APPLICATION RESTART CONTROL
Reference Manual: BATCH CONTROL FACILITY

Supporting

Version 4.1 of Application Restart Control *for IMS*

December 2015
Contacting BMC Software

Several methods are available for contacting BMC Software.

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

United States and Canada

<table>
<thead>
<tr>
<th>Address</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC SOFTWARE INC</td>
<td>1 713 918 8800</td>
<td>1 713 918 8000</td>
</tr>
<tr>
<td>2103 CITYWEST BLVD</td>
<td>or</td>
<td></td>
</tr>
<tr>
<td>HOUSTON TX 77042-2827 USA</td>
<td>1 800 841 2031</td>
<td></td>
</tr>
</tbody>
</table>

Outside United States and Canada

<table>
<thead>
<tr>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>+01 713 918 8800</td>
<td>+01 713 918 8000</td>
</tr>
</tbody>
</table>

© Copyright 1994-2016 BMC Software, Inc.

BMC, BMC Software, and the BMC Software logo are the exclusive properties of BMC Software, Inc., are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other BMC trademarks, service marks, and logos may be registered or pending registration in the U.S. or in other countries. All other trademarks or registered trademarks are the property of their respective owners.

DB2, IBM, IBM z, IMS, MVS, OS/390, and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

The information included in this documentation is the proprietary and confidential information of BMC Software, Inc., its affiliates, or licensors. Your use of this information is subject to the terms and conditions of the applicable End User License agreement for the product and to the proprietary and restricted rights notices included in the product documentation.

Restricted rights legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC SOFTWARE INC, 2101 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.
Customer support

Support website
You can obtain technical support from BMC 24 hours a day, 7 days a week at http://www.bmc.com/support. From this website, you can:

■ Read overviews about support services and programs that BMC offers
■ Find the most current information about BMC products
■ Search a database for problems similar to yours and possible solutions
■ Order or download product documentation
■ Download products and maintenance
■ Report a problem or ask a question
■ Subscribe to receive proactive e-mail alerts
■ Find worldwide BMC support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by telephone or e-mail
In the United States and Canada, if you need technical support and do not have access to the web, call 1 800 537 1813 or send an e-mail message to customer_support@bmc.com. (In the subject line, enter SupID:yourSupportContractID, such as SupID:12345). Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC
Have the following information available so that Customer Support can begin working on your issue immediately:

■ Product information
  — Product name
  — Product version (release number)
  — License number and password (trial or permanent)
■ Operating system and environment information
  — Machine type
  — Operating system type, version, and service pack or other maintenance level such as PUT or PTF
  — System hardware configuration
  — Serial numbers
  — Related software (database, application, and communication) including type, version, and service pack or maintenance level
■ Sequence of events leading to the problem
■ Commands and options that you used
■ Messages received (and the time and date that you received them)
  — Product error messages
  — Messages from the operating system
  — Messages from related software
License key and password information

If you have questions about your license key or password, contact Customer Support through one of the following methods:

- Send an e-mail message to customer_support@bmc.com. (In the Subject line, enter SupID:yourSupportContractID, such as SupID:12345.)

- In the United States and Canada, call 1 800 537 1813. Outside the United States and Canada, contact your local support center for assistance.

## Contents

**About this book** 11
- Related publications ................................................................. 11
- Conventions ............................................................................. 12
- Syntax statements ................................................................. 12

