (XCA[®]). XCA is a multiple address space subsystem architecture that follows current architectural trends used by IBM.

XCA has these components:

- BCSS—the XCA control facility that directs the other components
- set of compression techniques—Dynamic Adaptive, Arithmetic Adaptive, Extended Character, Static Huffman, Repeating Character, Hardware, and Custom Dictionary for Hardware
- REGISSET—a set of VSAM data sets that contain registration and control information for BMC Software products

Note: Not all components are used for all BMC Software products.

XCA provides the following benefits:

- BCSS provides commonly used services for the BMC Software products that use XCA. The BCSS also allocates compression routines in local address space (local compression) or a compression address space (CAS compression). Multiple jobs can use a single CAS; however, the BCSS creates and supports more compression address spaces when needed. The BCSS also maintains the registration data sets.
- XCA can be shared among many BMC Software products. This reduces the burden on your system when you have more than one BMC Software product that uses XCA.
- XCA provides resource management for itself and the BMC Software products that use XCA. When a product no longer needs a resource, XCA frees the resource.
- virtual storage requirements in local and CAS compression are reduced

If you choose local compression, all compression routines and work areas are located above the 16 MB line at the top of the extended private area. If you choose CAS compression, the compression routines and work areas are isolated in a separate address space.

- compression routines are downward compatible with previous releases

When applying changes or enhancements to the compression routines, the new techniques can expand data compressed with previously selected techniques.

BMCP

BMCP establishes supervisory services for BCSS and many BMC Software products. Because BMC Software products share BMCP, you need only one copy of BMCP on your system. All associated BMC Software products continue to operate normally even if BMCP terminates; however, BMC Software recommends that you leave BMCP running at all times.

BCSS

BCSS manages APF authorized functions and performs the processing for intercepted MVS requests. The requests that are intercepted are dependent on the BMC products utilizing BCSS services. BCSS also manages I/O to the registration data sets (collectively referred to as REGISET).

I/O to the REGISET is required anytime a product must store information. The BCSS utilizes a Dataspace Address Space to cache information stored in the REGISETS in order to reduce the I/O to the actual data set.

For products that utilize compression, BCSS creates and manages one or more compression address spaces (CAS). When a product requests a CAS, BCSS assigns a CAS to the request. BCSS creates a new CAS dynamically when an existing CAS cannot process new compression or expansion requests.

When an application closes a compressed data set, BMC Software Consolidated Subsystem sends a TERM message to the CAS. The TERM message informs the CAS that compression services are no longer needed. BMC Software Consolidated Subsystem then uses BCSS service routines to store compression statistics for the request in the primary REGISET. One CAS always remains in the system to process future compression and expansion requests.

BCSS must be active on the MVS system that will execute application programs that will use BMC products that require BCSS services.

Multiple BCSSs

More than one BCSS can be active on an MVS image. BMC products can share a BCSS (and the REGISET) with other BMC products or use a separate BCSS (and REGISET). The BMCP ensures that each BCSS receives control at the proper time for the products using that BCSS.

The decision to use one, or more than one, BCSS depends on how access to the REGISET will be managed. To isolate the I/O effects of the products, multiple BCSSs should be used. If the REGISET activity is not an issue, multiple products may use the same BCSS.

The MODE parameter must be used in the startup procedure to designate one (and only one) BCSS on an MVS image as the public BCSS. All others must be designated as private subsystems. The BMCP allows the public BCSS first access to an intercepted request.

BCSS Commands Data Set

The BCSS commands data set is a storage data set for BCSS and non-BCSS commands that need to be issued automatically after BCSS startup. You create this data set during product installation. The BCSS commands data set must be a sequential data set or a member of a partitioned data set (PDS). The name of this data set must appear on the COMMANDS DD statement in the BCSS startup JCL.

You can edit the BCSS commands data set to add or delete BCSS and non-BCSS commands. See “MVS Command Support” on page 1-5 for an explanation of the specification and the use of non-BCSS commands in the BCSS commands data set.

When entering BCSS commands into the BCSS commands data set, omit the BCSS ID prefix (*bcssid*) from each command. For example, the *bcssid* DAC CL LOCAL command appears in the BCSS commands data set as DAC CL LOCAL. You must start the commands in column one, and you must enter only one command per line. Commands cannot be continued onto another line. Place an asterisk (*) in column one for comment lines.

MVS Command Support

You can insert non-BCSS commands, including MVS console commands, in the BCSS commands data set. After the BCSS has completely initialized and is prepared for execution, non-BCSS commands execute sequentially.

You must prefix each MVS command that appears in the BCSS commands data set with *MVS*. For example, to start JES2 initiators 1 through 20 after BCSS initialization completes, you would add the *MVS* \$SI1-20 command to the BCSS commands data set. The JES command executes immediately after BCSS initialization. This ensures that no initiated tasks referencing compressed data begin executing before the BCSS has initialized.

Subsystem Naming Conventions

You must use a one- to four-character name to identify BMCP and BCSS to MVS. The installation procedure uses the default names *BMCP* and *BCSS*. If either name conflicts with a non-BMC Software subsystem installed on your system, you may have to use the *SUBSYSID* parameter in the BMCP or BCSS start-up procedure to change the subsystem names. For details about the *SUBSYSID* parameter, see the *Subsystem Installation Section*.

If you change the names of BMCP or BCSS after the subsystem starts, you must perform an IPL.

Note: This manual uses *bmcpid* and *bcssid* in all command samples to indicate where you would identify a specific BMCP subsystem or BCSS subsystem by its one- to four-character ID.

Subsystem Procedure Names

The Install System tailors two subsystem procedures, one to start the BMCP (the default procedure name *BMCP*) and one to start the BCSS (default procedure name *BMCBCSS*). Each procedure contains the required parameter and statements for the subsystem. If either default name conflicts with the name of another procedure member, you can use a different member name during installation.

Types of Procedure Libraries

For ease of use, execute the started tasks for the BMCP and the BCSS from the JES procedure library that is shared by all MVS images in your environment. Use the following subsystem command to start the subsystem, where *ssid* is the subsystem ID:

```
S ssid
```

SYS1.PROCLIB is usually not a shared procedure library. When the procedure name is the same as the subsystem ID (as they are if you use the installation defaults for the BMCP) and the procedure is in a non-shared library (such as the SYS1.PROCLIB), MVS runs the subsystem under the master scheduler. After the first attempt to start the procedure, any subsequent attempts to start it will result in an error (procedure not found) unless you specify the SUB=jesid parameter on the subsystem start command, where the *jesid* is the JES subsystem ID. For example, use the following command to start the BMCP if your JES subsystem ID is JES2:

```
S  BMCP , SUB=JES2
```

Registration Data Set DD Statements

The BCSS procedure contains DD statements that identify the registration data sets for the BCSS to use. The Install System automatically tailors these DD statements in the BCSS procedure to use the registration data set information you provide during installation.

A BCSS must use at least two registration data sets. The following combinations are valid:

- two (or more) primary data sets
- one primary data set and one (or more) duplex data sets
- one (or more) primary data sets and one (or more) duplex data sets

For more information about the registration data sets, see Chapter 4, “REGISET Concepts”

REGISET DD statements identify primary registration data sets. The first primary data set uses the ddname REGISET, a secondary primary data set uses the ddname REGISET0, a third uses REGISET1, and so on through ddname REGISET9.

DUPLEX DD statements identify duplex registration data sets. The first duplex data set uses the ddname DUPLEX, a secondary duplex data set uses the ddname DUPLEX0, a third uses DUPLEX1, and so on through ddname DUPLEX9.

Two or more BCSSs can share a REGISET. For some BMC product functions and features, there may be some REGISET sharing considerations if BCSSs must share a REGISET. For more information about these functions and features, see the applicable product manual.

Engine Level Mask

The Engine Level Mask (ELM) value determines which compression techniques are available for use. BMC Software changes the ELM value when it introduces new compression engines and when it modifies existing compression engines. Not all modifications to existing compression engines require a change to the ELM value.

BMC Software provides a new ELM value when modifications to existing compression techniques require backward compatibility with the technique modified. For example, data compressed with the latest version of BMC Software Consolidated Subsystem may be incapable of being expanded by an older version of the product.

The most current ELM value is set in the BCSS commands data set samples that are shipped on the distribution tape.

Warning! When you install a new release of BMC Software Consolidated Subsystem and change the ELM value, data compressed by the new engine may not be expandable by older engines. If you are testing the new release in a shared DASD environment, do not compress data accessible from multiple systems using new or enhanced engines unless these engines are available on all systems. Otherwise, expansion of the data on systems using the old engines may not be possible.

You can set an ELM value at the product level or the BCSS level. The product-level command sets an ELM value that supersedes the BCSS value. BMC Software recommends that you set a product-level ELM value. The BCSS-level command provides a default ELM for BMC Software Consolidated Subsystem and other BMC products that use BCSS but that did not set a product-level ELM.

Chapter 2 Subsystem Operations

This chapter describes the BMC Primary Subsystem (BMCP) and the BMC Consolidated Subsystem (BCSS), the functions performed by these subsystems, and the commands used in each subsystem. It also describes the BCSS commands data set, how you can add commands to the BCSS commands data set, and optional security parameters that you can use with the commands data set. The following sections are included:

BMCP Command Tasks	2-2
Command Summary	2-2
Starting BMCP	2-3
Terminating BMCP	2-4
Displaying the Status of BMCP	2-5
BCSS Operator Console Command Tasks	2-6
Command Summary	2-6
Starting BCSS	2-7
Terminating BCSS	2-8
Displaying the Status of BCSS	2-9
Reinitialization of BMC Products	2-10
Setting the BCSS ELM	2-11
Engine Level Mask	2-12
CAS Command Tasks	2-13
Starting a CAS	2-14
Stopping a CAS	2-15
Draining a CAS	2-16
Restarting a CAS	2-17
Displaying the Status of a CAS	2-18
Performing an Orderly Shutdown Before IPL	2-19

BMCP Command Tasks

This section provides task descriptions for BMCP commands. You can place these commands in a `COMMNDxx` member of `SYS1.PARMLIB` so they are issued automatically during an IPL, in a BCSS commands data set so they are issued at BCSS startup, or you can issue them as needed from the operator console.

Note: Remember that commands issued from the operator console remain active until the next IPL, at which time the values in the BCSS commands data set are reinstated.

The following task descriptions are provided:

- “Starting BMCP” on page 2-3
- “Terminating BMCP” on page 2-4
- “Displaying the Status of BMCP” on page 2-5

Command Summary

The following table (Table 2-1) lists the subsystem commands that you can use to start, stop, and display the status of the subsystems for BMC products. For a complete description of these commands, see the indicated page.

Note: If you are not using the default values of BMCP and BCSS for the subsystems, replace the *bmcpid* or *bcssid* in each command with the identifier used at your site.

Table 2-1 BMC Primary Subsystem (BMCP) Operator Commands Summary

Command	Function	Security Entity	Security Access	See Page
S BMCP<,SUB=MSTR JES2 JES3>	start the BMCP			2-3
<i>bmcpid</i> SHUTDOWN	stop the BMCP	<i>bmcpid</i> .SHUTDOWN	CONTROL	2-4
<i>bmcpid</i> STATUS	display BMCP status	<i>bmcpid</i> .STATUS	READ	2-5

Starting BMCP

Summary: You must start BMCP before starting any applications. For normal operations, place this MVS START command in a COMMNDxx member of SYS1.PARMLIB so BMCP starts automatically during an IPL. To start a BMCP manually after you have terminated it (see “Terminating BMCP” on page 2-4), issue the START command from the operator console.

To start BMCP, complete the following steps:

Step 1 Required. Issue the following MVS START command:

```
START bmcp
```

This command starts BMCP. The procedure library indicates the member that contains the BMCP procedure is indicated by *bmc*p.

Step 2 Required. Check the system log to ensure the command completed successfully. BMCP responds with messages indicating its availability.

Step 3 Optional. Issue the MVS DISPLAY ACTIVE LIST (D A,L) command. MVS responds with the following information. An active BMCP appears in this list.

JOBS	M/S	TS	USERS	SYSAS	INITS	ACTIVE/MAX	VTAM
00004	00025		00150	00009	00013	00150/00250	
BMBCSS	BMBCSS	SSAS	NSW	S	BMCP	BMCP13	BPSMDSP0 NSW S

Note: After an IPL, the BMCP subsystem name is not defined to MVS unless it has been added to the subsystem Name Table. The MVS START *bmc*p command will wait until the primary subsystem has initialized, then proceed as normal. However, once the subsystem has been started, the subsystem name for the BMCP is defined to MVS. If the procedure name and the subsystem name are the same, special considerations are necessary to avoid PROC NOT FOUND conditions from MVS. To avoid the PROC NOT FOUND conditions from MVS, either ensure the BMCP procedure is in the system SYS1.PROCLIB data set or include the SUB=JESx parameter on the MVS START command.

Terminating BMCP

Summary: You should leave BMCP running at all times; however, you must terminate it to install maintenance modules. You must also terminate BMCP as part of an orderly shutdown of MVS before an IPL (see “Performing an Orderly Shutdown Before IPL” on page 2-19).

To manually terminate BMCP, complete the following steps:

Step 1 Required. Issue the following command:

```
bmcpid SHUTDOWN
```

This command terminates the BMCP address space.

Step 2 Required. Check the system log to ensure the command completed successfully. BMCP responds with messages indicating that termination was initiated and completed.

Displaying the Status of BMCP

Summary: You can display the status of BMCP to determine whether BMCP is active.

To display the status of BMCP, complete the following steps:

Step 1 Required. Issue the following command:

```
bmcpid STATUS
```

This command displays status information about BMCP.

Step 2 Required. Check the system log to ensure the command completed successfully. BMCP responds with messages indicating the status of the address space.

BCSS Operator Console Command Tasks

This section provides task descriptions for BCSS operator commands. You can place these commands in a `COMMNDxx` member of `SYS1.PARMLIB` so they are issued automatically during an IPL, in a BCSS commands data set so they are issued at BCSS startup, or you can issue them as needed from the operator console.

Note: Remember that commands issued from the operator console remain active until the next IPL, at which time the values in the BCSS commands data set are reinstated.

The following task descriptions are provided:

- “Starting BCSS” on page 2-7
- “Terminating BCSS” on page 2-8
- “Displaying the Status of BCSS” on page 2-9
- “Reinitialization of BMC Products” on page 2-10
- “Setting the BCSS ELM” on page 2-11

Command Summary

The following table lists the subsystem commands that you can use to start, stop, and display the status of the subsystems for BMC products. For a complete description of these commands, see the indicated page.

Note: If you are not using the default values of `BMCP` and `BCSS` for the subsystems, replace the `bmcpid` or `bcssid` in each command with the identifier used at your site.

Table 2-2 BMC Consolidated Subsystem (BCSS) General Operator Commands Summary

Command	Function	Security Entity	Security Access	See Page
<code>S BMBCSS<,OPTION=REFRESH NREFRESH></code>	start the BCSS			2-7
<code>bcssid SHUTDOWN</code>	stop the BCSS	<code>bcssid.SHUTDOWN</code>	CONTROL	2-8
<code>bcssid STATUS</code>	display BCSS status	<code>bcssid.STATUS</code>	READ	2-9
<code>bcssid REINIT xxx</code>	start/reinitialize BMC products	<code>bcssid.REINIT</code>	CONTROL	2-10
<code>bcssid ELM value</code>	set engine level mask	<code>bcssid.ELM</code>	CONTROL	2-11

Starting BCSS

Summary: You must start BCSS before starting any applications. For normal operations, place the MVS START command in a COMMNDxx member of SYS1.PARMLIB so BCSS starts automatically during an IPL. To start BCSS manually after you have terminated it (see “Terminating BCSS” on page 2-8), issue the START command from the operator console.

To start BCSS, complete the following tasks:

Step 1 Required. Issue one of the following commands:

- START *bmc bcss*
- START *bcss*,OPTION=REFRESH

This command starts BCSS with new copies of resident BCSS modules. *bcss* indicates the procedure library member that contains the BCSS procedure. OPTION=REFRESH is the default for BCSS startup.

- START *bcss*,OPTION=NREFRESH

This command starts BCSS with existing copies of resident BCSS modules. *bcss* indicates the name of the procedure library member containing the BCSS procedure.

Step 2 Required. Check the system log to ensure the command completed successfully. BCSS responds with messages indicating that it is available.

Step 3 Optional. Issue the MVS DISPLAY ACTIVE LIST (D A,L) command. MVS responds with the following information. An active BCSS appears in this list.

JOB	M/S	TS	USERS	SYSAS	INITS	ACTIVE/MAX	VTAM
00004	00025	00150		00009	00013	00150/00250	
LLA	LLA	LLA	NSW	S RMF	A	IEFPROC	NSW S
NET	NET	NETNCP	OWT	S TSO	TSO	STEP1	NSW S
BMCBCSS	BMCBCSS	SSAS	NSW	S BMCP	BMCP0013	BPSMDSP0	NSW S

Terminating BCSS

Summary: You should leave the BCSS running at all times; however, you must terminate it to install maintenance modules. You must also terminate BCSS as part of an orderly shutdown of MVS before an IPL (see “Performing an Orderly Shutdown Before IPL” on page 2-19). Existing CASs are not affected by terminating BCSS.

To terminate BCSS, complete the following steps:

Step 1 Required. Issue the following command:

```
bcssid SHUTDOWN
```

This command terminates the BCSS address space.

Step 2 Required. Check the system log to ensure the command completed successfully. BCSS responds with messages indicating that termination was initiated and completed.

Displaying the Status of BCSS

Summary: You can display the status of BCSS.

To display the status of BCSS, complete the following steps:

Step 1 Required. Issue the following command:

```
bcssid STATUS
```

This command displays status information about BCSS.

Step 2 Required. Check the system log to ensure the command completed successfully. BCSS responds with messages indicating the status of the address space.

Reinitialization of BMC Products

Summary: You must reinitialize BMC products after an IPL, after registration data set fails, or after a product abends. When you issue the reinitialization command, the BMC product determines which components are licensed and initializes only those components.

To reinitialize BMC products, complete the following steps:

Step 1 Required. Issue the following command:

```
bcssid REINIT xxx
```

This command reinitializes the BMC product with the identifier of *xxx*.

Step 2 Required. Check the system log to ensure the command completed successfully. BCSS responds with messages indicating that it has initialized the BMC product; status messages also are displayed.

Setting the BCSS ELM

Summary: You can specify a particular ELM for BCSS.

To set a BCSS-level ELM value, complete the following steps:

Step 1 Required. Issue the following command:

```
bcssid ELM nnnnnnnn
```

This command sets the BCSS-level ELM to *nnnnnnnn*.

Note: If you are running multiple BMC Software products that use BCSS, BMC Software recommends that you set the BCSS-level ELM to a value of 00000000 (the default value if you do not specify an ELM value).

Step 2 Required. Check the system log to ensure the command completed successfully. BCSS responds with messages indicating the engine level in effect.

Engine Level Mask

The Engine Level Mask (ELM) value determines which compression techniques are available for use by the compression product. BMC Software changes the ELM value when it introduces new compression engines and when it modifies existing compression engines. Not all modifications to existing compression engines require a change to the ELM value.

BMC Software provides a new ELM value when modifications to existing compression techniques require backward compatibility with the technique modified. For example, data compressed with the latest version of BMC Software Subsystem may be incapable of being expanded by an older version of the product.

The most current ELM value is set in the BCSS commands data set samples that are shipped on the distribution tape.

Note: When you install a new release of BMC Subsystem and change the ELM value, data compressed by the new engine may not be expandable by older engines. If you are testing the new release in a shared DASD environment, do not compress data accessible from multiple systems using new or enhanced engines unless these engines are available on all systems. Otherwise, expansion of the data on systems using the old engines may not be possible.

CAS Command Tasks

This section provides task descriptions for CAS commands. You can issue CAS commands through BCSS by placing them in a BCSS commands data set so they are issued at BCSS startup. You can also issue these commands as needed from the operator console.

Note: Remember that commands issued from the operator console remain active until the next IPL, at which time the values in the BCSS commands data set are reinstated.

The following task descriptions are provided:

- “Starting a CAS” on page 2-14
- “Stopping a CAS” on page 2-15
- “Draining a CAS” on page 2-16
- “Restarting a CAS” on page 2-17
- “Displaying the Status of a CAS” on page 2-18

Note: If you do not know the compression address space ID (CASID) for a CAS that you need to stop, drain, restart, or display status information, the CASID is the same as the hexadecimal MVS address space ID (ASID) for the address space where the CAS is running.

Table 2-3 Compression Address Space (CAS) Operator Commands Summary

Command	Function	Security Entity	Security Access	See Page
<i>bcssid</i> CAS START	start a CAS	<i>bcssid</i> .CAS	CONTROL	2-14
<i>bcssid</i> CAS STOP < <i>casid</i> ALL>	stop a CAS	<i>bcssid</i> .CAS	CONTROL	2-15
<i>bcssid</i> CAS DRAIN < <i>casid</i> ALL>	drain activity from a CAS	<i>bcssid</i> .CAS	CONTROL	2-16
<i>bcssid</i> CAS RESTART < <i>casid</i> ALL>	restart a drained CAS	<i>bcssid</i> .CAS	CONTROL	2-17
<i>bcssid</i> CAS STATUS < <i>casid</i> ALL>	display CAS status	<i>bcssid</i> .CAS	CONTROL	2-18

Starting a CAS

Summary: You can instruct BCSS to start a CAS only when a previous attempt to start a CAS by BCSS fails because of a JCL error.

To start a CAS, complete the following steps:

Step 1 Required. Issue the following command:

```
bcssid CAS START
```

This command starts a CAS.

Step 2 Required. Check the system log to ensure the command completed successfully. BCSS responds with messages indicating that it has started a CAS.

Stopping a CAS

Summary: You can stop (drain and end) a specific CAS or all CASs. You must stop all CASs after applying maintenance. This allows newly created CASs to use the new CAS procedure. You must also stop all CASs as part of an orderly shutdown of MVS before an IPL (see “Performing an Orderly Shutdown Before IPL” on page 2-19).

To stop a CAS, complete the following steps:

Step 1 Required. Issue one of the following commands:

- `bssid CAS STOP casid`

This command stops the CAS identified by *casid*.

- `bssid CAS STOP ALL`

This command stops all CASs. When you stop all CASs, the CAS address space ends and the CAS is in an ENDED state (see “Displaying the Status of a CAS” on page 2-18).

Step 2 Required. Check the system log to ensure the command completed successfully. BCSS responds with messages for each CAS that has stopped.

Draining a CAS

Summary: You can drain one or all CASs. Any jobs that are using the draining CAS continue using the CAS, but no new jobs are assigned to the draining CAS. BCSS stops draining CASs when their link count becomes zero; however, BCSS leaves one active CAS available in the system. If necessary, you can stop the last remaining CAS (see “Stopping a CAS” on page 2-15).

To drain a CAS, complete the following steps:

Step 1 Required. Issue one of the following commands:

- *bcssid CAS DRAIN casid*

This command drains the CAS identified by *casid*.

- *bcssid CAS DRAIN ALL*

This command drains all CASs.

Step 2 Required. Check the system log to ensure the command completed successfully. BCSS responds with messages for each CAS that is draining.

Restarting a CAS

Summary: You can restart one or all CASs after you have drained them (see “Draining a CAS” on page 2-16). You cannot restart a CAS that you have stopped (see “Stopping a CAS” on page 2-15). BCSS dynamically starts new CASs when it needs them.

To restart a CAS, complete the following steps:

Step 1 Required. Issue one of the following commands:

- `bcssid CAS RESTART casid`

This command restarts the drained CAS identified by *casid*.

- `bcssid CAS RESTART ALL`

This command restarts all drained CASs.

Step 2 Required. Check the system log to ensure the command completes successfully. BCSS responds with messages for each CAS that is restarted.

Displaying the Status of a CAS

Summary: You can display the status of a specific CAS or of all CASs.

To display the status of a CAS, complete the following steps:

Step 1 Required. Issue one of the following commands:

- *bcssid CAS STATUS casid*

This command displays the status of the CAS identified by *casid*.

- *bcssid CAS STATUS ALL*

This command displays the status of all CASs.

Step 2 Required. Check the system log to ensure the command completes successfully. BCSS responds with status messages for each active or drained CAS.

Performing an Orderly Shutdown Before IPL

Summary: To perform an orderly shutdown of the BMC Subsystem, you must stop the BCSS, CAS, and BMCP on each CPU.

To perform an orderly shutdown, complete the following steps:

Warning! Issuing the following commands affects all BMC Software products in your system that use the BCSS being shut down.

- Step 1** Required. Issue the *bcssid* CAS STOP ALL command to terminate all CASs.
- Step 2** Required. Issue the *bcssid* SHUTDOWN command to terminate BCSS processing.
- Step 3** Required. Issue the *bmcpid* SHUTDOWN command to terminate BMCP processing.

Chapter 3 **Controlling the BMC Software Subsystems by Using Start-up Parameters**

This chapter describes the BMC Software subsystem startup parameters and when you should use them. Each of the parameters has a default value that the BMC Software subsystem will use if you do not specify an alternate value. The defaults are shown in this chapter.

Types of Parameters	3-2
Specifying BMC Software Subsystem Parameters.	3-2
BMC Software Primary Subsystem Parameters.	3-2
BMC Software Consolidated Subsystem Parameters.	3-4

Types of Parameters

There are two types of parameters—required and optional. There is no difference in the processing between these options unless a required parameter is not specified or incorrectly specified then an operator PROMPT is issued regardless of the NOPROMPT setting. The BMC Software subsystem offers many parameters to tailor the operation and function of the subsystem. For example, you can cause a reload of executable modules in common storage to effect maintenance changes without an IPL.

Specifying BMC Software Subsystem Parameters.

Parameter values are selected by specifying them in the PARM field of the procedures JCL EXEC statement for the BMC Software subsystem start up, or by specifying them in the actual START command during BMC Software subsystem start up.

You can specify the parameters in any order. If a conflicting parameter is requested like PROMPT and NOPROMPT then the last (or rightmost) parameter is utilized.

If two or more parameters are specified the list of parameters must be enclosed within single quotation marks or parentheses. Commas must separate each parameter. The parameter list can be no longer than 100 characters including commas.

BMC Software Primary Subsystem Parameters

A description of each of the parameters that can be utilized to control the BMC Software Primary Subsystem (BMCP) follows. These parameters are provided either through execution parameters or an operator response. If this is a restart of the subsystem the previous value for SUBSYS, SVCNO, and SMFRECID will be unconditionally utilized. An IPL will be required to change these values. There can be only one primary subsystem active on an MVS system at any time. All consolidated subsystems will utilize the services of the BMCP.

REFRESH**(Default value)**

During subsystem startup modules currently loaded into common storage are to be checked for upgrades. If a different version of the module is found in the normal system search then the module will be reloaded into common storage. This option can be utilized to load new versions of common storage modules without the need for an IPL.

Note: Not all common storage modules can be upgraded by the use of the REFRESH parameter, in these cases BMC Software support will issue a statement stating that the upgrade requires an IPL.

DEBUG

Indicates that the subsystem should write out diagnostic trace information to the system logs. This should be used with caution, as console traffic could become extremely heavy with this parameter.

NODEBUG**(Default value)**

This is the normal operation mode of the BMC Software subsystems no diagnostic messages are issued. Only normal processing and error messages are issued in this mode.

PROMPT

The operator will be prompted to correct errors or provide required parameters if they are omitted from the parameter lists.

CAPS**(Default value)**

Console messages will be issued in all upper case letters. This mode will satisfy the CUA language support for console operations as defined by IBM.

MIXED

Console messages will be issued in mixed case mode. This will make the messages somewhat easier to read. International language sites may not be able to view the entire BMC Software message because of DBCS constraints.

SMFRECID**(0)**

For BMC products that request the BMCP to create a BMC Software common SMF record this will be the SMF record id utilized in the SMF records. This number **MUST** be a valid user SMF value as defined by IBM. This value is specified as a decimal range between 200 and 255 that should not be utilized by any other products.

SUBSYSID

(BMCP)

Specifies the subsystem name utilized during the startup and operation of the primary subsystem. The name must be a unique subsystem name on the MVS image and cannot exceed four characters in length with the first being alphabetic. Example: SUBSYSID=BMCP sets the subsystem name to BMCP.

SVCNO

Specifies the SVC number utilized by the primary subsystem to service MVS services intercepts. This value should only be changed when instructed to by BMC Software Support.

BMC Software Consolidated Subsystem Parameters

A description of each of the parameters that can be utilized to control the BMC Software Consolidated Subsystem (BCSS) follows.

REFRESH

(Default value)

During subsystem startup modules currently loaded into common storage are to be checked for upgrades. If a different version of the module is found in the normal system search then the module will be reloaded into common storage. This option can be utilized to load new versions of common storage modules without the need for an IPL.

Note: Not all common storage modules can be upgraded by the use of the REFRESH parameter, in these cases BMC Software support will issue a statement stating that the upgrade requires an IPL.

NOREFRESH

During subsystem startup, modules currently loaded into common storage will NOT be checked for upgrades.

DEBUG

Indicates that the subsystem should write out diagnostic trace information to the system logs. This should be used with caution, as console traffic could become extremely heavy with this parameter.

NODEBUG

(Default value)

This is the normal operation mode of the BMC Software subsystems no diagnostic messages are issued. Only normal processing and error messages are issued in this mode.

PROMPT (Default value)

The operator will be prompted to correct errors or provide required parameters if they are omitted from the parameter lists.

NOPROMPT

The operator will be prompted only in the event of a required parameter omission.

CAPS (Default value)

Console messages will be issued in all upper case letters. This mode will satisfy the CUA language support for console operations as defined by IBM.

MIXED

Console messages will be issued in mixed case mode. This will make the messages somewhat easier to read. International language sites may not be able to view the entire BMC Software message because of DBCS constraints.

SKIPSYN

During BMC Software subsystem startup the BCSS will skip the synchronization of the REGISSETS if the REGISSETS are currently being managed by any other active subsystem in a shared DASD environment and this is the first start of the BCSS during the current IPL. If there are no other BCSS subsystems managing the REGISSETS or this is a restart of the BCSS the files will be synchronized during startup. The impact of the synchronization RESERVE can be minimized by the use of this parameter. When both conditions are met the synchronization process is bypassed and message BMC11058I is issued.

FORCESYN (Default value)

During BMC Software subsystem startup the BCSS will synchronize the REGISSETS to insure the files are all validated. During the startup the BCSS will RESERVE the volume that the REGISSETS reside on and no other BCSS subsystem can access the REGISSETS until synchronization is completed.

PRIVATE (Default value)

The BCSS subsystem will load an independent set of control blocks and intercept routines. There is no checking for duplicate functionality. Those products that request this mode of operation should use this option. Additional BCSS subsystems can be started to test new code releases, however, care should be taken to insure that BCSS services are invoked only once per request.

PUBLIC

The BCSS subsystem will load a common set of control blocks and intercept routines. There can be only one PUBLIC BCSS subsystem. If more than one consolidated subsystem is started as PUBLIC an error message is issued and the start of the consolidated subsystem will fail. The PUBLIC subsystem will receive control from the BMC Software intercept routines first regardless of BMC Software subsystem start order. Some BMC Software products require that they be operational in only the PUBLIC subsystem.

SUBSYSID**(BCSS)**

Specifies the subsystem name utilized during the startup and operation of the consolidated subsystem. The name must be a unique subsystem name on the MVS image and cannot exceed four characters in length with the first being alphabetic. For every instance of the consolidated subsystem started within a system a unique name must be utilized. Example: SUBSYSID=BCSS sets the subsystem name to BCSS.

CASPROC**(NODEFAULT)**

The Compression Address Spaces (CAS) are internally managed by the consolidated subsystem. When required the consolidated subsystem will automatically start a CAS. This parameter notifies the consolidated subsystem of the PROC name for the CAS. This parameter is required and has no default value. Example: CASPROC=BMCCAS sets the CAS procedure names to BMCCAS.

DSPSIZE**(1)**

This parameter notifies the consolidated subsystem how large the auxiliary DATASPACE should be allocated in order to cache REGISSET entries. This value is specified as megabytes the default is 1 and is large enough to hold approximately 1,000 REGISSET entries. If the REGISSET is expected to be larger, then the DSPSIZE should be adjusted. The cache is managed in a LRU manner. Example: DSPSIZE=1 sets the DATASPACE size allocated to one megabyte.

NOREGISET

This parameter will specify the subsystem is to operate without a REGISET (registration datasets). During the startup of the subsystem, the initialization routines check to insure that all required datasets are available for use. If you plan to operate without any registration datasets this parameter will have the initialization routines bypass the checking for those required datasets. Most BMC Software products require the use of the REGISETS, products that require the use of REGISETS will not operate correctly without REGISET availability. The BMC products that can operate without the REGISET will specify this option to be utilized. For more information about REGISETS see Chapter 4, “REGISET Concepts.”

NODUPLEX

This parameter will specify the subsystem is to operate without a DUPLEX registration dataset. During the startup of the subsystem, the initialization routines check to insure that all required datasets are available for use. If you plan to operate with only PRIMARY registration dataset this parameter will have the initialization routines bypass the checking for the DUPLEX registration datasets. For more information about REGISETS see Chapter 4, “REGISET Concepts.”

Chapter 4 REGISET Concepts

This chapter provides essential information about the collection of registration data sets—the REGISET—that products use as a repository for control information. This chapter contains the following information:

REGISET Overview	4-2
REGISET Data Sets	4-2
Registration Data Sets	4-3
Primary Registration Data Sets	4-4
Duplex Registration Data Sets	4-4
REGISET Cache.	4-5
Registration Data Set Failures.	4-5
Registration Data Set Recovery	4-6
Recovery Tasks	4-7
Recovering a Primary Registration Data Set (Additional Primary Data Set)	4-8
Recovering a Primary Registration Data Set Using the Duplex Registration Data Set.	4-9
Recovering a Primary Registration Data Set (Backup Copy)	4-10
Recovering a Duplex Registration Data Set	4-11
Making Off-Site Backups Using BMC Software EXTRACT	4-12

REGISET Overview

This section explains the physical structure of the REGISET and how it is associated with a BMC Software Consolidated Subsystem (BCSS).

REGISET Data Sets

The registration data sets (collectively called REGISET) allow you to control when and how a BMC Software product performs its compression and expansion functions. The REGISET can be used by every address space active in the system. This will make the REGISET some of the highest-activity data sets within an MVS environment. Also, the BCSS uses standard MVS serialization techniques to protect the integrity of the information stored in the REGISET. These factors make the placement of the REGISET data sets very important.

To optimize system performance, BMC Software strongly recommends that the REGISET data sets be placed on volumes that do not contain any other high-activity data sets or any data sets that hold a frequent or prolonged RESERVE on the volume. Examples of high-activity data sets are catalogs, interactive databases, a JES2 checkpoint file, the Computer Associate's MIM control file, and other REGISET data sets. Load libraries are also considered high-activity data sets since link-edits require a prolonged RESERVE during the library update.

Prior to release 1.1.00 of the BCSS, the REGISET serialization process did not permit the RESERVE issued under the QNAME of BMCBCSS to be converted to a global ENQ by system utilities like GRS from IBM or MIMGR from Computer Associates. If the RESERVE was converted to a global ENQ, data integrity safeguards were circumvented and REGISET drop conditions developed. With BCSS 1.1.00, the REGISET serialization process was modified to permit the RESERVE to be converted to a global enqueue. Once all shared REGISET systems are operating at the release 1.1.00 level or higher, it becomes possible to convert the RESERVE to global ENQ without risking data integrity issues, thus reducing the possibility of developing REGISET drop conditions.

However, the RESERVEs issued to the REGISET are of such frequent and short duration that BMC Software recommends that this RESERVE not be converted. The overhead incurred by converting the RESERVE with a disk-based management utility has shown to be a negative impact to system performance. CTCA-based products do not exhibit this performance degradation.

The data sets in the REGISET are VSAM key-sequenced data sets (KSDSs). There are two types of data sets—primary and duplex. The duplex registration data sets are optional and maintain active backup copies of the primary registration data sets.

The REGISET can consist of as many as 22 data sets, 11 primary and 11 duplex. As you increase the number of data sets, however, the REGISET access time increases. The minimum number of data sets is two: one primary and one duplex, or two primaries.

Note: When an application opens a data set, the BMC Software product checks the REGISET. It uses the information it finds to determine if services are required. This procedure comprises the product's basic method for intercepting data sets. You can, however, bypass the REGISET by using a user control exit to intercept data sets.

Registration Data Sets

The REGISET can consist of as few as two (or as many as 22) registration data sets. These data sets are VSAM key-sequenced data sets (KSDSs) that are defined during subsystem installation. The registration data sets that a BCSS uses are identified in the BCSS procedure (default name BMCBCSS) with DD statements.

Registration data sets are of two types—primary and duplex. Primary and duplex registration data sets have identical structures. Whether a registration data set is treated as a primary or a duplex is determined by the ddnames in the BCSS procedure.

Depending on your recovery needs, you can define one of the following scenarios for your site:

- a single primary registration data set and multiple duplex registration data sets
- multiple primary registration data sets and no duplex registration data sets
- multiple primary registration data sets and multiple duplex registration data sets

Each registration data set must reside on a separate volume and cannot be defined as a multivolume data set.

Primary Registration Data Sets

BCSS uses the primary registration data sets when providing BMC Software products with responses to REGISET queries. You must have at least one primary registration data set; you can have as many as 11.

When you define more than one primary registration data set, you are using the Multiple Primary feature. If the active primary registration data set fails, a message is issued, indicating that the failing data set has been dropped from processing. Because an additional primary registration data set is available, normal processing continues, allowing you to recover the failing data set at a more convenient time. In addition, recovery of the failing data set is simplified because restoration is not necessary; you can define a valid primary registration data set.

When the BCSS updates a primary registration data set, it issues a hardware reserve against the volume; a primary registration data set should not reside on the same volume with another data set that may be reserved, such as an MVS catalog IMS RECON, or another primary registration data set. The shared reserve does not lock out access from the same MVS image but does lock out access from other different MVS images in a shared DASD configuration. Reserves are of short duration and occur only during update activity. The reserve is held until updates to all primary registration data sets and all duplex registration data sets are complete. Primary registration data sets are defined with SHAREOPTIONS(4,3).

Duplex Registration Data Sets

The duplex registration data sets maintain active backup copies of the primary registration data sets. Using this type of registration data set is optional, but you can define as many as 11 duplex registration data sets.

The duplex registration data sets are used only for recovering primary registration data sets. When BCSS updates a primary registration data set, it also updates the duplex registration data sets. It makes these updates one at a time while holding a RESERVE on the primary registration data sets. This ensures that, if a failure occurs in the only available primary registration data set, the duplex registration data sets are, at most, one update behind. Subsystems can continue to function if one primary registration data set is available. Duplex registration data sets are defined with SHAREOPTIONS(4,3).

If you encounter a failure in a primary registration data set and there are no additional primary registration data sets, the BCSS stops. You can restore a primary registration data set by using an additional copy of the primary registration data set, one of the duplex registration data sets, or a backup copy of the primary registration data set.

Although the BCSS does not issue a reserve against the volume that contains a duplex registration data set, do not place a duplex registration data set on the same volume with a primary registration data set.

REGISET Cache

The REGISET Cache feature substantially reduces the amount of I/O activity to the REGISET. This feature uses IBM Data Space technology and is available only in an MVS/ESA or above environment.

The cache size is variable and is controlled by the DSPSIZE parameter in the BCSS procedure (default name BMCBCSS). You can specify 1 MB to 2047 MB for DSPSIZE.

Registration Data Set Failures

If the BCSS detects a REGISET failure, it issues a BMC Software message indicating that a registration data set (primary or duplex) has failed (dropped) and requires recovery. System programmers should recover the registration data set(s). If you are using multiple primary registration data sets, immediate recovery is not necessary as long as additional primary registration data sets are available for recovery purposes.

The purpose of the duplex registration data sets is for recovery processing. When BCSS updates a primary registration data set, it also updates the duplex registration data sets. If a failure occurs in the only available primary registration data set, this process ensures that the duplex registration data sets are, at most, one update behind. Subsystems can continue to function if one primary registration data set is available.

Registration Data Set Recovery

This section provides instructions for recovering a primary or a duplex registration data set. Because the information in registration data sets changes constantly, a system backup of a registration data set is rarely current enough for recovery processing. When possible, use a duplex registration data set rather than a backup copy to recover a lost primary registration data set.

If the primary registration data set fails and an additional primary registration data set is not available, the subsystem will terminate and no further BCSS processing will occur.

If installation guidelines were not followed when defining when the primary and duplex registration data sets were defined, it is possible that a single DASD error could destroy all primary and duplex registration data sets. You must then use a system backup copy for recovery if all of the following conditions occur:

- the primary registration data set fails.
- there are no other primary registration data sets available.
- no duplex registration data sets are available to recover the failed primary registration data set.

Recovery Tasks

This section contains the following task descriptions for recovering primary and duplex registration data sets:

- “Recovering a Primary Registration Data Set (Additional Primary Data Set)” on page 4-8
- “Recovering a Primary Registration Data Set Using the Duplex Registration Data Set” on page 4-9
- “Recovering a Primary Registration Data Set (Backup Copy)” on page 4-10
- “Recovering a Duplex Registration Data Set” on page 4-11
- “Making Off-Site Backups Using BMC Software EXTRACT” on page 4-12

Recovering a Primary Registration Data Set (Additional Primary Data Set)

Summary: You can recover the primary registration data set when additional primary registration data sets are available.

To recover a primary registration data set when additional primary registration data sets are available, complete the following steps:

- Step 1** Required. Stop the BCSS (see “Terminating BCSS” on page 2-8).
- Step 2** Required. Determine the cause of the primary registration data set failure and complete the necessary corrective actions.
- Step 3** Required. Restart the BCSS (see “Terminating BCSS” on page 2-8).