**Chapter 1 Introduction** 15
- **BCF environment** ................................................................. 15
  - BATCH CONTROL FACILITY component ................................ 15
  - APPLICATION RESTART CONTROL services and facilities .......... 15
  - APPLICATION RESTART CONTROL ........................................ 16
- **BCF components** ................................................................. 16
  - BMC Software subsystems .......................................................... 17
  - Registration data sets and REGISET .......................................... 18
  - AR/CTL application supervisor .................................................... 19
  - BCF execution controller ............................................................ 19
  - BCF Physical Logger .................................................................. 21
  - Enhanced processing ................................................................... 23
  - Cancel intercept processing ....................................................... 24
  - Abend filter processing ............................................................... 25
  - History data set and statistical reports ........................................ 25
  - BCF ISPF interface .................................................................... 26
  - Batch utilities ............................................................................ 26
- **Other features of BCF** .......................................................... 27
  - Conditional backout step bypass ............................................... 27
  - Dummy logging environment support ........................................ 28
  - Batch Backout Assist feature ..................................................... 28
  - SVC99 retry processing ............................................................. 29
  - GDG support ............................................................................ 30
  - Fast OSAM Buffer Purge ............................................................ 30
- **Summary of benefits** .......................................................... 31
  - Data integrity ............................................................................ 31
  - Data availability ......................................................................... 32
  - Resource conservation ............................................................. 33
- **Considerations for using BCF** ............................................... 34
  - General considerations ............................................................ 34
  - IMS support ............................................................................. 34
<table>
<thead>
<tr>
<th>Chapter 2</th>
<th>Getting started</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of BCF component tasks</td>
<td>37</td>
</tr>
<tr>
<td>Getting started with BCF installation</td>
<td>38</td>
</tr>
<tr>
<td>Working with BMC Software subsystems</td>
<td>39</td>
</tr>
<tr>
<td>BMCP</td>
<td>39</td>
</tr>
<tr>
<td>BCSS</td>
<td>39</td>
</tr>
<tr>
<td>Working with the REGISET</td>
<td>40</td>
</tr>
<tr>
<td>The REGISET</td>
<td>40</td>
</tr>
<tr>
<td>Types of records</td>
<td>41</td>
</tr>
<tr>
<td>Record key qualifiers</td>
<td>43</td>
</tr>
<tr>
<td>Wildcard characters in record keys</td>
<td>44</td>
</tr>
<tr>
<td>Getting started with the BCF ISPF interface</td>
<td>45</td>
</tr>
<tr>
<td>Setup requirements for your ISPF session</td>
<td>45</td>
</tr>
<tr>
<td>Using the Application Enhancement Series primary menu</td>
<td>46</td>
</tr>
<tr>
<td>Using the BATCH CONTROL FACILITY primary menu</td>
<td>47</td>
</tr>
<tr>
<td>Using the Active BMC Consolidated Subsystems panel</td>
<td>49</td>
</tr>
<tr>
<td>Limiting the list of records</td>
<td>50</td>
</tr>
<tr>
<td>Listing records</td>
<td>50</td>
</tr>
<tr>
<td>Adding a record</td>
<td>51</td>
</tr>
<tr>
<td>Deleting a record</td>
<td>52</td>
</tr>
<tr>
<td>Implementing BCF in application programs</td>
<td>53</td>
</tr>
<tr>
<td>Working with IMS environment registration records</td>
<td>53</td>
</tr>
<tr>
<td>Working with IMS program registration records</td>
<td>55</td>
</tr>
<tr>
<td>Modification of JCL</td>
<td>56</td>
</tr>
<tr>
<td>Excluding BCF from program execution</td>
<td>59</td>
</tr>
<tr>
<td>Working with IMS program exclusion records</td>
<td>59</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 3</th>
<th>Using BCF options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of the options</td>
<td>61</td>
</tr>
<tr>
<td>Option levels</td>
<td>61</td>
</tr>
<tr>
<td>How BCF merges option values</td>
<td>62</td>
</tr>
<tr>
<td>Selecting the appropriate option level</td>
<td>64</td>
</tr>
<tr>
<td>Methods for setting options</td>
<td>65</td>
</tr>
<tr>
<td>Option descriptions</td>
<td>65</td>
</tr>
<tr>
<td>Summary of the options</td>
<td>66</td>
</tr>
<tr>
<td>Working with processing option records in ISPF</td>
<td>68</td>
</tr>
<tr>
<td>Accessing global processing option records</td>
<td>69</td>
</tr>
<tr>
<td>Accessing system processing option records</td>
<td>69</td>
</tr>
<tr>
<td>Accessing job step processing option records</td>
<td>70</td>
</tr>
</tbody>
</table>
Processing Options panel ..................................................................................................................70

Options on page 1 of the Processing Options panel ............................................................................72
  BATCH CONTROL FACILITY Active ...............................................................................................73
  Use Execution Controller ..................................................................................................................74
  Use Physical Logger ............................................................................................................................75
  DASD Unit Name ...................................................................................................................................76
  Standard Tape Unit Name .....................................................................................................................77
  Buffered Tape Unit Name ......................................................................................................................78
  JOBLOG Messages ...............................................................................................................................79
  History Data Set Name ..........................................................................................................................80

Options on page 2 of the Processing Options panel ..................................................................................81
  Physical Logger Error Option ...............................................................................................................81
  Delete Volume Records .........................................................................................................................82
  Qualify with Program ..............................................................................................................................83
  Automatic Batch Backout ......................................................................................................................84
  Buffered Device Support .......................................................................................................................86
  Discard Logging Active .........................................................................................................................88
  Notify Option for DBRC ........................................................................................................................90
  Exchange Log Device Type ....................................................................................................................91
  Dynamic Log Option ..............................................................................................................................93
  Dynamic Log Type .................................................................................................................................95

Options on page 3 of the Processing Options panel ..................................................................................95
  SYSOUT Class .........................................................................................................................................96
  Tape BKO ..............................................................................................................................................96
  Max SLDS Block Count ...........................................................................................................................97
  DL/I Activity Statistics ...........................................................................................................................98
  Record Activity Stats To ........................................................................................................................100
  Number of Statistics Sets .......................................................................................................................100
  Logger Perf % Threshold .......................................................................................................................101
  Alert User ID ......................................................................................................................................102

Options on page 4 of the Processing Options panel .................................................................................104
  Backout Variable (BKO) .......................................................................................................................104
  Department Variable (DEPT) .................................................................................................................105
  Use Log Compaction ............................................................................................................................105
  Manual Intervention Override ...............................................................................................................106
  Ignore Checkpoints ...............................................................................................................................107
  Use Abend Code Series ........................................................................................................................108