As the BCSS initializes, it will initialize the corrected primary registration data set and then synchronize it with the most current primary registration data set that was previously active.

Recovering a Primary Registration Data Set Using the Duplex Registration Data Set

Summary: You can recover a primary registration data set when no additional primary registration data sets are available if you had an active duplex registration data set at the time of failure.

To recover the primary registration data set using a duplex registration data set when no additional primary registration data sets are available, complete the following steps:

- Step 1** Required. Stop the BCSS (see “Terminating BCSS” on page 2-8).
- Step 2** Required. Determine the cause of the primary registration data set failure and complete the necessary corrective actions.
- Step 3** Required. Redefine the primary registration data set.
- Step 4** Required. Use the IDCAMS REPRO command to copy the duplex registration data set into *one* of the corrected primary registration data sets with JCL similar the following sample:

```
//jobname JOB . . .
//REPRO EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//DUPLEX DD DSN=duplex.regiset.dataset,DISP=SHR
//PRIMARY DD DSN=primary.regiset.dataset,DISP=SHR
//SYSIN DD *
REPRO INFILE(DUPLEX) OUTFILE(PRIMARY) REPLACE
//
```

The primary registration data sets are not automatically re-synchronized at BCSS startup when you use a previously active *duplex* registration data set.

- Step 5** Required. Restart the BCSS (see “Starting BCSS” on page 2-7).

The BCSS will initialize any additional primary registration data sets that were corrected and then synchronize them with the primary registration data set that was populated by the REPRO command in Step 4.

Recovering a Primary Registration Data Set (Backup Copy)

Summary: You can recover a primary registration data set by using a backup copy of the primary registration data set or a BMC Software EXTRACT. It does not matter whether the backup copy is contained in a full volume physical or logical dump format, or whether it is created stand-alone or periodically. However, any such data will be outdated and will not reflect your most current compression environment. You should expect some inconsistencies.

To recover the primary registration data set by using a backup copy of the primary registration data set, complete the following steps:

- Step 1** Required. Stop the BCSS (see “Terminating BCSS” on page 2-8).
- Step 2** Required. Determine the cause of the primary registration data set failure and complete the necessary corrective actions.
- Step 3** Required. Restore one of the corrected primary registration data sets using the restore facilities of the backup software (these may be directed to alternate volumes). If you are using the BMC Software EXTRACT facility, you must use the IDCAMS REPRO to copy the data into a corrected primary registration data set.

Note: See “Registration Data Sets” on page 4-3 for more information on defining the REGISSET(s), especially if you are relocating the REGISSET(s) to different volumes.

- Step 4** Required. Restart the BCSS (see “Starting BCSS” on page 2-7).

As the BCSS initializes, it will synchronize the remaining primary registration data sets that were corrected in Step 2 with the data that was restored in Step 3.

Recovering a Duplex Registration Data Set

Summary: The BCSS corrects most duplex registration data set failures automatically. When the BCSS cannot fix a failed duplex registration data set, it issues a message stating that you will need to recover a particular duplex registration data set.

To recover a duplex registration data set, complete the following steps:

- Step 1** Required. Stop the BCSS (see “Terminating BCSS” on page 2-8).
- Step 2** Required. Determine the cause of the duplex registration data set failure and complete the necessary corrective actions.

Note: You do not need to restore the contents of a duplex registration data set. When the BCSS initializes, it makes the necessary changes to ensure the duplex registration data set is an exact copy of the primary registration data set.

- Step 3** Required. Restart the BCSS (see “Starting BCSS” on page 2-7).

Making Off-Site Backups Using BMC Software EXTRACT

Summary: For disaster recovery, make EXTRACTs of the registration data sets to take off-site at regular intervals.

To make EXTRACT copies of your registration data sets, complete the following steps:

Step 1 Required. Use the BCSEXAM in JCL.

Note: BMC Software recommends that you create backup copies of the registration data sets by using this JCL only. This JCL uses subsystem services to make the backup copies and guarantees the output as being the most recent copies of the registration data sets.

Step 2 Required. Modify the JCL as needed, and submit the job.

Chapter 5 Utilities and Programs

This chapter describes BMC Software subsystem utilities and programs. Using these utilities and programs is optional. The following sections are included:

Extract Utility	5-2
Using the Extract Utility.	5-3
Creating the INCLUDE and EXCLUDE Statements.	5-4
Using the PARM and BCSSID Parameters	5-6
Using the Output Statement	5-7
Calling the Extract Utility from Other Programs.	5-8
Installation Check Utility	5-9
Executing the Installation Check Utility	5-10
JCL Requirements	5-11
Parameter Options	5-12
REGISET Cleanup Utility	5-13
Submitting the First Job	5-14
Submitting the Second Job.	5-16
Preparing the PARM Parameter	5-17
Registration Data Set Records	5-18
Fixed Section—Registration Header Record	5-19
Data Set Control Record Block Section.	5-20
Program Control Record Block Section.	5-24
Statistical Section	5-27

Extract Utility

The Extract utility retrieves all or a specified subset of the entries in the registration data sets (the REGISSET) and places these entries in a sequential data set. This utility makes REGISSET entries and their statistical data available for interpretation by other utilities.

This section describes the following tasks for the Extract utility.

- “Using the Extract Utility” on page 5-3
- “Creating the INCLUDE and EXCLUDE Statements” on page 5-4
- “Using the PARM and BCSSID Parameters” on page 5-6
- “Using the Output Statement” on page 5-7
- “Calling the Extract Utility from Other Programs” on page 5-8

Using the Extract Utility

Summary: Use the Extract utility to create a sequential file that contains registration and statistical information. See figure Figure 5-1 on page 5-6 for sample JCL.

To use the Extract utility, complete the following steps:

- Step 1** Required. Modify the JOB statement, as necessary.
- Step 2** Required. Modify the SYSUT2 DD statement to include the name of the data set in which the Extract utility will place the REGISSET records.
- Step 3** Required. Add as many INCLUDE and EXCLUDE statements as necessary to the SYSIN DD statement to reflect the entries you want the Extract utility to include and exclude. See “Creating the INCLUDE and EXCLUDE Statements” on page 5-4. If you do not specify any INCLUDE or EXCLUDE statements, the Extract utility extracts all entries in the REGISSET.
- Step 4** Optional. Add the PARM parameter to change the name of the SYSIN DD statement and/or the name of the subsystem. Add the BCSSID parameter if you are extracting data from a test subsystem. See “Using the PARM and BCSSID Parameters” on page 5-6.
- Step 5** Required. Submit the job.

Creating the INCLUDE and EXCLUDE Statements

Summary: You can limit the entries that you want the Extract utility to process by using one or more INCLUDE or EXCLUDE statements. The INCLUDE statement instructs the Extract utility to process only the entries in this statement. The EXCLUDE statement instructs the Extract utility to process all entries in the REGiset except the entries in this statement.

To create the INCLUDE and EXCLUDE statements for the Extract utility, complete the following steps:

- Step 1** Required. Begin the statement with the INCLUDE or EXCLUDE command. Ensure each INCLUDE or EXCLUDE statement adheres to the following rules:
- Use columns 2 through 71 only.
 - Place all INCLUDE and EXCLUDE statements *after* any OUTPUT control statements.
 - Place each INCLUDE and EXCLUDE statement in a separate job step.
 - Do not continue an INCLUDE or EXCLUDE statement on a second line; begin a new INCLUDE or EXCLUDE statement.
- Step 2** Required. Create operands to indicate what the Extract utility should include or exclude. An operand is a character string. Its first character identifies the type of entry you want to include or exclude:

D data set
P program

The second character identifies whether the entry is discrete or generic:

D discrete
G generic

The remaining characters identify all or part of an entry name. You can use wildcard characters to create a wildcard pattern for the entry name.

If you are using more than one operand on an INCLUDE or EXCLUDE statement, separate the operands with commas.

Examples INCLUDE DGACCT,PDBACK,DDABC.BSAM.DATA

This statement indicates that the Extract utility should include in its output the following items:

- generic data set entries with names starting with the string ACCT
- program entries with names starting with the string BACK
- discrete data set entries with names starting with the string ABC.BSAM.DATA

EXCLUDE DDALPH,DGOUTP,PD

This statement indicates that the Extract utility should exclude from its output the following items:

- discrete data set entries with names that start with the string ALPH
- generic data set entries with names that start with the string OUTP
- program entries

Using the PARM and BCSSID Parameters

Summary: You can add parameters to the EXEC statement to change the SYSIN ddname and to extract data from a test subsystem.

To use the PARM and BCSSID parameters with the Extract utility, complete the following steps:

- Step 1** Optional. Use the PARM parameter to specify the ddname, which can be a maximum of eight characters. In Figure 5-1, the new name for the SYSIN DD statement is *CTLCRDS*.
- Step 2** Optional. Use the BCSSID parameter to specify the name of a test subsystem, which can be a maximum of four characters. In Figure 5-1, the new subsystem name is *TEST*.

Figure 5-1 EXEC Statement with Additional Parameters

```
//jobname JOB (acct), 'name', CLASS=x, REGION=2048K
//BCSUEXTR EXEC PGM=BCSUEXTR, PARM='CTLCRDS, BCSSID=TEST'
//SYSUT2 DD DSN=data.set.name, DISP=(, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL,(5,15), RLSE)
//SYSPRINT DD SYSOUT=*
//CTLCRDS DD *
EXCLUDE P
EXCLUDE DDSYS1
/*
//
```

Using the Output Statement

Summary: You can change the SYSPRINT and SYSUT2 ddnames on the OUTPUT statement.

To change the SYSPRINT and SYSUT2 ddnames used by the Extract utility, complete the following steps:

Step 1 Required. Ensure each OUTPUT statement adheres to the following rules:

- Use columns 2 through 71 only.
- Place one or two OUTPUT statements in the input stream.
- Ensure OUTPUT statements precede any INCLUDE or EXCLUDE statements.

Step 1 Optional. Include an OUTPUT control statement in the input control stream, as shown in Figure 5-2. The Extract utility processes output statements before it opens the SYSPRINT data set; therefore, errors found in the OUTPUT statements do not appear in the SYSPRINT data set. A write-to-operator (WTO) function displays these error messages. The error messages also appear in the JES job log.

The Extract utility saves the OUTPUT statements in storage and prints them with other control statements after it opens the SYSPRINT data set. If you did not provide a SYSPRINT ddname, the Extract utility uses the default SYSOUT class for the job to allocate the SYSPRINT data set dynamically.

Figure 5-2 Input Stream with an OUTPUT Statement

```
//jobname JOB (acct), 'name', CLASS=x, REGION=2048K
//BCSUEXTR EXEC PGM=BCSUEXTR, PARM='CTLCRDS, BCSSID=TEST'
//DATAOUT DD DSN=data.set.name, DISP=(,CATLG,DELETE),
// UNIT=SYSDA, SPACE=(CYL,(5,15),RLSE)
//SYSOUT DD SYSOUT=*
//CTLCRDS DD *
OUTPUT SYSPRINT=SYSOUT, SYSUT2=DATAOUT
EXCLUDE P
EXCLUDE DDSYS1
/*
//
```

Calling the Extract Utility from Other Programs

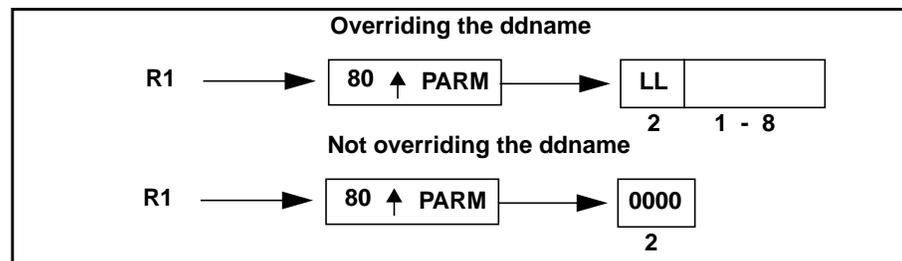
Summary: Other programs can call the Extract utility as a subroutine. The calling program can specify an alternate ddname for the SYSIN statement in the Extract utility through a PARM when the program is called (see Figure 5-3).

To call the Extract utility from other programs, complete the following steps:

Step 1 When you specify an alternate ddname, the PARM parameter consists of two fields that must be contiguous in storage. The first field is a halfword binary value that is the length of the second field. The second field is a 1-byte to 8-byte EBCDIC value used as the alternate ddname for the SYSIN statement.

If the calling program does not specify an alternate ddname for the SYSIN statement, the PARM is a halfword binary value containing zeros.

Figure 5-3 PARM Diagram



Installation Check Utility

The Installation Check utility searches for other BMC Software products in your system that share the common subsystems. The Installation Check utility will compare the modules in the installation library with those found in the system or pointed to by DD statements for existing products. The Installation Check utility will indicate the highest level modules and make recommendations on what modules to copy into the production libraries.

Note: The Installation Check utility should be run whenever a new release of a BMC Software product that utilizes the common subsystems is installed.

This following information is provided in this section

- executing the Installation Check utility
- JCL requirements
- parameter options

Executing the Installation Check Utility

Summary: The Installation Check utility will compare the modules in the installation library with those found in the system or pointed to by DD statements for existing products, indicate the highest level modules, and make recommendations on what modules to copy into the production libraries.

To execute the installation check utility, complete the following steps.

Note: Execute the Installation Check utility during normal operating conditions on each MVS or OS/390 image that uses the products you are installing. This utility requires READ access to all LINKLIST libraries.

- Step 1** Customize the JCL found in the BCSCHECK member of the installation JCL library.
- Step 2** Submit the customized JCL on each system on which the product being installed will operate.
- Step 3** Check the program output when the utility ends. The output includes
- a report of the current subsystem modules installed on the running system
 - installation modification instructions based on the review of the current environment
 - a module cross-reference
- Step 4** Modify the installation process based on the recommendations of the installation check utility output.

JCL Requirements

The following describes the JCL necessary for the operation of the Installation Check utility.

Following is a sample JCL for running the Installation Check utility.

```
//BCS EXEC PGM=BCSCHECK , PARM=( ' PREFIX(BPS , BCS) ' , CHECK , XREF , NOBCS , NOSSI )  
//STEPLIB DD DSN=BCS.V1R1D.LOAD , DISP=SHR  
//SYSPRINT DD SYSOUT=$  
//REPORT DD SYSOUT=$  
//JCLLIB DD SYSOUT=$  
//SYSLIB DD DSN=BCS.V1R1D.LOAD , DISP=SHR
```

The first JCL statement is the EXEC statement. You must specify the program name BCSCHECK and provide the optional parameter values as needed. The parameter names and values are described in Table 5-1 on page 5-12.

The second JCL statement is a STEPLIB to point to the load library containing the BCSCHECK program. If the BCSCHECK program is installed into a LINKLIST library then this statement can be ignored.

The third JCL statement defines the file DD where the log messages and necessary warning and/or error messages will be placed.

The fourth JCL statement defines the file DD where the BCSCHECK program will output the module cross-reference and module status information.

The fifth JCL statement defines the file DD that the BCSCHECK program will output the optional JCL statements to copy the modules identified as upgraded.

The sixth JCL statement defines the file DD of the “new” load library that BCSCHECK will compare to the modules found on the current system.

Parameter Options

Table 5-1 describes the parameters necessary to customize the operation of the Installation Check utility. These are passed to the BCSCHECK program through the **PARM** field of the **EXEC** statement.

Table 5-1 Installation Check Utility Parameter Options

Option	Description
CHECK	perform the regression check analysis and scan for current install anomalies
NOCHECK	only check existing BMC Software subsystems existence and scan for current install anomalies
XREF	generate the module cross-reference report during regression analysis
NOXREF	bypass generating the module Cross-reference report only report modules that will be regressed
NOLINKLIST	do not check the link list concatenation for subsystem module during regression analysis
LINKLIST	check the Link List concatenation for subsystem modules during regression analysis
ALLPDSUPGRADE	regression scan will contain all product datasets
NOALLPDSUPGRADE	regression scan will only contain Link list and LPA module scans
NOBCSCHECK	do not Check for compression products
BCSCHECK	check for compression products
NOSSICHECK	do not use the BMC Software subsystem running to check for module regression
SSICHECK	use the current BMC Software subsystem running for module regression analysis
SYSLIB()	DDNAME to use in place of the SYSLIB name
SYSPRINT()	DDNAME to use in place of the SYSPRINT name
DOCLIB	reproduce the DOCLIB member
PDS()	check the product PDS for regression and optionally set the prefix for the DDNAME to use pointing to the product datasets
REPORT()	DDNAME to use in place of the REPORT name
SSID()	used to specify which BMC Software subsystem to use for regression analysis

REGISET Cleanup Utility

The REGISET Cleanup utility generates Batch Registration utility delete commands to remove from the REGISET discrete data set entries that meet the following criteria:

- The discrete data set entry must indicate that compressed data exists for the associated data set.
- No pending records for the discrete data set entry must exist.
- The data set associated with the discrete data set entry must be uncataloged.
- A generic data set entry must cover the discrete data set entry, unless you have also included the FORCE keyword (for more information, see “FORCE | NOFORCE” on page 5-17).

Note: The REGISET Cleanup utility is for use with the BMC Software subsystem components only.

The JCL for the REGISET Cleanup utility is contained in members BCSURDS0 and BCSURDS1 of the installation data set on the distribution tape. Member BCSURDS0 generates the Batch Registration utility delete commands. The REGISET Cleanup utility places these delete commands into a UBREGOUT data set and also generates a report that provides the following information:

- which generic data set entry covers the discrete data set entry being flagged for deletion
- if the FORCE keyword is the reason the entry is eligible for deletion

After you review the delete commands and the report, you can execute a second job that submits the Batch Registration utility delete commands. The REGISET Cleanup utility uses the UBREGOUT data set generated by the first job as input to the second job. Member BCSURDS1 executes the Batch Registration utility delete commands. The selected discrete data set entries are deleted from the REGISET.

Submitting the First Job

Summary: Submit the job in member BCSURDS0 to generate the Batch Registration utility delete commands and a report that explains the delete commands.

To generate the delete commands and a report, complete the following steps:

- Step 1** Required. Use the JCL in member BCSURDS0 of data set BMC.DAC.JCL on the distribution tape.
- Step 2** Required. Modify the JOB statement, as necessary.
- Step 3** Required. Modify all instances of *?USERID?* to reflect the high-level qualifier that you want to use. Use the same qualifier in all DD statements.
- Step 4** Required. Modify *?SHAREDQUAL?* in the STEPLIB DD statement to reflect the high-level qualifier of your BMC Software load library. If you installed the REGISET Cleanup utility JCL in a linklist, you can delete the STEPLIB DD statement.
- Step 5** Required. Modify the PARM parameter in the EXEC statement to identify which REGISET entries are eligible for cleanup. See “Preparing the PARM Parameter” on page 5-17 for instructions about constructing the PARM parameter.
- Step 6** Required. Submit the job.
- Step 7** Required. The *?USERID?.BCS.UBREGOUT* data set lists any discrete data set entries meeting your criteria for the REGISET Cleanup utility. Review the Batch Registration utility delete commands in this data set to ensure you are not deleting necessary discrete data set entries. Figure 5-4 shows a sample of the *?USERID?.BCS.UBREGOUT* data set.

Figure 5-4 Sample *?USERID?.BCS.UBREGOUT* Data Set

```

PROFILE NOPREFIX MSGID
DACUBREG DEL ABC.COMPOPEN.AIX009.CL          DIS FORCE
DACUBREG DEL ABC.COMPOPEN.KSDS              DIS FORCE

```

- Step 8** Required. The REGISSET Cleanup utility report indicates the generic data set entry that would cover an existing discrete data set entry. This report also indicates whether the REGISSET Cleanup utility flagged the discrete data set entry as eligible for deletion because of the *FORCE* keyword. Figure 5-5 contains a sample of this report. Use this report to help you make any additional changes to the ?USERID?.BCS.UBREGOUT data set.

Figure 5-5 Sample REGISSET Cleanup Utility Report

```

month dd, yyyy          BMC SOFTWARE, INC
hh.mm                  REGISSET CLEANUP REPORT

PARAMETER: PRODUCTID = DPV    LEVEL = ABC
DELETE GENERATED FOR DISCRETE    GENERIC ENTRY COVERING DISCRETE
-----
ABC.COMPOPEN.AIX009.CL        ABC.COMPOPEN
ABC.COMPOPEN.KSDS            ABC.COMPOPEN

```

- Step 9** Required. Delete any Batch Registration utility delete commands from the ?USERID?.BCS.UBREGOUT data set if you want to retain the entry in the REGISSET.

Submitting the Second Job

Summary: Submit the second job to execute the delete commands, which remove unneeded discrete data set entries from the REGISSET.

To execute the delete commands, complete the following steps:

- Step 1** Required. Use the JCL in member BCSURDS1 of data set BMC.DAC.JCL on the distribution tape.
- Step 2** Required. Modify the JOB statement, as necessary.
- Step 3** Required. Modify *?USERID?* in the SYSTSIN DD statement to reflect the high-level qualifier that you want to use. Use the same qualifier as you did in the first job.
- Step 4** Required. Modify *?SHAREDQUAL?* in the STEPLIB DD statement to reflect the high-level qualifier of your BMC Software load library. If you installed the REGISSET Cleanup utility JCL in a linklist, you can delete the STEPLIB DD statement.
- Step 5** Required. Submit the job. Ensure that the job completes successfully.

Preparing the PARM Parameter

The PARM parameter has the following format. When you include more than more than one parameter, you must separate the parameters with commas:

```
PARM= ' DPV , FORCE , LEVEL=xxxxxxxxx '
```

You can enter the following three values for the PARM parameter:

- DPV | DPO

Required. Specify DPV to delete VSAM component discrete data set entries. Specify DPO to delete non-VSAM component discrete data set entries. If you include the FORCE, NOFORCE, or LEVEL parameter, you must place the DPV or DPO parameter in the first position.

- FORCE | NOFORCE

Optional. Specify FORCE or NOFORCE to indicate whether the REGISSET Cleanup utility should delete (FORCE) or retain (NOFORCE) the entry if all conditions are met except for the one concerning the existence of a generic data set entry. The default is NOFORCE.

- LEVEL=xxxxxxxx

Optional. Specify a high-level qualifier to limit the selection of data sets entries eligible for deletion. For example, LEVEL=ABC will search all discrete data set entries whose associated data set has ABC as the high-level qualifier. The default is LEVEL=**, which will include all discrete data set entries that belong to the component indicated in the first parameter position (DPV or DPO).

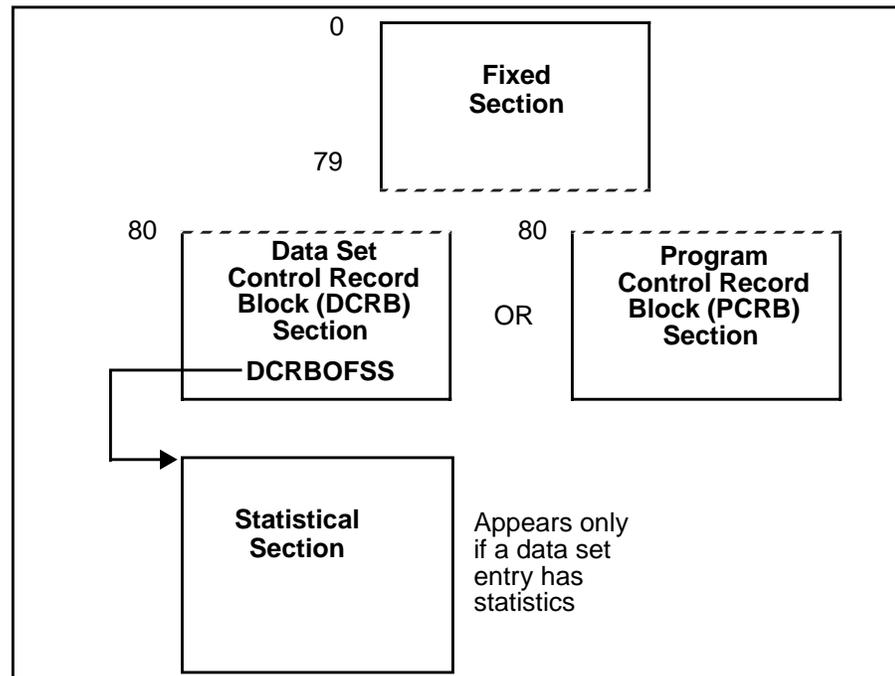
Registration Data Set Records

The records extracted from the REGISET by the Extract utility (see “Extract Utility” on page 5-2) have one fixed section (registration header record) and one or more variable sections as shown in Figure 5-6.

The fixed section begins at offset zero (0) and ends at offset position 79 (inclusive). If the record represents a data set entry, the first variable section contains data set information. If the record represents a data set entry that contains statistics, a second variable section appears at the offset contained in the DCRBOFSS field of the first data set section.

If the record represents a program entry, the first variable section contains program information.

Figure 5-6 Sections of REGISET Record



Fixed Section—Registration Header Record

The fields shown in Table 5-2 appear in the fixed section of a REGISET record.

Table 5-2 REGISET Record Fixed Section

Fieldname	Dec Offset	Hex Offset	Length	Type	Description
RCRKTYPE	0	0	1	char	record type: 'D' for data set or 'P' for program
RCRKSUBT	1	1	1	char	record subtype: 'D' for discrete or 'G' for generic (invalid for program)
RCRKDSN or RCRKPGMN	2 2	2 2	44 8	char char	data set entry name program entry name
filler	10	A	36	x'00'	filler
reserved	46	2E	8	bin	reserved
RCRKSEQN	54	36	2	bin	A value of 0000 indicates the current state of the data set. A value of 0080 indicates that the information in this record will become current at the next open. A value of 0100 indicates that the information in this record will become current at the next load.
reserved	56	38	8	bin	reserved
RCRTOD	64	40	8	bin	time of day of the last update to this record; used to detect synchronization errors
RCRVRSN	72	48	1	bin	version of control record
RCRRLSE	73	49	1	bin	release of control record
reserved	74	4A	2	bin	reserved
reserved	76	4C	4	bin	reserved

Data Set Control Record Block Section

The fields shown in Table 5-3 appear when the REGISET record represents a data set entry.

Table 5-3 REGISET Record DCRB Section (Part 1 of 4)

Fieldname	Dec Offset	Hex Offset	Length	Type	Description
DCRBSID	80	50	4	char	section acronym 'DCRB'
DCRBSSZE	84	54	2	bin	size of the section
DCRSSIZE	86	56	2	bin	size of storage allocated for all sections managed by this section (not including DCRB)
reserved	88	58	2	bin	reserved
reserved	90	5A	2	bin	reserved
DCRBOFSS	92	5C	2	bin	offset to statistical section
reserved	94	5E	2	bin	reserved
reserved	96	60	2	bin	reserved
reserved	98	62	2	bin	reserved
reserved	100	64	2	bin	reserved
reserved	102	66	2	bin	reserved
DCRBBOI	104	68	1	char	exclude data set from processing (Y or N) Discrete: Y indicates that open interceptions for this entity should be bypassed. Generic: Y indicates that open interceptions for any matching entity should be bypassed.
DCRBCMPR	105	69	1	char	compress data (Y or N) Discrete: Y indicates that compression is being or will be performed. Generic: Y indicates that compression will begin when data set is opened.
DCRBCDE	106	6A	1	char	compressed data exists (Y or N) Discrete: Y indicates that at least one record is compressed. Generic: Y is invalid.
DCRBSIM	107	6B	1	char	simulate compression (Y or N); no data is modified
reserved	108	6C	4	bin	reserved

Table 5-3 REGISET Record DCRB Section (Part 2 of 4)

Fieldname	Dec Offset	Hex Offset	Length	Type	Description
DCRBVSAM	112	70	1	char	VSAM data set (Y or N) Discrete: Y indicates that this entry represents a VSAM data set. Generic: Y indicates that VSAM data sets are included in the scope of this entry. When RCRVRSN=01, this field contains binary zeroes. When RCRVRSN=02, this field contains a four-character product ID. DPO=non-VSAM component DPV=VSAM component
DCRBOS	113	71	1	char	non-VSAM data set (Y or N) Discrete: Y indicates that this entry represents a non-VSAM data set. Generic: Y indicates non-VSAM data sets are included in the scope of this entry. When RCRVRSN=01, this field contains binary zeroes. When RCRVRSN=02, this field contains a four-character product ID. DPO=non-VSAM component DPV=VSAM component
DCRBPDS	114	72	1	char	partitioned data set (Y or N) Discrete: Y indicates that this entry represents a non-VSAM portioned data set. Generic: Y indicates that partitioned data sets are included in the scope of this entry.
DCRBSEQ	115	73	1	char	sequential data set (Y or N) Discrete: Y indicates that this entry represents a non-VSAM sequential data set. Generic: Y indicates that sequential data sets are included in the scope of this entry.
DCRBOSQST	116	74	4	char	QSAM Truncation option (Y or N) Discrete: Y indicates that QSAM truncation is enabled. Generic: Y indicates that QSAM truncation is enabled.
DCRBDASD	120	78	1	char	DASD-resident (Y or N) Discrete: Y indicates that this data set resides on DASD. Generic: Y indicates that this data set must reside on DASD to be included in the scope of this entry.
DCRBTAPE	121	79	1	char	Tape -resident (Y or N) Discrete: Y indicates that this data set resides on tape. Generic: Y indicates that this data set must reside on tape to be included in the scope of this entry.

Table 5-3 REGISET Record DCRB Section (Part 3 of 4)

Fieldname	Dec Offset	Hex Offset	Length	Type	Description
reserved	122	7A	2	bin	reserved
DCRBDPL	124	7C	1	char	delete pending load (Y or N) Discrete: Y indicates that this entry will be deleted after the data set is loaded uncompressed. Valid only for upgrade records. Generic: Y is invalid.
DCRBCPL	125	7D	1	char	compress pending load (Y or N) Discrete: Y indicates that records in this data set will start being compressed at the next load. Valid only for upgrade records Generic: Y indicates that records in these data sets will start being compressed at the next load. Valid only for upgrade records.
DCRBUPL	126	7E	1	char	update pending load (Y or N) Discrete: Y indicates that the entry is updated when the data set is loaded. Valid for upgrade records only. Generic: Y indicates that the entry is updated when the data set is loaded. Valid for upgrade records only.
reserved	127	7F	1	bin	reserved
DCRBSRPO	128	80	1	char	Reset statistics at open (Y or N) Discrete: Y indicates that the statistics will be reset when the data set is opened. Valid for upgrade records only. Generic: Y is invalid.
reserved	129	81	3	bin	reserved
DCRBLRIB	132	84	1	char	(This is a VSAM-only field.) Logical Record Interface is bypassed (BMC Software subsystem detected a Control-Interval access against the data set.) Discrete: Y indicates that the open/close intercept or the logical record interface detected the condition. Generic: Y is invalid.
DCRBUPIP	133	85	1	char	(This is a VSAM-only field.) Update-in-Place occurred or was attempted (BMC Software subsystem detected an RBA update-in-place attempt against the data set.) Discrete: Y indicates that the open/close intercept or the logical record interface detected the condition. Generic: Y is invalid.
reserved	134	86	10	bin	reserved
DCRBCOSET	144	90	4	bin	offset from the start of the record to the first compressible byte of data

Table 5-3 REGISET Record DCRB Section (Part 4 of 4)

Fieldname	Dec Offset	Hex Offset	Length	Type	Description
DCRBSOPT	148	94	1	char	Space Release option (Y or N) Y indicates release unused space at next close.
reserved	149	95	1	bin	reserved
DCRBSPCT	150	96	2	bin	Space Release Percent. Percentage of unused space to be retained during space release.

Program Control Record Block Section

The fields shown in Table 5-4 appear when the REGISET record represents a program entry. These fields can occur once for each product that registers the program entry. When multiple products register a program, the second program control record block (PCRB) immediately follows the first, and you must subtract the sum of the header length and the PCRB length from the total record length to determine whether a second PCRB exists.

Table 5-4 REGISET Record PCRB Section (Part 1 of 3)

Fieldname	Dec Offset	Hex Offset	Length	Type	Description
PCRBSID	80	50	4	char	section name: 'PCRB'
PCRBSSZE	84	54	2	bin	size of the section
PCRSSIZE	86	56	2	bin	size of the storage allocated for all sections managed by this section (not including RCR) (currently zero)
reserved	88	58	8	bin	reserved
PCRBBOI	96	60	1	char	bypass interception (Y or N) of opens for all data sets used by the program
reserved	97	61	3	bin	reserved
PCRBVSAM	100	64	1	char	VSAM data set (Y or N) N indicates that VSAM data sets opened by this program are exempt from compression and expansion. When RCRVRSN=01, this field is present. When RCRVRSN=02, this field contains binary zeroes.
PCRBOS	101	65	1	char	non-VSAM indicator (Y or N) N indicates that non-VSAM data sets opened by this program are exempt from compression and expansion. When RCRVRSN=01, this field is present. When RCRVRSN=02, this field contains binary zeroes.
PCRBPDS	102	66	1	char	partitioned data set indicator (Y or N) N indicates that partitioned data sets opened by this program are exempt from compression and expansion.
PCRBSEQ	103	67	1	char	sequential data set indicator (Y or N) N indicates that sequential data sets opened by this program are exempt from compression and expansion.
reserved	104	68	4	bin	reserved

Table 5-4 REGSET Record PCRB Section (Part 2 of 3)

Fieldname	Dec Offset	Hex Offset	Length	Type	Description
PCRBPID	108	6C	4	char	When RCRVRSN=01, this field is described by the PCRBDASD and PCRBTAPE fields. When RCRVRSN=02, this field contains a four-character product ID. DPO=non-VSAM component DPV=VSAM component
PCRBDASD	108	6C	1	char	DASD data set indicator (Y or N) N indicates that DASD data sets opened by this program are exempt from compression and expansion. When RCRVRSN=01, this field is present. When RCRVRSN=02, this field is described by the PCRBPID field.
PCRBTAPE	109	6D	1	char	Tape data set indicator (Y or N) N indicates that tape data sets opened by this program are exempt from compression and expansion. When RCRVRSN=01, this field is present. When RCRVRSN=02, this field is described by the PCRBPID field.
reserved	110	6E	2	bin	reserved When RCRVRSN=01, this field is present.
When PCRBID=DPO or DPV					
reserved	112	70	1	char	reserved
reserved	113	71	1	char	reserved
reserved	114	72	2	bin	reserved
PCRBFLRI	116	74	1	char	Forced Logical Record Interface option (Y or N) Y indicates this program must use a logical record interface to access data.
PCRBCEL	117	75	1	char	Compression Location option (D, C, or L) D indicates use compression location established at product initialization. C indicates use CAS compression. L indicates use local compression.
PCRBOSQST	118	76	1	char	QSAM Truncation option (Y or N) Y indicates truncate data set. N indicates do not truncate data set.
reserved	119	77	1	char	reserved
reserved	120	78	8	bin	reserved
reserved	128	80	16	bin	reserved

Table 5-4 REGISET Record PCRB Section (Part 3 of 3)

Fieldname	Dec Offset	Hex Offset	Length	Type	Description
When PCRBID=DAC					
reserved	112	70	1	char	reserved
PCRBVDEFER	116	74	1	char	VSAM Defer Writes option (D, Y, or N) D indicates default to global options. Y indicates defer buffer writes. N indicates write buffers as they occur.
reserved	117	75	3	bin	reserved
reserved	120	78	8	bin	reserved
reserved	128	80	1	char	reserved
reserved	132	84	12	bin	reserved

Statistical Section

The fields shown in Table 5-5 appear when the REGISET record represents a data set entry and the entry contains statistics. Y equals the hexadecimal offset to the beginning of the data set section (X'50') plus the hexadecimal number contained in the DCRBOFSS field.

Table 5-5 REGISET Record Statistical Section (Part 1 of 2)

Fieldname	Dec Offset	Hex Offset	Length	Type	Description
DCRSSID	Y+0	Y+0	4	char	section acronym 'DCRS'
DCRSSIZE	Y+4	Y+4	2	bin	size of the section
reserved	Y+6	Y+6	2	bin	reserved
DCRSSTOD	Y+8	Y+8	8	bin	time of day of last use
CSBID	Y+16	Y+10	4	char	control block acronym 'CSB'
reserved	Y+20	Y+14	4	bin	reserved
CSBVRSN	Y+24	Y+18	1	bin	version
CSBRLSE	Y+25	Y+19	1	bin	release
CSBPFLAG	Y+26	Y+1A	1	bin	processing flags
CSBIFLAG	Y+27	Y+1B	1	bin	statistical indicators
reserved	Y+28	Y+1C	4	bin	reserved
CSBLDTOP	Y+32	Y+20	8	bin	load time of day used as time of compression
reserved	Y+40	Y+28	56	bin	reserved
CSBCRCNT	Y+96	Y+60	4	bin	number of compress requests
CSBCRNSE	Y+100	Y+64	4	bin	number of unsuccessful compress requests that do not fall into another unsuccessful compression category
CSBCRITS	Y+104	Y+68	4	bin	number of unsuccessful compress requests that occurred because the data was too short to compress
CSBCRCNC	Y+108	Y+6C	4	bin	number of unsuccessful compress requests that occurred because the data input was uncompressible
CSBCRIAC	Y+112	Y+70	4	bin	number of unsuccessful compress requests that occurred because the data input was already compressed
reserved	Y+116	Y+74	28	bin	reserved
CSBCNBID	Y+144	Y+90	8	bin	number of bytes of input data
CSBCNBIC	Y+152	Y+98	8	bin	number of bytes of input data that were compressed
CSBCNBCO	Y+160	Y+A0	8	bin	number of bytes of compressed output

Table 5-5 REGISET Record Statistical Section (Part 2 of 2)

Fieldname	Dec Offset	Hex Offset	Length	Type	Description
CSBCNBSP	Y+168	Y+A8	8	bin	number of bytes of record pad in the compressed output data
reserved	Y+176	Y+B0	48	bin	reserved
CSBERCNT	Y+224	Y+E0	4	bin	number of expand requests
CSBERNSE	Y+228	Y+E4	4	bin	number of unsuccessful expand requests that do not fall into another unsuccessful expand category
CSBERINC	Y+232	Y+E8	4	bin	number of unsuccessful expand requests that occurred because the data was not compressed
reserved	Y+236	Y+EC	36	bin	reserved
CSBENBID	Y+272	Y+110	8	bin	number of bytes of input data
CSBENBIE	Y+280	Y+118	8	bin	number of bytes of input data expanded
CSBENBEO	Y+288	Y+120	8	bin	number of bytes of expanded output data
reserved	Y+296	Y+128	56	bin	reserved

Chapter 6 BMC Software Subsystem Messages

This section contains general information about the messages that are issued by the BMC Software Primary Subsystem (BMCP) and the BMC Software Consolidated Subsystem (BCSS). It describes the following items:

- interpreting subsystem messages
- BMCP messages
- BCSS messages

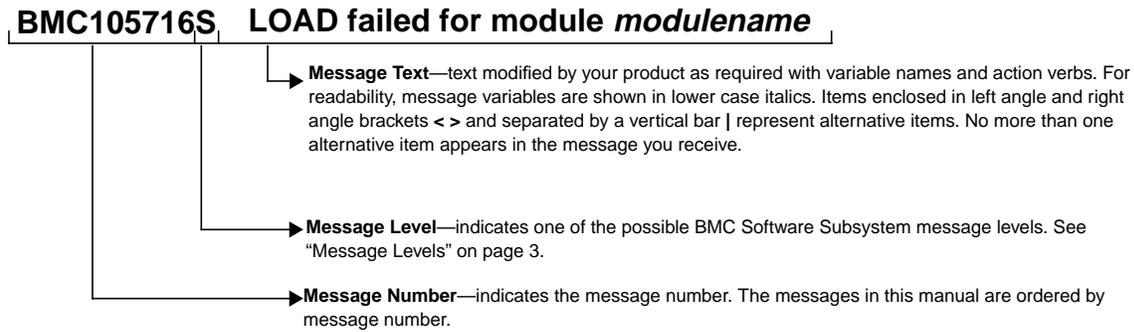
This chapter contains the following information:

Parts of a Message	6-2
Components that Generate Messages	6-2
Description Format	6-3
Message Levels	6-3
Contacting BMC Software Product Support.	6-4
BMC10000–BMC10999.	6-5
BMC11000–BMC11999.	6-16

Parts of a Message

Figure 6-1 shows and describes the parts of a BMC Software Subsystem message.

Figure 6-1 Parts of a BMC Software Subsystem Message



Message text that is italicized (*example*) and in lower case indicates variable text that will be determined when the message is issued.

Message text that is enclosed in left angle and right angle brackets and separated by vertical lines <ON | OFF | DEFAULT> indicates actual values, one of which will be included at that point in the message.

Components that Generate Messages

The BMC Software Subsystem generates messages from any of several pieces of software. Table 6-1 lists the components that generate BMC Software Subsystem messages.

Table 6-1 Components that Generate BMC Software Subsystem Messages

Message Range	Component
BMC10000–BMC10999	BMC Software Primary Subsystem (BMCP)
BMC11000–BMC11999	BMC Software Consolidated Subsystem (BCSS) and the Extract program BCSUEXTR

Description Format

The following information is provided for each message:

- *Explanation* explains why the product issued the message.
- *System Action* describes what the product did as a result of encountering the situation.
- *User Response* describes what you should do in this situation.

Message Levels

Table 6-2 describes the BMC Software Subsystem message levels.

Table 6-2 BMC Software Subsystem Message Levels

Message Level	Description
A (action)	Immediate action is required.
E (error)	The function that you requested was not completed. Action is required.
I (information)	Information only. No action is required.
R (reply)	You must reply to the message before the system can continue.
S (severe)	A severe error occurred. Action is required.
W (warning)	The system is still operating and no immediate action is required. When time is available, further investigation is needed.

Contacting BMC Software Product Support

Some message descriptions instruct you to contact your BMC Software product support representative. The product support representative can help you resolve the problem quickly if you can answer the following questions before calling:

- What kind of problem do you have?
- Can you repeat the problem or preceding conditions?
- Have you reviewed the diagnostic procedures described in the reference manual for your specific product?
- Do you have supporting dumps or other diagnostic information?
- What has changed in your MVS environment:
 - Have you recently installed a new product on your system?
 - Have you recently modified an application program?
 - Have you recently installed a BMC Software product or product maintenance tape?

BMC10000–BMC10999

This section describes messages in the range BMC10000 to BMC10999, which are generated by the BMC Software Primary Subsystem (BMCP).

BMC1000R Enter initialization options or “U”. *bmcpid*

Explanation: The BMCP is prompting the MVS operator for options that control the method of initializing the environment.

System Action: The BMCP waits for an operator response.

User Response: The MVS system operator should respond with the proper initialization options. For example, reply *xx,SUBSYSID=bmcpid* where *xx* is the identification number (id) and *bmcpid* is the requested subsystem identifier. Reply **U** to use default values for any non-required parameters.

BMC10002E Invalid initialization option(s) specified. *bmcpid*

Explanation: One or more of the initialization options specified are invalid. Previously specified options are retained.

System Action: The invalid option(s) are ignored.

User Response: See the User Response for message BMC10000R, which immediately follows this message, and respond with the action it requires.

BMC10003E All required option(s) were not specified. *bmcpid*

Explanation: You did not specify one or more required BMCP command options when you submitted the command.

System Action: The command is rejected.

User Response: Verify the format of the command and include any missing required options.

BMC10004I SVCNO(*number*) specification being used. *bmcpid*

Explanation: This message identifies the number of the SVC being used to perform supervisory services. *number* indicates the SVC number.

System Action: The BMCP initialization process continues.

User Response: This message is informational only.

BMC10006I SSCT(*address*) for subsystem obtained and queued. *bmcpid*

Explanation: The BMCP issues this message when it has to construct and insert an subsystem control table (SSCT) into the SSCT queue. The BMCP was not previously active and an entry was not defined in any IEFSSN members of SYS1.PARMLIB. *address* indicates the SSCT address.

System Action: The BMCP initialization process continues.

User Response: This message is informational only.

BMC10009S Abend *abendcode* occurred during subsystem address space initialization. *bmcpid*

Explanation: An error occurred during the initialization of the BMCP. This message is issued during recovery processing and represents a severe error. *abendcode* indicates the generated abend code.

System Action: The BMCP initialization process terminates. Facilities provided by the BMCP may not be available.

User Response: Contact BMC Software Product Support.

BMC10020S Service module values not consistent with existing SSVT(*address*). *bmcpid*

Explanation: You attempted to use an earlier version of your product after the latest version had been initialized. Internal control block and/or module attributes are incorrect. *address* indicates the subsystem vector table (SSVT) address where the inconsistency occurred.

System Action: The BMCP initialization process terminates. Facilities provided by the BMCP may not be available.

User Response: Reinstall the latest BMCP. If this fails to solve the problem, contact BMC Software Product Support.

BMC10021S Subsystem down-leveling attempted, *oldlevel*, *newlevel*, SVT(*address*). *bmcpid*

Explanation: The BMCP tried to initialize at a lower level than previously active. The installed BMCP was created by a set of modules from the latest maintenance level. *oldlevel* and *newlevel* indicate the version and modification levels of the current modules and those being initialized. *address* indicates the SVT address.

System Action: The BMCP initialization process terminates. Facilities provided by the BMCP may not be available.

User Response: Restore the latest version of the subsystem modules, and restart the BMCP.

BMC10022S Primary subsystem address space(*asid*) is currently active. *bmcpid*

Explanation: An attempt was made to start a BMCP, but it has detected that another BMCP is already active in another address space. *asid* indicates the hexadecimal address space ID of the active BMCP.

System Action: The BMCP initialization process terminates. Since another BMCP is available, facilities provided by the BMCP are available.

User Response: Continue using the BMCP that is already active.

BMC10024S Service module FVT *offset/size(offset/size)* not consistent with the SSVT(*offset/size*). *bmcpid*

Explanation: You attempted to use an earlier version of your product after the latest version had been initialized. Internal control block and/or module attributes are incorrect. *offset/size* indicates the offset and size where the inconsistency occurred.

System Action: The BMCP initialization process terminates. Facilities provided by the BMCP may not be available.

User Response: Reinstall the latest BMCP modules. If this fails to solve the problem, contact BMC Software Product Support.

BMC10026S Service module SVT *offset/size(offset/size)* not consistent with the SSVT(*offset/size*). *bmcpid*

Explanation: You attempted to use an earlier version of your product after the latest version had been initialized. Internal control block and/or module attributes are incorrect. *offset/size* indicates the offset and size where the inconsistency occurred.

System Action: The BMCP initialization process terminates. Facilities provided by the BMCP may not be available.

User Response: Reinstall the latest BMCP modules. If this fails to solve the problem, contact BMC Software Product Support.

BMC10028S Service module PVT *offset/size(offset/size)* not consistent with the SSVT (*offset/size*). *bmcpid*

Explanation: You attempted to use an earlier version of your product after the latest version had been initialized. Internal control block and/or module attributes are incorrect. *offset/size* indicates the offset and size where the inconsistency occurred.

System Action: The BMCP initialization process terminates. Facilities provided by the BMCP may not be available.

User Response: Reinstall the latest BMCP. If this fails to solve the problem, contact BMC Software Product Support.

BMC10030S ***text* R15(*returncode*), *reasoncode*. *bmcpid***

Explanation: An unexpected non-zero return code or reason code was received in response to a BMCP verify request. A non-BMC Software product may have overlaid the BMCP, or an internal subsystem error may have occurred. *text* provides message text about the error. *returncode* and *reasoncode* indicate the generated return code and reason code.

System Action: The BMCP is terminated.

User Response: Check for any non-BMC Software product that may have overlaid BMCP, and try to correct the problem. If this is not applicable, contact BMC Software Product Support.

BMC10032S **Invalid subsystem affinity index(*index*) returned by subsystem verify. *bmcpid***

Explanation: An illogical subsystem affinity index was returned by the master MVS subsystem for the BMCP. *index* indicates the affinity index in hexadecimal format.

System Action: The BMCP is terminated.

User Response: Contact BMC Software Product Support.

BMC10034S **Default subsystem name(*name*) not consistent with subsystem being initialized. *bmcpid***

Explanation: An attempt was made to change the name of the BMCP during a restart operation. *name* indicates the default name of the BMCP.

System Action: The BMCP is terminated.

User Response: Restart the BMCP, and specify the correct SUBSYSID parameter. Use the subsystem ID that is appended to this message.

BMC10038S **Subsystem initialization process terminated. *bmcpid***

Explanation: The initialization process is being terminated due to one or more previous errors. Messages have been issued identifying the problem(s) encountered during the initialization process.

System Action: The BMCP is terminated.

User Response: See the User Response for the message(s) issued prior to this message.

BMC10040S **Default module address via SVC(*svcaddress*) not consistent with SSVT(*ssvtaddress*). *bmcpid***

Explanation: The address of a common area module returned by the SVC routine is not consistent with the address contained in the subsystem vector table (SSVT). This condition usually indicates that a storage overlay in the common service area (CSA) has occurred. *svcaddress* and *ssvtaddress* indicate the inconsistent module addresses.

System Action: The BMCP is terminated.

User Response: Contact BMC Software Product Support. An SVC dump of the common areas (CSA, LPA, NUC, and SQA) may be required for diagnostic purposes.

BMC10044E **SVCUPDTE service request for ESR SVC(*address*) failed, R15(*returncode*). *bmcpid***

Explanation: The attempt to insert the BMCP service routine into the SVC for the operating system has failed. *address* indicates the SVC address where the error occurred. *returncode* indicates the generated return code.

System Action: The BMCP initialization process is terminated.

User Response: Contact BMC Software Product Support. An SVC dump of the common areas (CSA, LPA, NUC, and SQA) may be required for diagnostic purposes.

BMC10060S ***text*, R15(*returncode*). *bmcpid***

Explanation: A non-BMC Software product may have overlaid the BMCP, or an internal subsystem error may have occurred. *returncode* indicates the generated return code.

System Action: The BMCP initialization process is terminated.

User Response: Check for any non-BMC Software product that may have overlaid the BMCP and correct any problems. If this is not applicable, contact BMC Software Product Support.

BMC10100I **Subsystem ready. *bmcpid***

Explanation: Initialization of the BMCP is complete, and the functions provided by the BMCP are available.

System Action: The BMCP is available.

User Response: This message is informational only.

BMC10106I Subsystem termination initiated. *bmcpid*

Explanation: The BMCP is performing a shutdown in response to an operator request. Termination of the BMCP is in progress.

System Action: The subsystem dispatcher terminates all activity within the BMCP. The shutdown process takes approximately two minutes to complete.

User Response: This message is informational only.

BMC10108S REQUESTED SUBSYSTEM STORAGE KEY CONFLICTS WITH PREVIOUS SUBSYSTEM KEY. *NEWKEY=newkey*, *OLDKEY=oldkey* *bmcpid*

Explanation: The BMCP was started with PGM=BPSMDSP n , but a previous BMCP initialization used a different subsystem protect key. The n value of BPSMDSP n indicates a particular subsystem protect key. *oldkey* and *newkey* indicate the previous value of n and the currently requested value of n .

System Action: The BMCP control program terminates.

User Response: An IPL is required to have the BMCP initialize with a new subsystem key that differs from a previous value. See the BMCP documentation regarding changing the subsystem key.

BMC10109W RECYCLING INITIALIZATION WITH PREVIOUS STORAGE KEY. *OLDKEY=oldkey*, *NEWKEY=newkey* *bmcpid*

Explanation: The BMCP was started with PGM=BPSMDSP n , but a previous BMCP initialization used a different subsystem protect key. The n value of BPSMDSP n indicates a particular subsystem protect key. *oldkey* and *newkey* indicate the previous value of n and the currently requested value of n .

System Action: The BMCP control program resets the subsystem protect key to the previous value and continues with initialization.

User Response: An IPL is required to have the BMCP initialize with a new subsystem key that differs from a previous value. See the BMCP documentation regarding changing the subsystem key.

BMC10202E Schedule of command into subsystem failed, R15(*returncode*). *bmcpid*

Explanation: An attempt to schedule an operator-initiated command into the BMCP was not successful due to a storage shortage. *returncode* indicates the generated return code.

System Action: The command is ignored.

User Response: Confirm that the BMCP is active, and issue the command again. If the condition persists, contact BMC Software Product Support.

BMC10204I Subsystem cancellation initiated. *bmcpid*

Explanation: The BMCP is performing a shutdown in response to an operator request. Termination of the BMCP is in progress.

System Action: The subsystem dispatcher terminates all activity within the BMCP. The shutdown process takes approximately two minutes to complete.

User Response: This message is informational only.

BMC10208W Command not processed (*reasoncode*). *bmcpid*

Explanation: An operator-initiated BMCP command was not scheduled. *reasoncode* indicates the generated reason code, which can be any of the following reasons:

- subsystem dispatcher not active
- no command text present
- cross memory link to subsystem failed
- function not successful

System Action: The BMCP ignores the command.

User Response: Restart the subsystem, and issue the command again.

BMC10210W Terminating subsystem ASID(*asid1*) inconsistent with dispatcher ASID(*asid2*) in SSVT. *bmcpid*

Explanation: This message is issued during end-of-task and/or end-of-memory processing and indicates that the BMCP being terminated does not match the BMCP previously in control of the subsystem vector table (SSVT). *asid1* and *asid2* indicate the inconsistent address space IDs.

System Action: The BMCP is flagged as terminated, but facilities provided by the subsystem continue.

User Response: No action is required.

BMC10219E Abend (*abendcode*) occurred during end-of-task processing. *bmcpid*

Explanation: The subsystem end-of-task resource manager abended during resource clean-up. *abendcode* indicates the generated abend code.

System Action: The requestor is notified that the request has abended.

User Response: This message provides error-reporting information that may be required by BMC Software Product Support if you need to report a problem.

BMC10220W **Terminating subsystem ASID(*asid1*) inconsistent with dispatcher ASID(*asid2*) in SSVT. *bmcpid***

Explanation: This message is issued during end-of-task and/or end-of-memory processing and indicates that the BMCP being terminated does not match the BMCP previously in control of the subsystem vector table (SSVT). *asid1* and *asid2* indicate the inconsistent address space IDs.

System Action: The BMCP is flagged as terminated, but facilities provided by the subsystem continue.

User Response: Restart the BMCP.

BMC10228I **Subsystem termination completed. *bmcpid***

Explanation: The BMCP has terminated and has been processed by the end-of-memory resource manager.

System Action: Processing continues without the BMCP.

User Response: This message is informational only.

BMC10229E **Abend (*abendcode*) occurred during end-of-memory processing. *bmcpid***

Explanation: The subsystem end-of-memory resource manager abended during resource clean-up. *abendcode* indicates the generated abend code.

System Action: The requestor is notified that the request has abended.

User Response: This message provides error-reporting information that may be required by BMC Software Product Support if you need to report a problem.

BMC10239E **Abend (*abendcode*) occurred during allocation processing. *bmcpid***

Explanation: The cross-memory environment allocation service ESTAE routine received control as a result of an abend. *abendcode* indicates the generated abend code.

System Action: The attempt to connect to the cross-memory environment for the subsystem is terminated. A non-zero return code is provided to the calling process.

User Response: Attempt to obtain a SLIP dump of this situation, and contact BMC Software Product Support. The SLIP dump should include the following specification: SDATA=(RGN,LSQA,NUC,SQA,CSA,LPA).

BMC10250W **LPA-module-create request abend *abendcode*. *bmcpid***

Explanation: The request to create a link pack area (LPA) module failed. *abendcode* indicates the generated abend code.

System Action: An SVC dump is placed into a SYS1.DUMP data set. The process requesting the LPA module create service issues a message.

User Response: Save the SVC dump, and contact BMC Software Product Support.

BMC10262E ***text*, R15(*returncode*). *bmcpid***

Explanation: A service request for an update procedure failed. A non-BMC Software product may have overlaid the subsystem, or an internal subsystem error may have occurred. *text* provides message text about the error. *returncode* indicates the generated return code.

System Action: The service request for an update is cancelled.

User Response: Check for and try to correct any non-BMC Software product that may have overlaid BMCP. If this is not applicable, contact BMC Software Product Support.

BMC10264W **SVCTABLE service request abend *completioncode*. *bmcpid***

Explanation: The routine that updates the SVCTABLE abended. *completioncode* indicates the generated completion code.

System Action: The update of the SVCTABLE fails.

User Response: Stop and restart the BMCP.

BMC10275E ***text* R15(*returncode*), *reasoncode*. *bmcpid***

Explanation: The BMCP allocation service routine was unable to allocate the address space for the requesting process. A non-BMC Software product may have overlaid the subsystem, or an internal subsystem error may have occurred. *text* provides message text about the error. *returncode* indicates the generated return code. *reasoncode* indicates the generated reason code.

System Action: The attempt to allocate address space is terminated. A non-zero return code is provided to the calling process.

User Response: Check for and try to correct any non-BMC Software product that may have overlaid BMCP. If this is not applicable, contact BMC Software Product Support.

BMC10276A ***jobname* in ASID (*asid*) waiting for primary subsystem restart. *bmcpid***

Explanation: The job has attempted to access the BMCP. The BMCP is not active, and the job is waiting for it to be restarted. *jobname* indicates the job waiting for execution. *asid* indicates the address space in which the job is running.

System Action: The task issuing the message is placed in a wait state until the subsystem is restarted. The message is deleted from the console when the job is notified of the availability of the system.

User Response: Restart the BMCP.

BMC10279E **Abend (*abendcode*) occurred during cross-memory connection processing. *bmcpid***

Explanation: The cross-memory connection ESTAE routine received control as a result of an abend. *abendcode* indicates the generated abend code.

System Action: The attempt to connect the cross-memory environment for the BMCP is terminated. A non-zero return code is provided to the calling process.

User Response: Attempt to obtain a slip dump of this situation, and contact BMC Software Product Support.

BMC10280I **Unexpected return code, subsystem(*subsysid*), R15(*returncode*), SSOBRETN(*reasoncode*). *bmcpid***

Explanation: The BMCP received a return code that is unusual, but which does not necessarily indicate an error. *subsysid* indicates the affected BCSS subsystem. *returncode* indicates the generated return code. *reasoncode* indicates the generated reason code.

System Action: The BMCP continues processing requests.

User Response: This message is informational only.

BMC10284S **Invalid XST(*address*) detected on queue. *bmcpid***

Explanation: Invalid information was detected in the XST (extended SVC screening) table. This may be the result of a non-BMC Software product overlaying part of the BMCP. *address* indicates the address of the invalid XST.

System Action: The BMCP terminates.

User Response: Restart the BMCP. If the condition persists, check for overlays of the BMCP, and contact BMC Software Product Support.

BMC10410E SXSS(address) failed SYNCH service validation. bmcpid

Explanation: A call to an SXSS address failed then validation process. The failure may be the result of a non-BMC Software product overlaying part of the BMCP or the SXSS. *address* indicates the address of the SXSS that failed.

System Action: The caller to the SXSS terminates, but the BMCP continues processing.

User Response: Stop and restart the BMCP. If the condition persists, check for overlays of the BMCP and/or contact BMC Software Product Support.

BMC10412E SXSS(address) not located within SYNCH service workarea. bmcpid

Explanation: A call to an SXSS address failed because the address was not in the work area. The failure may be the result of a non-BMC Software product overlaying part of the BMCP or the SXSS. *address* indicates the address of the missing SXSS.

System Action: The caller to the SXSS terminates, but the BMCP continues processing.

User Response: Stop and restart the BMCP. If the condition persists, check for overlays of the BMCP, and contact BMC Software Product Support.

BMC10700I Status, subsystem address space *asid* is active BMCP.

Explanation: If active, the BMCP returns this status message in response to the bmcpid STATUS command. If BMCP is not active, the log responds with a command error. *asid* indicates the address space of the BMCP.

System Action: BMCP processing continues.

User Response: This message is informational only.

BMC10790E Abend *abendcode* occurred during command processing. bmcpid

Explanation: The command processing manager for the BMCP (BPSMCPM0) abended while executing an operator-specified command. *abendcode* indicates the generated abend code.

System Action: Processing for the current command is terminated.

User Response: Contact BMC Software Product Support.

BMC11000–BMC11999

This section describes the messages in the range BMC11000 to BMC11999, which are generated by the BMC Consolidated Subsystem (BCSS) and the Extract program (BCSUEXTR).

- BCSS generates messages BMC11000 through BMC11899
- BCSUEXTR generates messages BMC11900 through BMC11999

BMC11000R Enter initialization options or “U”. *bcssid*

Explanation: The BCSS is prompting the MVS operator for options that control the method of initializing the environment.

System Action: The BMCP address space waits for an operator response.

User Response: The MVS system operator should respond with the proper initialization options. For example, reply ***xx,SUBSYSID=bcssid*** where *xx* is the identification number (id) and *bcssid* is the requested subsystem identifier. Reply **U** to use default values for any non-required parameters.

BMC11002E Invalid initialization option(s) specified. *bcssid*

Explanation: One or more of the initialization options specified are invalid. Previously specified options are retained.

System Action: The invalid option(s) are ignored.

User Response: See the User Response for message BMC11000R, which immediately follows this message, and respond with the action it requires.

BMC11006I SSCT(*address*) for subsystem obtained and queued. *bcssid*

Explanation: The BCSS issues this message when it has to construct and insert a subsystem control table (SSCT) into the SSCT queue. The subsystem was not previously active and an entry was not defined in any IEFSSNxx members of SYS1.PARMLIB. *address* indicates the SSCT address.

System Action: The BMCP initialization process continues.

User Response: This message is informational only.

- BMC11019S** **Abend (*abendcode*) occurred during subsystem address space initialization. *bcssid***
- Explanation:* An error occurred during the initialization of the BCSS. This message is issued during recovery processing and represents a severe error. *abendcode* indicates the generated abend code.
- System Action:* The BCSS initialization process terminates.
- User Response:* Contact BMC Software Product Support.
-
- BMC11020S** **Service module values not consistent with existing SSVT(*address*). *bcssid***
- Explanation:* You attempted to use an earlier version of your product after the latest version had been initialized. Internal control block and/or module attributes are incorrect. *address* indicates the subsystem vector table (SSVT) address where the inconsistency occurred.
- System Action:* The BCSS terminates.
- User Response:* Reinstall the latest BCSS. If this fails to solve the problem, contact BMC Software Product Support
-
- BMC11021S** **Subsystem down-leveling attempted, *oldlevel*, *newlevel*, SSVT(*address*). *bcssid***
- Explanation:* The BCSS tried to initialize at a lower level than previously active. The installed BCSS was created by a set of modules from the latest maintenance level. *oldlevel* and *newlevel* indicate the version and modification levels of the current modules and those being initialized. *address* indicates the SSVT address.
- System Action:* The BCSS address terminates.
- User Response:* Restore the latest version of the subsystem modules, and restart the BCSS.
-
- BMC11022S** **Consolidated subsystem address space(*asid*) is currently active. *bcssid***
- Explanation:* An attempt was made to start a BCSS, but it has detected that another BCSS is already active in another address space. *asid* indicates the hexadecimal address space ID of the active BCSS.
- System Action:* The BCSS initialization process terminates because another BCSS is available.
- User Response:* Continue using the BCSS that is already active.

- BMC11024S** **Service module FVT offset/size(*offset size*) not consistent with the SSVT(*offset size*). *bcssid***
- Explanation:* You attempted to use an earlier version of your product after the latest version had been initialized. Internal control block and/or module attributes are incorrect. *offset size* indicates the offset and size where the inconsistency occurred.
- System Action:* The BCSS address terminates.
- User Response:* Reinstall the latest BCSS. If this fails to solve the problem, contact BMC Software Product Support.
-
- BMC11026S** **Service module SVT offset/size(*offset size*) not consistent with the SSVT(*offset size*). *bcssid***
- Explanation:* You attempted to use an earlier version of your product after the latest version had been initialized. Internal control block and/or module attributes are incorrect. *offset size* indicates the offset and size where the inconsistency occurred.
- System Action:* The BCSS address terminates.
- User Response:* Reinstall the latest BCSS. If this fails to solve the problem, contact BMC Software Product Support.
-
- BMC11028S** **Service module PVT offset/size(*offset size*) not consistent with the SSVT(*offset size*). *bcssid***
- Explanation:* You attempted to use an earlier version of your product after the latest version had been initialized. Internal control block and/or module attributes are incorrect. *offset size* indicates the offset and size where the inconsistency occurred.
- System Action:* The BCSS address terminates.
- User Response:* Reinstall the latest BCSS. If this fails to solve the problem, contact BMC Software Product Support.
-
- BMC11029S** **Abend *abendcode* occurred during subsystem interface initialization. *bcssid***
- Explanation:* An abend occurred during the initialization phase of the BCSS. *abendcode* indicates the generated abend code.
- System Action:* The BCSS initialization process is terminated.
- User Response:* Contact BMC Software Product Support.

BMC11030S ***text R15(returncode), reasoncode. bcssid***

Explanation: An unexpected non-zero return code or reason code was received in response to a BCSS verify request. A non-BMC Software product may have overlaid the BMCP, or an internal subsystem error may have occurred. *text* provides message text about the error. *returncode* and *reasoncode* indicate the generated return code and reason code.

System Action: The BCSS is terminated.

User Response: Check for any non-BMC Software product that may have overlaid BCSS, and try to correct the problem. If this is not applicable, contact BMC Software Product Support.

BMC11032S ***Invalid subsystem affinity index(index) returned by subsystem verify. bcssid***

Explanation: An illogical subsystem affinity index was returned by the master MVS subsystem for the BCSS. *index* indicates the affinity index in hexadecimal format.

System Action: The BCSS is terminated.

User Response: Contact BMC Software Product Support.

BMC11034S ***Default subsystem name(name) not consistent with subsystem being initialized. bcssid***

Explanation: An attempt was made to change the name of the BCSS during a restart operation. *name* indicates the default name of the BCSS.

System Action: The BCSS is terminated.

User Response: Restart the BCSS, and specify the correct SUBSYSID parameter. Use the subsystem ID that is appended to this message.

BMC11036R**Subsystem address space ASID(*asid*) has not terminated. *bcssid***

Explanation: The BCSS issuing the message detected that the subsystem control areas are already owned by the currently-executing address space. The currently-executing address space must terminate for the starting subsystem to continue. *asid* indicates the address space ID in hexadecimal format of the currently-executing address space.

System Action: Ownership determination checks occur every ten seconds while this message is outstanding. When the ownership changes and the subsystem control areas become available to the starting address space, the message is deleted and initialization continues.

User Response: Respond according to one of the following situations:

- Multiple subsystem address spaces have been started in error. In this situation, type **T** to terminate the issuing subsystem address space.
- This message is issued for a restarted subsystem when the MVS resource management routines are performing significant end-of-memory processing in the MVS *MASTER* address space and the clean-up of a terminated subsystem address space has not concluded. This can occur during maximum CPU usage. Wait for the message to be deleted, which usually occurs within 30 seconds. If the issuing subsystem address space does not start, or this message remains longer than 30 seconds, contact BMC Software Product Support.

BMC11037S**A Public subsystem is already initialized, only one Public subsystem may be active. *bcssid***

Explanation: An attempt was made to start another BCSS using the MODE=PUBLIC parameter while another public consolidated subsystem was already active.

System Action: The BCSS initialization process terminates.

User Response: If you are attempting to start a private consolidated subsystem, use the MODE=PRIVATE parameter.

BMC11038S**Subsystem initialization process terminated. *bcssid***

Explanation: The initialization process is being terminated due to one or more previously-reported errors. Messages have been issued identifying the problem(s) encountered during the initialization process.

System Action: The BCSS terminates.

User Response: Follow the User Response for message(s) issued prior to this message.

BMC11039S A public subsystem is attempting to initialize as private.

Explanation: An attempt is being made to initialize a public consolidated subsystem as a private consolidated subsystem.

System Action: The initialization process is terminated.

User Response: Correct the subsystem name, and submit the request again.

BMC11040E *ddname*,registration data set has reached an excessive number of extents and must be rebuilt. *bcssid*

Explanation: A registration data set has reached a number of extents that the subsystem control record (SCR) describing the registration data set cannot support. *ddname* indicates the DD statement associated with the registration data set.

System Action: If the indicated registration data set is the primary registration data set or the last available duplex registration data set, processing terminates.

User Response: Rebuild the indicated registration data set, combining the extents into as few extents as possible.

BMC11041W *ddname*,registration data set is nearing an excessive number of extents and should be rebuilt. *bcssid*

Explanation: A registration data set is nearing a number of extents that the subsystem control record (SCR) describing the registration data set cannot support. *ddname* indicates the DD statement associated with the registration data set.

System Action: The BCSS continues execution. It is possible that the BCSS could fail if the registration data set extents exceed the SCR limit during execution.

User Response: Rebuild the indicated registration data set; combine the extents into as few extents as possible.

BMC11042I Registration data set synchronization will not be performed in parallel. *bcssid*

Explanation: Your product cannot use parallel synchronization for the registration data set. Serial synchronization is used.

System Action: Processing continues with serial synchronization.

User Response: This message is informational only.

BMC11043E ***ddname, does not reference a valid VSAM data set. bcssid***

Explanation: A DD statement does not reference a valid VSAM data set. *ddname* indicates the DD statement associated with the registration data set.

System Action: The DD statement is ignored, and processing continues.

User Response: Correct the DD statement for the next initialization.

BMC11044E ***ddname, all processing will be bypassed for this data set. bcssid***

Explanation: An error was encountered during initialization processing for the registration data set. Another message immediately precedes this message explaining the problem encountered. *ddname* indicates the DD statement associated with the registration data set.

System Action: The registration data set is closed, and all other processing for the data set is bypassed.

User Response: Refer to the Explanation of the previously-issued message for a description of the problem and a possible solution.

BMC11045E ***ddname, the defined record length matches the defined CI size. bcssid***

Explanation: When a registration data set was defined, the logical record length and the control interval size were identical. *ddname* indicates the DD statement associated with the registration data set.

System Action: The registration data set is closed, and all other processing for the data set is bypassed.

User Response: The logical record length must be defined with a smaller value than the control interval size. Use IDCAMS to delete and redefine the registration data set with proper values.

BMC11046S ***No usable duplex registration data set available. bcssid***

Explanation: The initialization process was not successful in establishing a duplex registration data set.

System Action: The BCSS terminates automatically.

User Response: Refer to the Description of the previously-issued message to determine why the duplex registration data set failed during initialization processing.

BMC11047E ***ddname*, record size exceeds that defined for this data set. *bcssid***

Explanation: The maximum logical record length defined for the specified registration data set is less than the maximum logical record length required by BCSS. *ddname* indicates the DD statement associated with the registration data set.

System Action: The registration data set is closed and other processing bypassed.

User Response: Use IDCAMS to delete and redefine the registration data set with the maximum logical record length recommended in the installation instructions.

BMC11048E ***ddname*, key size does not match that defined for this data set. *bcssid***

Explanation: The key length defined for the specified registration data set is not correct. The key length is predetermined and cannot be altered from the length defined during installation. *ddname* indicates the DD statement associated with the registration data set.

System Action: The registration data set is closed and other processing bypassed.

User Response: Use IDCAMS to delete and redefine the registration data set with the key length specified during installation.

BMC11049E ***ddname*, relative key position does not match that defined for this data set. *bcssid***

Explanation: The relative key position (key offset) for the specified registration data set is not correct. The starting key position is predetermined and cannot be altered from that defined during installation. *ddname* indicates the DD statement associated with the registration data set.

System Action: The registration data set is closed and other processing bypassed.

User Response: Use IDCAMS to delete and redefine the registration data set with the relative key position specified during installation.

BMC11050S ***ddname*, ACB OPEN FAILED, R15(*returncode*), ACBERFLG(*errorflag*), <INITIAL | FINAL>. *bcssid***

Explanation: An attempt to open the ACB of a registration data set was not successful. The IEC161I messages are issued by the MVS operating system. *ddname* indicates the DD statement associated with the registration data set. *returncode* indicates the generated return code. *errorflag* indicates the generated ACB error flag.

System Action: Processing of the registration data set is bypassed.

User Response: Consult the IBM documentation for an explanation of message IEC161I, and resolve the problem with the cluster. If the problem cannot be resolved, consult BMC Software Product Support.

BMC11051S***ddname* is not synchronized and has been dropped, RIBRFLAGS(*errorflag*). *bcssid***

Explanation: A registration data set was not synchronized; your product will not use the registration data set. This message is preceded by other BMC Software messages that further explain the problem. *ddname* indicates the DD statement associated with the registration data set. *errorflag* indicates the generated error flag, which will assist BMC Software Product Support in determining the reason for the condition.

System Action: The BCSS initialization process continues.

User Response: Contact BMC Software Product Support if you are not able to correct the problem using the information obtained from the other BMC Software messages.

BMC11052W**Primary registration data set not provided. *bcssid***

Explanation: You did not provide a primary registration data set during initialization.

System Action: The BCSS continues without use of registration data sets.

User Response: If it was not your intent to run without registration data sets, correct the DD statement during the next initialization.

BMC11053S***ddname*,<TESTCB | SHOWCB> request failed, r15(*returncode*). *bcssid***

Explanation: A VSAM service request failed with a non-zero return code. A non-BMC Software product may have overlaid the subsystem, or an internal subsystem error may have occurred. *ddname* indicates the DD statement associated with the problem. *returncode* indicates the generated return code.

System Action: Processing of the registration data set is bypassed.

User Response: Check for any non-BMC Software product that may have overlaid BCSS, and attempt to correct the problem. If this is not applicable, contact BMC Software Product Support.

BMC11054S ***ddname*, VSAM error, R15(*returncode*), RPLREQ(*requesttype*), RPLFDBK(*feedback*), RPLOPTCD(*optioncode*), R14(*offset*). *bcssid***

Explanation: A VSAM logical record mode request for a registration data set was not successful. A non-BMC Software product may have overlaid the subsystem, or an internal subsystem error may have occurred. *ddname* indicates the DD statement associated with the problem. *returncode* indicates the generated return code. *requesttype*, *feedback*, *optioncode*, and *offset* indicate values associated with the request. These values are shown in hexadecimal format. Use a VSAM Logic manual from IBM to interpret the values.

System Action: Processing of the registration data set is bypassed.

User Response: Check for any non-BMC Software product that may have overlaid BCSS, and attempt to correct the problem. If this is not applicable, contact BMC Software Product Support.

BMC11055I ***ddname*, reusable duplexing registration data set initialized. *bcssid***

Explanation: A reusable duplexing registration data set has been initialized with the contents of the registration data set. *ddname* indicates the DD statement associated with the registration data set.

System Action: Subsystem initialization continues.

User Response: This message is informational only.

BMC11056I ***ddname*, duplexing registration data set initialized. *bcssid***

Explanation: A duplexing registration data set has been initialized with the contents of the registration data set. *ddname* indicates the DD statement associated with the registration data set.

System Action: The BCSS initialization process continues.

User Response: This message is informational only.

BMC11057I ***ddname*, primary registration data set <synchronized | initialized>. *bcssid***

Explanation: A primary registration data set has been synchronized or initialized. *ddname* indicates the DD statement associated with the registration data set.

System Action: The BCSS initialization process continues.

User Response: This message is informational only.

- BMC11059E** **<DSPSERV | ALESERV> service failed for <GENERIC | PROGRAM | DATA SET> dataspace, R15(H'*returncode*'), R0(X'*returncode*'), dataspace will not be used. *bcssid***
- Explanation:* The DSPSERV or ALESERV operating system macro has failed for generic, program, or data set data space. *returncode* indicates the generated return codes. The return codes are in hexadecimal format.
- System Action:* Processing continues, but your product does not use the data space.
- User Response:* Contact your systems programmer to determine the meaning of the DSPSERV and ALESERV return codes.
-
- BMC11060S** ***servicename* service request failed R15(*returncode*).**
- Explanation:* This message is issued during cross-memory environment initialization when an MVS services fails with a non-zero return code. *servicename* indicates the service that failed: AXRES, AXSET, LXRES, ETCON, and ETCRE. *returncode* indicates the generated return code.
- System Action:* The initialization process is terminated.
- User Response:* Contact BMC Software Product Support.
-
- BMC11061W** **<GENERIC | PROGRAM | DATA SET> caching dataspace size of *megs1*MB was not used. *megs2*MB dataspace was allocated. *bcssid***
- Explanation:* You attempted to allocate a caching data space that was too large. A smaller data space was allocated for the generic, program, or data set data space. *megs1* indicates the requested megabytes (MB). *megs2* indicates the allocated MB.
- System Action:* Subsystem initialization continues.
- User Response:* In the future, specify a smaller data space size.
-
- BMC11062I** **<GENERIC | PROGRAM | DATA SET> caching dataspace initialized. *bcssid***
- Explanation:* The generic, program, or data set caching data space has initialized.
- System Action:* The system initializes a data space.
- User Response:* This message is informational only.

BMC11063E ***ddname2,registration data set resides on the same device as ddname1. bcssid***

Explanation: Only one registration data set can reside on a device. *ddname1* and *ddname2* indicate the DD statements associated with the registration data sets in question.

System Action: Processing continues with *ddname1* only. *ddname2* is ignored.

User Response: Allocate registration data sets on separate devices. Correct the BCSS JCL, and recycle the subsystem.

BMC11064E ***ddname2,registration data set references the same cluster as ddname1. bcssid***

Explanation: You have duplicated the name of a registration data set. Each registration data set must have a unique name. *ddname1* and *ddname2* indicate the DD statements associated with the registration data sets in question.

System Action: Processing continues with *ddname1* only. *ddname2* is ignored.

User Response: Correct the DD statements before the next subsystem initialization.

BMC11065E ***ddname,registration data set resides on multiple volumes and will not be used. bcssid***

Explanation: *ddname* references a data set that is a multivolume data set. Registration data sets must reside on a single volume; you cannot use multivolume data sets. *ddname* indicates the DD statement associated with the registration data set.

System Action: Processing continues without the registration data set in error.

User Response: Allocate registration data sets as single volume data sets.

BMC11066E ***ddname,System Control Record (SCR) construction failed, R15(H'returncode'). REGSET is not usable. bcssid***

Explanation: Construction has failed for the system control record (SCR) of a registration data set. *ddname* indicates the DD statement associated with the registration data set. *returncode* indicates the generated return code in hexadecimal format. The registration data set cannot be used.

System Action: Processing continues without the registration data set in error.

User Response: Contact BMC Software Product Support.

- BMC11067I** ***ddname* will be used to synchronize or initialize other REGISETs.
*bcssid***
- Explanation:* An initialized registration data set is used to synchronize or initialize other registration data sets. *ddname* indicates the DD statement associated with the registration data set that will be used in the synchronization.
- System Action:* Processing continues.
- User Response:* This message is informational only.
-
- BMC11068E** ***ddname* could not be initialized and can not be used to synchronize
other regisets. *bcssid***
- Explanation:* A registration data set could not be initialized and cannot be used to synchronize or initialize other registration data sets. *ddname* indicates the DD statement associated with the registration data set that could not be initialized. This message is preceded by other BMC Software messages that further explain the problem.
- System Action:* Processing continues without the registration data set in error.
- User Response:* Contact BMC Software Product Support if you are not able to correct the problem using the information obtained from the other BMC Software messages.
-
- BMC11069S** **Abend *abendcode* occurred during cross-memory environment
initialization. *bcssid***
- Explanation:* An abend occurred during cross-memory environment initialization. *abendcode* indicates the generated abend code.
- System Action:* The BCSS initialization process is terminated.
- User Response:* Contact BMC Software Product Support.
-
- BMC11071I** **<GENERIC | PROGRAM | DATA SET> dataspace primed with *recs*
records. *bcssid***
- Explanation:* The generic, program, or data set data space has been primed with a number of records. *recs* indicates the number of records used to prime the data space.
- System Action:* Processing continues.
- User Response:* This message is informational only.

BMC11072E ***ddname*,registration data set is in use by another subsystem on this MVS image. *bcssid***

Explanation: A data set has been allocated and initialized as a registration data set by another BMC Software consolidated subsystem running on the same MVS image. Registration data sets can be used by only one BMC Software consolidated subsystem per MVS image. *ddname* indicates the DD statement associated with the registration data set.

System Action: The BCSS initialization process continues without the registration data set.

User Response: Correct the allocation JCL. Restart the BCSS using registration data sets that are not shared by other BCSSs on the same MVS image.

BMC11073S **Subsystem address space initialization is being terminated due to missing REGISETs. *bcssid***

Explanation: Initialization of a BCSS cannot continue without the necessary registration data sets.

System Action: The BCSS initialization process is terminated.

User Response: Correct the REGISET problem before or with the next initialization.

BMC11079S **Abend *abendcode* occurred during commands data set processing. *bcssid***

Explanation: The subsystem commands processor abended while processing the commands data set. *abendcode* indicates the generated abend code.

System Action: The current command is not processed.

User Response: Contact BMC Software Product Support with the PSW and registers.

BMC11092I ***ddname* System Control Record checkpoint completed. *bcssid***

Explanation: A registration data set system control record (SCR) has been successfully written to DASD. *ddname* indicates the DD statement associated with the registration data set.

System Action: Processing continues.

User Response: This message is informational only.

BMC11100I Subsystem ready BCSS. *bcssid*

Explanation: Initialization of the BCSS is complete, and the services provided by the BCSS are available.

System Action: The subsystem dispatcher waits for operator input.

User Response: This message is informational only.

BMC11102A Waiting for BMC primary subsystem initialization. *bcssid*

Explanation: During BCSS initialization processing, the BCSS detected that a service provided by the BMC Software primary subsystem (BMCP) is required but the subsystem has not been started.

System Action: The BCSS waits for a BMCP to be started.

User Response: Start a BMCP. If you are not familiar with the process, refer to “Starting BMCP” on page 2-3.

BMC11106I Subsystem termination initiated. *bcssid*

Explanation: The BCSS is performing a shutdown in response to an operator request. Termination of the BCSS is in progress.

System Action: The subsystem dispatcher terminates all activity within the address space. The shutdown process takes approximately two minutes to complete.

User Response: This message is informational only.

BMC11160E *ddname*, request failed, RPLREQ(*requesttype*), RPLFDBK(*feedback*), RPLOPTCD(*optioncode*). *bcssid*

Explanation: A VSAM logical record mode request for a primary registration data set was not successful. The detected error was either a logical or physical error. *ddname* indicates the DD statement associated with the problem. *requesttype*, *feedback*, and *optioncode* indicate values associated with the request. These values are shown in hexadecimal format. Refer to a VSAM Logic manual from IBM to interpret the values.

System Action: The request to the registration data set is terminated.

User Response: Contact BMC Software Product Support.

BMC11162E ***ddname*, request failed, RPLREQ(*requesttype*), RPLFDBK(*feedback*), RPLOPTCD(*optioncode*). *bcssid***

Explanation: A VSAM logical record mode request for a primary registration data set was not successful. The detected error was either a logical or physical error. *ddname* indicates the DD statement associated with the problem. *requesttype*, *feedback*, and *optioncode* indicate values associated with the request. These values are shown in hexadecimal format. Refer to a VSAM Logic manual from IBM to interpret the values.

System Action: The request to the registration data set is terminated. Your product produces an additional message explaining what actions, if any, are being produced by the failure.

User Response: Contact BMC Software Product Support.

BMC11164E ***ddname*, duplex failed, RPLREQ(*requesttype*), RPLFDBK(*feedback*), RPLOPTCD(*optioncode*). *bcssid***

Explanation: A VSAM logical record mode request for a duplex registration data set was not successful. The detected error was either a logical or physical error. *ddname* indicates the DD statement associated with the problem. *requesttype*, *feedback*, and *optioncode* indicate values associated with the request. These values are shown in hexadecimal format. Use a VSAM Logic manual from IBM to interpret the values.

System Action: The request to the registration data set is terminated. Your product produces an additional message explaining what actions, if any, are being produced by the failure.

User Response: Contact BMC Software Product Support.

BMC11166E ***ddname*, registration data set has reached an excessive number of extents and must be rebuilt. *bcssid***

Explanation: A registration data set has reached a number of extents that the subsystem control record (SCR) describing the registration data set cannot support. This message is issued by BCSMSRAM during subsystem initialization. *ddname* indicates the DD statement associated with the registration data set.

System Action: If the indicated registration data set is the primary registration data set or the last available duplex registration data set, processing terminates.

User Response: Rebuild the indicated registration data set, combining the extents into as few extents as possible.

BMC11167W **<GENERIC | PROGRAM | DATA SET> dataspace capacity has been exceeded. Record caching will no longer be performed. *bcssid***

Explanation: Record caching terminates when the product exceeds the generic, program, or data set data space capacity. This message indicates a logic error.

System Action: Processing continues without caching.

User Response: Contact BMC Software Product Support.

BMC11168E **TERMRPL request failed, R15(*returncode*), R0(*reasoncode*), RPLREQ(*requesttype*), RPLFDBK(*feedback*), RPLOPTCD(*optioncode*). *bcssid***

Explanation: A VSAM TERMRPL request failed. A non-BMC Software product may have overlaid the BCSS, or an internal subsystem error may have occurred. *returncode* indicates the generated return code. *reasoncode* indicates the generated reason code. *requesttype*, *feedback*, and *optioncode* indicate values associated with the request. These values are shown in hexadecimal format. Refer to a VSAM Logic manual from IBM to interpret the values.

System Action: The request to terminate a VSAM I/O request ended with the indicated error.

User Response: Check for any non-BMC Software product that may have overlaid the BCSS, and attempt to correct the problem. If this is not applicable, contact BMC Software Product Support.

BMC11169E **Duplex REGISets unusable. Consolidated subsystem terminating. *bcssid***

Explanation: A physical error has occurred on the last, or only, duplex registration data set. This message should be preceded with the message BMC11168E, which indicates the specific error that occurred.

System Action: The BCSS terminates so that the problem with the affected registration data set(s) can be corrected. The BCSS indicates that Failsoft mode is active and that all potentially compressed data should be processed by the expansion routines.

Unless previously disabled from doing so, the BCSS provides the following information to any BMC Software product requesting its service:

- Failsoft mode is active.
- All potentially-compressed data should be processed by the expansion routines.

User Response: Correct the problem with the duplex registration data set(s), then restart the BCSS.

BMC11171E ***ddname* being dropped from processing, continuing with remaining primaries. *bcssid***

Explanation: An I/O error or initialization error (for example, mismatched REGISETs) caused this REGISET to be out of sync with the others. The run may continue if there are sufficient primary registration data sets remaining. *ddname* indicates the DD statement associated with the registration data set that has dropped.

System Action: The named REGISET is excluded from further processing. Run continues.

User Response: Correct to reason for the REGISET failure, and recycle the subsystem.

BMC11173E ***ddname*,REGISET dropped for reason(X'*reason*'),*reasontext* is now down-level. *bcssid***

Explanation: A registration data set has been dropped. *ddname* indicates the DD statement associated with the registration data set that has dropped. *reason* and *reasontext* indicate the reason the registration data set has dropped in hexadecimal and character formats. This message indicates a serious problem.

System Action: Your product takes a slip dump (SDUMP), and processing continues.

User Response: Contact BMC Software Product Support.

BMC11174W **No usable duplexing REGISETs are available. *bcssid***

Explanation: No duplex registration data sets are available for use.

System Action: Processing continues without duplex registration data sets.

User Response: If it was not your intent to run without duplex registration data sets, reinitialize the BCSS and include the required registration data sets.

BMC11175E **No primary REGISETs are usable. Duplexing REGISETs have been disabled. *bcssid***

Explanation: No primary registration data sets are available for use. All primary registration data sets have been dropped.

System Action: The BCSS terminates, and your product goes into Failsoft mode.

User Response: This message is preceded by other BMC Software messages that explicitly state the reason(s) for the registration data set drops. Also, check the Logrec and the SDSF Log for messages showing reasons for registration data set drops. Contact BMC Software Product Support if reasons do not explain the drops.

BMC11176S No REGSETs are usable. Subsystem dispatcher has been posted for termination. *bcssid*

Explanation: No registration data sets are available for use. All registration data sets have been dropped.

System Action: The BCSS terminates, and your product goes into Failsoft mode.

User Response: This message is preceded by other BMC Software messages that explicitly state the reason(s) for the registration data set drops. Also, check the Logrec and the SDSF Log for messages showing reasons for registration data set drops. Contact BMC Software Product Support if the messages do not adequately explain the drops.

BMC11180W Recycling initialization with previous storage key. *Oldkey=oldkey, Newkey=newkey bcssid*

Explanation: The BCSS was started with PGM=BCSMDSP n , but a previous BCSS initialization used a different subsystem protect key. The n value of BCSMDSP n indicates a particular subsystem protect key. *oldkey* and *newkey* indicate the previous value of n and the currently requested value of n .

System Action: The BCSS control program resets the subsystem protect key to the previous value and continues with initialization.

User Response: An IPL is required to have the BCSS initialize with a new subsystem key that differs from a previous value. See the BCSS documentation regarding changes to the subsystem key.

BMC11181S Requested subsystem storage key conflicts with previous subsystem key. *Newkey=newkey, Oldkey=oldkey bcssid*

Explanation: The BCSS was started with PGM=BCSMDSP n , but a previous BCSS initialization used a different subsystem protect key. The n value of BCSMDSP n indicates a particular subsystem protect key. *oldkey* and *newkey* indicate the previous value of n and the currently requested value of n .

System Action: The BCSS control program terminates.

User Response: An IPL is required to have the BCSS initialize with a new subsystem key that differs from a previous value. See the BCSS documentation regarding changes to the subsystem key.

BMC11202E Schedule of command into subsystem failed, R15(reasoncode). bcssid

Explanation: An attempt to schedule the operator-initiated command in the BCSS was not successful. The failure is normally due to storage limitations. *reasoncode* indicates the generated reason code.

System Action: The command is ignored.

User Response: Ensure that the BCSS is active, and issue the command again. If the condition persists, contact BMC Software Product Support.

BMC11204I Subsystem cancellation initiated. bcssid

Explanation: The BCSS is terminating as the result of an operator termination request.

System Action: Termination of the BCSS continues.

User Response: This message is informational only.

BMC11208W Command not processed. bcssid

Explanation: An operator-initiated BCSS command was not scheduled because the BCSS is not active.

System Action: The command is ignored.

User Response: Restart the BCSS, and issue the command again.

BMC11210W Terminating subsystem ASID(aside1) inconsistent with dispatcher ASID(aside2). bcssid

Explanation: The BCSS being terminated does not match the subsystem address space previously in control of the SSVT. This message is issued during end-of-task processing when a storage-overlay occurs. *aside1* and *aside2* indicate the address space IDs that are inconsistent.

System Action: Processing continues. The BCSS is flagged as terminated. The end-of-memory subsystem routines will correct this problem.

User Response: Restart the BCSS.

BMC11212W **Link for jobname(*jobname*) in ASID(*asid*) broken due to CAS(*casid*) failure. *bcssid***

Explanation: The end-of-task resource manager detected that the specified CAS ended. A message is issued for each address space that is using the terminating CAS. *jobname* indicates the name of a job that was using the address space. *asid* indicates the address space in which the job was running. *casid* indicates the CAS that has failed.

System Action: Termination of the CAS continues. The affected user address space attempts to allocate a different CAS when compression and/or expansion is to be performed.

User Response: No action is required.

BMC11214W **Service task failure, request(*requesttype*), job(*jobname*), ASID(*asid*), TCB(*address*). *bcssid***

Explanation: A subsystem service task was processing a request on behalf of a user-initiated process when the BCSS terminated. *requesttype* indicates the type of request that was processing. *jobname* indicates the name of the job that was processing. *asid* indicates the address space in which the job was processing. *address* indicates the address of the task control block (TCB).

System Action: The requestor is notified that the request had abended.

User Response: Issue the request again once the BCSS is available.

BMC11217E **Abend *abendcode* occurred during end-of-task processing. *bcssid***

Explanation: The subsystem end-of-task resource manager abended during resource clean-up. *abendcode* indicates the generated abend code.

System Action: The resource management process is attempted again. Clean-up of resources within the address space may not be completed when message BMC11219E is issued *after* this message.

User Response: This message provides error-reporting information that may be required by BMC Software Product Support if you need to report a problem.

BMC11219E **Abend *abendcode* occurred during end-of-task processing. *bcssid***

Explanation: The subsystem end-of-task resource manager abended during resource clean-up. *abendcode* indicates the generated abend code.

System Action: The resource management process is attempted again. Clean-up of resources within the address space may not be completed when this message is issued.

User Response: This message provides error-reporting information that may be required by BMC Software Product Support if you need to report a problem.

- BMC11220W** **Terminating subsystem ASID(*asid1*) inconsistent with dispatcher ASID(*asid2*). *bcssid***
- Explanation:* This message is issued during end-of-task and/or end-of-memory processing and indicates that the BCSS being terminated does not match the subsystem address space previously in control of the subsystem vector table (SSVT). *asid1* and *asid2* indicate the inconsistent address space IDs.
- System Action:* The BMCP address space is flagged as terminated, but facilities provided by the subsystem continue.
- User Response:* Restart the BMCP.
-
- BMC11222W** **Link for *jobname(jobname)* in ASID(*asid*) broken due to CAS(*casid*) failure. *bcssid***
- Explanation:* The end-of-task resource manager detected that the specified CAS ended. A message is issued for each address space that is using the terminating CAS. *jobname* indicates the name of a job that was using the address space. *asid* indicates the address space in which the job was running. *casid* indicates the CAS that has failed.
- System Action:* Termination of the CAS continues. The affected user address space attempts to allocate a different CAS when compression and/or expansion is to be performed.
- User Response:* No action is required.
-
- BMC11227E** **Abend *abendcode* occurred during end-of-memory processing. *bcssid***
- Explanation:* The subsystem end-of-memory resource manager abended during resource clean-up. *abendcode* indicates the generated abend code.
- System Action:* The resource management process is retried. Clean-up of the common area resources may not be completed when message BMC11229E is issued after this message.
- User Response:* This message provides error-reporting information that may be required by BMC Software Product Support if you need to report a problem.
-
- BMC11228I** **Subsystem termination completed. *bcssid***
- Explanation:* The subsystem address space has been terminated and processed by the end-of-memory or end-of-task resource manager.
- System Action:* Processing continues without an active BCSS.
- User Response:* This message is informational only.

BMC11229E *Abend **abendcode occurred during end-of-memory processing. **bcssid*****

Explanation: The subsystem end-of-memory resource manager abended during resource clean-up. *abendcode* indicates the generated abend code.

System Action: The resource management process is retried. Clean-up of the common area resources may not be completed when this message is issued.

User Response: This message provides error-reporting information that may be required by BMC Software Product Support if you need to report a problem.

BMC11238I *ASID(asid**) connected to cross-memory environment. **bcssid*****

Explanation: An address space has been dynamically connected to cross-memory environment of the BCSS. *asid* indicates the ID of the address space.

System Action: Control returns to the problem-state caller of the cross-memory connection service routine.

User Response: This message is informational only.

BMC11239E *Abend **abendcode occurred during allocation processing. **bcssid*****

Explanation: The cross-memory environment allocation service ESTAE routine received control as a result of an abend. *abendcode* indicates the generated abend code.

System Action: The attempt to connect to cross-memory environment of the subsystem is terminated.

User Response: Attempt to obtain a slip dump (SDUMP) of this situation, and contact BMC Software Product Support. The SDUMP should include the following specification: SDATA=(RGN, LSQA, NUC, SQA, CSA, LPA).

BMC11240A *jobname** in ASID(**asid**) waiting for consolidated subsystem restart. **bcssid*****

Explanation: A job attempted to access the BCSS. The BCSS is not active, and the job is waiting for BCSS to be restarted. *jobname* indicates the name of the job waiting for BCSS facilities. *asid* indicates the address space in which the job is executing.

System Action: The task issuing the message is in the wait state until the BCSS restarts. The message is deleted from the console when the job is notified of the availability of the BCSS.

User Response: Restart the BCSS.

BMC11250E ASCB(*ascb*) validation failed for secondary ASID(*asid*), FREEMAIN(*returncode*). *bcssid*

Explanation: Verification of the secondary address space was unsuccessful because the user address space was terminated. The storage is not freed if the generated return code is not zero. *ascb* indicates the type of validation. *asid* indicates the address of the secondary address space. *returncode* indicates the generated return code from the FREEMAIN of the process-related CSA.

System Action: Processing continues without accessing the user address space.

User Response: If the user address space is still active, contact BMC Software Product Support.

BMC11252E Requesting TCB(*tcbaddress*) /RB(*rbaddress*), ASID(*asid*), job(*jobname*), not suspended, FREEMAIN(*returncode*). *bcssid*

Explanation: Posting of the user process was not successful because the process was not in the wait state. The storage is not freed if the generated return code is not zero. *tcbaddress* is the address of the task control block (TCB). *rbaddress* is the address of the request block (RB). *asid* indicates the address space of the job. *jobname* indicates the name of the job making the request. *returncode* indicates the generated return code from the FREEMAIN of the process-related CSA.

System Action: The process is terminated.

User Response: If the user address space is still active, contact BMC Software Product Support.

BMC11253E Resume process memory access failure, ASID(*asid*), job(*jobname*), SPL(*spladdress*). *bcssid*

Explanation: A memory access failure occurred when the BCSS was unable to access the address space that had attempted to OPEN or CLOSE an ACB. This error represents a significant failure in MVS or your product's services. The error is probably due to the swapping out of the address space even when a DONTSWAP SYSEVENT was issued to prevent such an occurrence. This message is issued in conjunction with another BMC Software message. *asid* indicates the address space in which the job is executing. *jobname* indicates the name of the job. *spladdress* indicates the SPL address.

System Action: The address space that opened or closed the ACB remains in the suspended state until it is cancelled.

User Response: Attempt to obtain a slip dump (SDUMP) of this situation, and contact BMC Software Product Support. The SDUMP should include the following specification: SDATA=(RGN, LSQA, NUC, SQA, CSA, LPA).

BMC11260A ***jobname* in ASID(*asid*) waiting for compression address space. *bcssid***

Explanation: A job attempted to access a CAS but none was available. The job is now waiting for the BCSS to start a new CAS. *jobname* indicates the name of the job waiting to execute. *asid* indicates the address space in which the job is executing.

System Action: The task attempting to access the CAS is placed in the wait state until the BCSS starts the new CAS. The message is deleted from the console when the CAS is available for use.

User Response: Ensure a CAS is started by the BCSS.

BMC11269E **Abend *abendcode* occurred during CAS allocation processing. *bcssid***

Explanation: The CAS allocation service ESTAE routine received control as a result of an abend. *abendcode* indicates the generated abend code.

System Action: The attempt to allocate a CAS is terminated. A non-zero return code is provided to the calling process.

User Response: Attempt to obtain a slip dump of this situation, and contact BMC Software Product Support.

BMC11275E **Allocation request failed, ASID(*asid*), R15(*returncode*).
reason(*reasoncode*)**

Explanation: The CAS allocation service routine was unable to successfully allocate a CAS for the requesting process. A non-BMC Software product may have overlaid the subsystem, or an internal subsystem error may have occurred. *asid* indicates the address space in which the request was made. *returncode* indicates the generated return code. *reasoncode* indicates the generated reason code.

System Action: The attempt to allocate a CAS is terminated. A non-zero return code is provided to the calling process.

User Response: Check for any non-BMC Software product that may have overlaid the BCSS, and attempt to correct the problem. If this is not applicable, contact BMC Software Product Support.

BMC11276A ***jobname* in ASID(*asid*) waiting for consolidated subsystem restart.
*bcssid***

Explanation: A job attempted to access the BCSS. The BCSS is not active, and the job is waiting for it to be restarted. *jobname* indicates the name of the job waiting to execute. *asid* indicates the address space in which the job is executing.

System Action: The job issuing the message is placed in the wait state until the BCSS restarts. The message is deleted from the console when the job is notified of the BCSS availability.

User Response: Restart the BCSS.

- BMC11279E** **Abend *abendcode* occurred during cross-memory connection processing. *bcssid***
- Explanation:* The cross-memory connection ESTAE routine received control as a result of an abend. *abendcode* indicates the generated abend code.
- System Action:* The attempt to connect the cross-memory environment for the subsystem address space is terminated. A non-zero return code is provided to the calling process.
- User Response:* Attempt to obtain a slip dump of this situation, and contact BMC Software Product Support.
-
- BMC11400I** **REGISET service task has started. *bcssid***
- Explanation:* The BCSS is initializing, and the REGISET service task has been attached.
- System Action:* Initialization of the BCSS continues.
- User Response:* This message is informational only.
-
- BMC11408I** **REGISET service task is stopping. *bcssid***
- Explanation:* The BCSS is terminating, and the REGISET service task has been instructed to terminate.
- System Action:* Termination of the BCSS continues.
- User Response:* This message is informational only.
-
- BMC11502E** **CASB storage not available, R15(*returncode*). *bcssid***
- Explanation:* While processing a request for creation of a new CAS, a subsystem service task was unable to obtain storage in the extended common service area (CSA) for a control block to represent the CAS. *returncode* indicates the generated return code.
- System Action:* The request to create the CAS is flagged as incomplete.
- User Response:* Resolve the CSA storage shortage. If the problem persists, contact BMC Software Product Support.
-
- BMC11504E** **CAS create request failed, R15(*returncode*). *bcssid***
- Explanation:* The start command to MVS failed. *returncode* indicates the generated return code.
- System Action:* The CAS terminates.
- User Response:* If you cannot resolve the problem, contact BMC Software Product Support.

BMC11506E CAS initialization failed, code(*returncode*). *bcssid*

Explanation: The newly-created CAS sent a return code to the service task indicating that initialization of the CAS was not successful. The CAS should have issued one or more messages about its inability to initialize. *returncode* indicates the generated return code.

System Action: The request to create the CAS is flagged as incomplete. The requesting product issues a message about the incomplete operation.

User Response: If you cannot resolve the problem, contact BMC Software Product Support.

BMC11509E Abend *abendcode* occurred during CAS create processing. *bcssid*

Explanation: The CAS create processor abended during CAS created processing. *abendcode* indicates the generated abend code.

System Action: The CAS terminates.

User Response: Contact BMC Software Product Support with the PSW and registers.

BMC11512E SMGETMAIN failed, R15(*returncode*), ASID(*asid*), TCB(*tcb*), size(*size*), SP(*subpool*), KEY(*key*), jobname(*jobname*). *bcssid*

Explanation: Control storage was unavailable in an address space where your product attempted to process a VSAM data set that an application attempted to open. It is possible that a limitation was placed on the amount of storage above the 16 MB line that can be allocated within an address space. *returncode* indicates the generated return code. *asid* indicates the address space of the requesting task. *tcb* indicates the task control block (TCB) of the requesting task. *size* indicates the amount of storage requested by the task. *subpool* indicates the storage pool ID. *key* indicates the hardware protect key of the storage used. *jobname* indicates the name of the job requesting the task.

System Action: Failsoft processing is enabled, and records placed in the VSAM data set are not compressed. Failsoft processing permits the expansion of compressed records at all times. The BMC Software message BMC108402I is issued in the address space that attempted to open the VSAM data set.

User Response: Resolve the storage availability problem and then close and reopen the VSAM data set. This action prompts your product to process the VSAM data set normally. If attempts to resolve the storage availability problem are unsuccessful, contact BMC Software Product Support.

BMC11520I VSAM ACB close terminated abnormally, reason(*reasoncode*). *bcssid*

Explanation: A service task, processing an ACB CLOSE request on behalf of a routine executing in another address space, determined that the CLOSE operation could not be completed successfully. *reasoncode* indicates the generated reason code>

System Action: The process requesting the ACB CLOSE is informed of the condition and usually responds by issuing an informational message.

User Response: This message is informational only.

BMC11544I <Data set | cluster> (*name*). *bcssid*

Explanation: This message provides the data set or cluster name of the affected entity. Other BMC Software messages are issued in conjunction with this message. *name* indicates the name of the data set or cluster.

System Action: Processing continues.

User Response: This message is informational only.

BMC11545E REGISET access error, RPLREQ(*requesttype*), RPLFDBK(*feedback*), RPLOPTCD(*option*), STB(*address*). *bcssid*

Explanation: This message is produced by the service request task whenever messages BMC11160E or BMC11162E have been issued. *requesttype* indicates the VSAM request under execution. *feedback* indicates feedback provided by VSAM that identifies the status of the request. *option* indicates the type of VSAM processing.

System Action: The BCSS service request processing is not completed

Explanation: This message may be an early indication of a REGISET problem—such as running out of space—that needs prompt attention. Contact BMC Software Product Support for assistance in identifying and resolving the condition.

BMC11546I RPLARG(*value*). *bcssid*

Explanation: This message is produced whenever an error is encountered during registration data set processing. *value* indicates the EBCDIC key being accessed.

System Action: Processing continues.

User Response: This message is informational only.

BMC11547I ***hexz(zone). bcssid***
hexn(numeric). bcssid

Explanation: These messages are produced whenever an error is encountered during registration data set processing and is used to dump the key of the record being accessed. *zone* indicates the zone values. *numeric* indicates the numeric values.

System Action: Processing continues.

User Response: This message is informational only.

BMC11549E **Abend *abendcode* occurred during request processing. *bcssid***

Explanation: An error was encountered while processing a service request for a user address space. Processing a service request is a standard function of the BCSS, and an abend indicates that a potentially serious error has occurred. *abendcode* indicates the generated abend code.

System Action: The indication of the abend is provided to the requestor. The requesting process usually issues an informational message describing the failed request.

User Response: Contact BMC Software Product Support.

BMC11700I **Status, subsystem address space *asid* is active BCSS. *bcssid***
Status, task (*ii*), NDISP (*disp*), *taskstatus* BCSS. *bcssid*

Explanation: This message is issued in response to the *bcssid* STATUS command. If active, the BCSS returns the general purpose status message. If multiple subsystem service tasks are active in the BCSS address space, each service task has a message. *asid* indicates the address space in which the task is running. *ii* indicates the unique identifier for the service task. *disp* indicates the disposition. *taskstatus* indicates the status of the task, which can be one of the following:

ACTIVE—The task is being processed.

INACTIVE—The task is not being processed.

SUSPENDED—The task is waiting for a request.

HALTING—The task is beginning normal termination.

STOPPING—The task is proceeding with normal termination.

ABENDING—The task is recovering from a failure. When recovery is complete, the task returns to a SUSPENDED or ACTIVE status.

System Action: Processing continues.

User Response: This message is informational only.

BMC11790E Abend *abendcode* occurred during command processing. *bcssid*

Explanation: An operator-initiated command to BCSS abnormally terminated. It is possible that the command is detecting storage-overlay and/or control-block errors. *abendcode* indicates the generated abend code.

System Action: Execution of the operator-initiated command terminates.

User Response: Issue the command again. If the problem persists, contact BMC Software Product Support with the *abendcode* value.

BMC11800S *servicename* service request failed, R15(*returncode*). *bcssid*

Explanation: An MVS cross-memory service was requested during CAS initialization and was not successful. *servicename* indicates the service that failed: AXRES, ATSET, ETCRE, ETCON, LXDES, LXDIS, and LXRES. *returncode* indicates the generated return code.

System Action: CAS initialization terminates. The cross-memory service must execute successfully to complete the initialization process.

User Response: If you do not have sufficient explanation of the problem based on the return code to resolve the problem, contact BMC Software Product Support.

BMC11840I CAS initialization completed for *jobname(jobname)* in ASID(*asid*). *bcssid*

Explanation: A CAS has completed initialization for a job and is ready to handle compression and/or expansion requests. *jobname* indicates the name of the job requesting CAS services. *asid* indicates the address space in which the job is executing.

System Action: CAS continues processing.

User Response: This message is informational only.

BMC11846E CAS terminating with reason code(*reasoncode*). *bcssid*

Explanation: The CAS is terminating with a non-zero reason code. This message is produced when an operator manually starts the CAS without the BCSS requiring it. *reasoncode* indicates the generated reason code.

System Action: The CAS terminates.

User Response: Record the abend code for problem tracking.

BMC11848I CAS termination completed for asid(*asid*). *bcssid*

Explanation: Termination of a CAS has completed. *asid* indicates the address space of the CAS.

System Action: CAS processing has terminated.

User Response: This message is informational only.

BMC11850S Abend *abendcode* occurred during CAS processing. *bcssid*

Explanation: An unrecoverable error occurred during CAS processing. Normally, only IBM S80A and S878 abend codes should be expected; any other abend represents a potentially serious problem. *abendcode* indicates the generated abend code.

System Action: The CAS terminates.

User Response: Try to resolve the abend by increasing the REGION size on the BMCCAS procedure and restarting the CAS. If this is unsuccessful, contact BMC Software Product Support.

BMC11860I LAS initialization completed for jobname(*jobname*) in ASID(*asid*). *bcssid*

Explanation: A consolidated environment has been initialized for a job. Compression and/or expansion requests will be processed by this *local* environment. *jobname* indicates the name of the job. *asid* indicates the address space in which the job is executing.

System Action: The job continues processing.

User Response: This message is informational only.

BMC11867S Abend *abendcode* occurred during LAS initialization. *bcssid*

Explanation: An error has occurred during initialization of a LAS compression environment. *abendcode* indicates the generated abend code.

System Action: The application may be abended.

User Response: If the error resulted from being unable to load modules from the linklist, correct the linklist problem. Otherwise, use CAS by either turning off local compression (*bcssid xxx CL CAS*) or by altering the program registration entries to specify *CL(CAS)*. Contact BMC Software Product Support with appropriate LOGREC and SYSLOG.

BMC11869E**Abend *abendcode* occurred during LAS processing. *bcssid***

Explanation: An error has occurred during LAS processing. *abendcode* indicates the generated abend code.

System Action: LAS processing terminates.

User Response: Contact BMC Software Product Support to report the PSW and registers.

BMC11870S**CASB not available to consolidated memory. *bcssid***

Explanation: The control block representing a valid CAS is not available. This message is issued whenever a CAS is started by an operator and the BCSS does not need additional compression address spaces active.

System Action: The CAS terminates.

User Response: If the START command for the CAS was issued by BCSS and not the operator, contact BMC Software Product Support.

BMC11872S**CASB(*address*) failed validation. *bcssid***

Explanation: The control block representing a valid CAS is not valid. This condition normally indicates a CSA overlay. *address* indicates the address of the control block.

System Action: The CAS terminates.

User Response: Examine both the BMCCAS and BCSS procedures to ensure that no STEPLIBs are being used. Contact the system programmer maintaining the BCSS modules and ensure that multiple levels of the modules are not being executed inadvertently. Cycle the BCSS. If the problem persists, contact BMC Software Product Support.

BMC11890S***servicename* service request failed R15(*returncode*). *bcssid***

Explanation: An MVS cross-memory service was requested during CAS termination and was not successful. *servicename* indicates the service that failed: AXRES, ATSET, ETCRE, ETCO, LXDES, LXDIS, and LXRES. *returncode* indicates the generated return code.

System Action: The CAS abends.

User Response: If the IBM publication describing the service *returncode* does not provide a sufficient explanation of the problem and a resolution, contact BMC Software Product Support.

BMC11901E Module (*modulename*) not found, execution terminated.

Explanation: The user exit for OPEN can not be located in a STEPLIB or linklist library. *modulename* indicates the module that was not located.

System Action: The Extract program terminates with a return code of 20.

User Response: The OPEN exit, BCSUOPNE, distributed with the product, needs to be placed in a load library accessible to this program execution.

BMC11902E Consolidated subsystem unavailable, module (*modulename*) not found, execution terminated.

Explanation: One of the BCSS modules was not found in the link pack area. *modulename* indicates the module that was not located.

System Action: The Extract program terminates with a return code of 20.

User Response: The BCSS has not been started on the system on which the Extract program is being executed. Either start the BCSS on the affected system, or run the Extract program on a system where the BCSS is being executed.

BMC11903E BMC consolidated subsystem is not initialized, execution terminated.

Explanation: The Extract program is being executed on a system on which the BCSS is not being executed.

System Action: The Extract program terminates with a return code of 20.

User Response: Either start the BCSS on the affected system, or run the Extract program on a system where the BCSS is being executed.

BMC11904E The consolidated subsystem (*subsysid*) is unavailable (SSCT not found), execution terminated.

Explanation: The Extract program is being executed on a system on which the BCSS is not being executed. *subsysid* indicates the BCSS subsystem name.

System Action: The Extract program terminates with a return code of 20.

User Response: Either start the BCSS on the affected system, or run the Extract program on a system where the BCSS is being executed.

BMC11905E The DD statement (*ddname*) was not supplied, execution terminated.

Explanation: One of the two required DD statements was not supplied, or one of them was incorrectly specified on an OUTPUT control statement. *ddname* indicates the DD statement.

System Action: The Extract program terminates with a return code of 20.

User Response: Either add the missing DD statement, or correct the invalid *ddname* specified on the OUTPUT control statement.

BMC11906E The consolidated subsystem (*subsysid*) is unavailable (SSVT not found), execution terminated.

Explanation: The Extract program is being executed on a system on which the BCSS is not being executed. *subsysid* indicates the BCSS subsystem name.

System Action: The Extract program terminates with a return code of 20.

User Response: Either start the BCSS on the affected system or run the Extract program on a system where the BCSS is being executed.

BMC11907W The PARM field supplied is invalid and has been ignored, execution continues.

Explanation: The only acceptable value in the PARM field is a replacement *ddname* for the SYSIN data set. An invalid *ddname* was entered or the value supplied exceeds 8 characters.

System Action: The Extract program continues execution if a SYSIN DD statement has been supplied.

User Response: Either correct the invalid DD name or remove the PARM field from the EXEC statement.

BMC11908E OPEN failed for (*ddname*) DDNAME, execution terminated.

Explanation: An attempt to OPEN a data set failed. An IBM error message precedes this error message. *ddname* indicates the DD statement associated with the data set.

System Action: The Extract program terminates with a return code of 20.

User Response: Please refer to the appropriate IBM System Messages Manual to determine the cause of the failure.

- BMC11909E** **Dynamic allocation of (*ddname*) DDNAME failed, RC(*returncode*) RSNCD(*reasoncode*), execution terminated.**
- Explanation:* An attempt to dynamically allocate a data set failed. *ddname* indicates the DD statement associated with the data set. *returncode* indicates the generated return code. *reasoncode* indicates the generated reason code.
- System Action:* The Extract program terminates with a return code of 20.
- User Response:* Contact BMC Software Product Support.
-
- BMC11910W** **CARD # *cardnumber*, COLUMN # *columnnumber*. This parameter is not recognized, remainder of record ignored.**
- Explanation:* A parameter is invalid and is assumed to be a comment. *cardnumber* and *columnnumber* indicate input record card and column where the error is located.
- System Action:* The Extract program ignores the remainder of the input record.
- User Response:* If the parameter is *not* a comment, correct the spelling, and resubmit the job for execution.
-
- BMC11911W** **CARD # *cardnumber*, COLUMN # *columnnumber*. The new DDNAME is missing. Remainder of record ignored.**
- Explanation:* An attempt was made to change a *ddname*. The new *ddname* was not supplied. *cardnumber* and *columnnumber* indicate record input card and column where the error is located.
- System Action:* The Extract program ignores the remainder of the input record.
- User Response:* Either supply a new *ddname*, or remove the OUTPUT control statement from the input stream.
-
- BMC11912W** **CARD # *cardnumber*, COLUMN # *columnnumber*. This operand is too long. Remainder of record ignored.**
- Explanation:* A supplied parameter is greater than 8 characters in length. *cardnumber* and *columnnumber* indicate input record card and column where the error is located.
- System Action:* The Extract program ignores the remainder of the input record.
- User Response:* Either correct the parameter, or remove the OUTPUT control statement from the input control card stream.

BMC11913E The DD statement (*ddname*) was not supplied, execution terminated.

Explanation: The *ddname* specified in the preceding error message was not allocated. This is a required DD statement. *ddname* indicates the missing DD statement.

System Action: The Extract program terminates with a return code of 20.

User Response: Add the required DD statement to the job stream, and submit the job for execution.

BMC11914E OPEN failed for (*ddname*) DCB, execution terminated.

Explanation: An attempt to OPEN the named DCB failed. An operating system message precedes this message explaining the error. *ddname* indicates the DD statement associated with the DCB.

System Action: The Extract program terminates with a return code of 20.

User Response: Specification of invalid DCB parameters on the DD statement could result in this error. If the problem cannot be resolved, contact BMC Software Product Support.

BMC11915E BMC consolidated subsystem services unavailable, execution terminated.

Explanation: An attempt to retrieve a registration control record using BCSS service routines failed, indicating that the subsystem services are not available.

System Action: The Extract program terminates with a return code of 20.

User Response: Contact BMC Software Product Support.

BMC11916W Input command not recognized. This input statement will be ignored.

Explanation: An input record contained an unsupported command. The input record is ignored.

System Action: The Extract program continues processing.

User Response: Please refer to your product's documentation for an explanation of the valid input commands for the Extract program. Contact BMC Software Product Support for assistance if the problem can not be resolved.

BMC11917I The DDNAME for the (*ddname*) DCB has been altered to (*ddname*).

Explanation: As requested with an OUTPUT command, the *ddname* for the indicated DCB has been altered prior to OPEN. *ddname* indicates the DD statement associated with the DCB.

System Action: The Extract program continues processing.

User Response: This message is informational only.

BMC11918W No records were retrieved from the registration data set. Review your control cards.

Explanation: This execution of the Extract program resulted in no registration data set records being extracted.

System Action: The Extract program terminates with a return code of 20.

User Response: If INCLUDE or EXCLUDE commands were used, refer to your products documentation for an explanation of the keywords for these commands. Contact BMC Software Product Support for assistance if the problem cannot be resolved.

BMC11919E A VSAM I/O request delivered to the consolidated subsystem ended with a 'SPL' code of code(*returncode*).

Explanation: An error was found with the request delivered to the BCSS. *returncode* indicates the generated return code in decimal and hexadecimal format.

System Action: The Extract program terminates with a return code of 20.

User Response: Contact BMC Software Product Support.

BMC11920E A VSAM GET ended with a return code of *returncode* and a feedback code of *feedback*.

Explanation: A VSAM error was encountered by the BCSS processing a request for the Extract program. *returncode* indicates the generated return code in decimal and hexadecimal format. *feedback* indicates the generated feedback code in decimal and hexadecimal format.

System Action: The Extract program terminates with a return code of 20.

User Response: Contact BMC Software Product Support.

- BMC11921E** **The requested subsystem (*subsysid*) is not a compression subsystem, execution terminated.**
- Explanation:* The subsystem requested is not a compression subsystem. *subsysid* indicates the invalid subsystem.
- System Action:* The Extract program terminates.
- User Response:* Correct the subsystem designation, and run the program again.
-
- BMC11922W** **EXCLUDE control statement ignored. Cannot be mixed with INCLUDE statements.**
- Explanation:* EXCLUDE control statements are not processed with INCLUDE control statements.
- System Action:* The Extract program continues processing, but ignores the invalid EXCLUDE and INCLUDE statements.
- User Response:* Place EXCLUDE and INCLUDE control statements on separate lines before attempting another execution.
-
- BMC11923W** **INCLUDE control statement ignored. Cannot be mixed with EXCLUDE statements.**
- Explanation:* EXCLUDE control statements are not processed with INCLUDE control statements.
- System Action:* The Extract program continues processing, but ignores the invalid EXCLUDE and INCLUDE statements.
- User Response:* Place EXCLUDE and INCLUDE control statements on separate lines before attempting another execution.
-
- BMC11924W** **This record was ignored. 'OUTPUT' records must precede all other input records.**
- Explanation:* OUTPUT control statements must precede all other input control statements.
- System Action:* The Extract program continues processing.
- User Response:* Place all OUTPUT control statements first in the input stream.

BMC11925E **ABEND *abendcode* occurred during execution. Program is terminating.**

Explanation: An abend has occurred during the execution of BCSUEXTR. *abendcode* indicates the generated abend code.

System Action: The job step terminates with a return code of 20. If a SYSMDUMP, SYSUDUMP or SYSABEND DD statement was included in the job step, a dump is produced. A dump is required before the error can be corrected.

User Response: Contact BMC Software Product Support.

BMC11926W **The BCSSID parameter is invalid or incomplete, the parameter is ignored.**

Explanation: The BCSSID keyword is either invalid or incomplete.

System Action: Processing continues.

User Response: Correct the BCSSID designation, and submit the request again.

Index

A

address space
 compression 1-2
 in XCA 1-2
 local 1-2

B

Batch Registration utility 5-13
BCSS
 hardware reserve for REGISSET 4-4
 REGISSET usage 4-2
BCSS CAS DRAIN command 2-16
BCSS CAS RESTART command 2-17
BCSS CAS START command 2-14
BCSS CAS STATUS command 2-18
BCSS CAS STOP command 2-15
BCSS ELM command 2-11
BCSS REINIT DAC command 2-10
BCSS SHUTDOWN command 2-8
BCSS START command 2-7
BCSS STATUS command 2-9
BCSUEXTR utility 5-2
BCSURDS0 utility 5-13
BMC Consolidated Subsystem (BCSS)
 architecture 1-2
 commands data set 1-5
 displaying status 2-9
 ELM command 2-11
 Engine Level Mask (ELM) overview 1-8,
 2-12

BMC Consolidated Subsystem (BCSS) *cont*
 overview 1-2, 1-4
 registration data set (REGISSET) failure
 detection 4-5
 REINIT DAC command 2-10
 reinitializing DATA ACCELERATOR 2-10
 setting the ELM 2-11
 SHUTDOWN command 2-8
 START command 2-7
 starting 2-7
 STATUS command 2-9
 stopping 2-8
 subsystem ID 1-6
BMC Primary Subsystem (BMCP)
 architecture 1-2
 commands 2-2
 displaying status 2-5
 overview 1-3
 SHUTDOWN command 2-4
 START command 2-3
 starting 2-3
 STATUS command 2-5
 stopping 2-4
 subsystem ID 1-6
BMC Software, contacting ii
BMCP SHUTDOWN command 2-4
BMCP STATUS command 2-5

C

cache feature of REGISSET 4-5
command summary 1-5

commands

- BCSS CAS DRAIN 2-16
- BCSS CAS RESTART 2-17
- BCSS CAS START 2-14
- BCSS CAS STATUS 2-18
- BCSS CAS STOP 2-15
- BCSS ELM 2-11
- BCSS REINIT DAC 2-10
- BCSS SHUTDOWN 2-8
- BCSS START 2-7
- BCSS STATUS 2-9
- BMCP SHUTDOWN 2-4
- BMCP STATUS 2-5
- START BMCP 2-3

commands data set 1-5

compression address space (CAS) 1-2

- displaying status 2-18
- DRAIN command 2-16
- draining 2-16
- RESTART command 2-17
- restarting 2-17
- START command 2-14
- starting 2-14
- STATUS command 2-18
- STOP command 2-15
- stopping 2-15

conventions

- document xiv
- syntax statements xv
- typographical xiv

customer support ii

D

document conventions xiv

documentation

- electronic
 - online help xiii
 - online xiii

duplex registration data set

- overview 4-3
- usage 4-4

E

electronic documentation

- online help xiii

Engine Level Mask (ELM)

- overview 1-8, 2-12
- setting for BMC Consolidated Subsystem (BCSS) 2-11

Extended Compression Architecture (XCA) 1-2

Extract utility (BCSUEXTR)

- record layout data set section 5-20
- record layout header 5-19
- record layout overview 5-18
- record layout program section 5-24
- record layout statistical section 5-27

H

hardware reserve 4-4

I

Installation Check utility 5-9

interface, EDM ISPF

- online help xiii

IPL 2-19

M

Multiple Primary feature 4-4

O

online documentation xiii

- online help xiii

orderly shutdown 2-19

P

primary registration data set

- overview 4-3
- usage 4-4

product support ii

R

REGISET

- cache feature 4-5
- duplex registration data set 4-3, 4-4
- hardware reserve 4-4
- Multiple Primary feature 4-4
- number of data sets 4-3
- overview 4-2
- primary registration data set 4-3, 4-4
- registration data sets 4-3
- SHAREOPTIONS 4-4
- registration data set (REGISET)
 - failure detection 4-5
 - off-site backup 4-12
 - recovery 4-6
- related documentation xiii
- release notes xiii
- reserve, hardware 4-4
- resource management 1-2

S

- shutdown before IPL 2-19
- START BMCP command 2-3
- subsystem architecture 1-2
- support, product ii
- syntax statement conventions xv

T

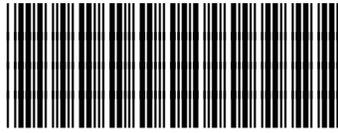
- technical support ii
- typographical conventions xiv

U

utilities

- Batch Registration 5-13
- Extract (BCSUEXTR) 5-2
- Installation Check (BCSCHECK) 5-9
- REGISET Cleanup (BCSURDS0) 5-13

Notes



100042477