Working with options at the execution (BCFSYSIN) level .......................................................................109
  BCFSYSIN control statement syntax .....................................................................................................109
  BCFSYSIN syntax errors .......................................................................................................................110
  BCFSYSIN keyword summary .............................................................................................................110
## Chapter 4  Preparation for dynamic allocation of log data sets

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding BCF dynamic allocation</td>
<td>113</td>
</tr>
<tr>
<td>Dynamic allocation uses</td>
<td>113</td>
</tr>
<tr>
<td>Conditions for dynamic allocation</td>
<td>114</td>
</tr>
<tr>
<td>Preparation for dynamic allocation</td>
<td>115</td>
</tr>
<tr>
<td>Device classes for dynamic allocation records</td>
<td>115</td>
</tr>
<tr>
<td>Conditions for new log allocation</td>
<td>118</td>
</tr>
<tr>
<td>Dynamic allocation process</td>
<td>120</td>
</tr>
<tr>
<td>Functions and options for dynamic allocation support</td>
<td>121</td>
</tr>
<tr>
<td>Accessing dynamic allocation records with the ISPF interface</td>
<td>123</td>
</tr>
<tr>
<td>Accessing global dynamic allocation records</td>
<td>123</td>
</tr>
<tr>
<td>Accessing system dynamic allocation records</td>
<td>124</td>
</tr>
<tr>
<td>Accessing job step dynamic allocation records</td>
<td>125</td>
</tr>
<tr>
<td>Dynamic allocation options panels</td>
<td>126</td>
</tr>
<tr>
<td>Setting dynamic allocation options on page 1</td>
<td>126</td>
</tr>
<tr>
<td>Setting dynamic allocation options on page 2</td>
<td>129</td>
</tr>
<tr>
<td>Data set name skeletons</td>
<td>132</td>
</tr>
<tr>
<td>Restrictions on generation data groups</td>
<td>133</td>
</tr>
<tr>
<td>Syntax and substitution keywords</td>
<td>133</td>
</tr>
<tr>
<td>Example skeletons</td>
<td>135</td>
</tr>
<tr>
<td>Dynamic recovery from allocation errors</td>
<td>136</td>
</tr>
<tr>
<td>Generation data groups</td>
<td>137</td>
</tr>
<tr>
<td>GDG log data set support</td>
<td>137</td>
</tr>
<tr>
<td>GDG environments</td>
<td>138</td>
</tr>
<tr>
<td>Allocation skeleton setup</td>
<td>139</td>
</tr>
<tr>
<td>Use of &amp;COPY for the data set name</td>
<td>139</td>
</tr>
<tr>
<td>Use of GDGs with the same skeleton and current log base</td>
<td>140</td>
</tr>
<tr>
<td>Use of GDGs with different skeleton and current log bases</td>
<td>142</td>
</tr>
<tr>
<td>Allocation of old logs</td>
<td>143</td>
</tr>
<tr>
<td>Allocation using the Exchange Log Device Type option</td>
<td>143</td>
</tr>
<tr>
<td>Restrictions on the use of GDGs</td>
<td>144</td>
</tr>
<tr>
<td>Examples of GDG processing</td>
<td>145</td>
</tr>
<tr>
<td>Planning considerations</td>
<td>149</td>
</tr>
</tbody>
</table>

## Chapter 5  Using BCF execution control records

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of types of records that BCF uses</td>
<td>151</td>
</tr>
<tr>
<td>Execution control records</td>
<td>151</td>
</tr>
<tr>
<td>Log volume records</td>
<td>152</td>
</tr>
<tr>
<td>Accessing execution control records</td>
<td>152</td>
</tr>
<tr>
<td>Displaying an execution control record</td>
<td>154</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Modifying an execution control record</td>
<td>159</td>
</tr>
<tr>
<td>Cataloging of log volumes after system failure</td>
<td>161</td>
</tr>
<tr>
<td>Use of the Active Job utility</td>
<td>161</td>
</tr>
<tr>
<td>JCL requirements for the Active Job utility</td>
<td>162</td>
</tr>
<tr>
<td>BCFPRINT reports</td>
<td>165</td>
</tr>
<tr>
<td>Utility return and abend codes</td>
<td>168</td>
</tr>
<tr>
<td>Handling nonstandard situations</td>
<td>169</td>
</tr>
<tr>
<td>Special restart processing situations</td>
<td>169</td>
</tr>
<tr>
<td>Manual intervention</td>
<td>174</td>
</tr>
<tr>
<td>REGISET problems</td>
<td>178</td>
</tr>
<tr>
<td>System problems</td>
<td>179</td>
</tr>
<tr>
<td>BCF Automatic Restart Preparation utility</td>
<td>180</td>
</tr>
<tr>
<td>JCL requirements in BCF Automatic Restart Preparation utility</td>
<td>181</td>
</tr>
<tr>
<td>EXEC statement parameters</td>
<td>181</td>
</tr>
<tr>
<td>DD statements</td>
<td>183</td>
</tr>
<tr>
<td>Utility reports</td>
<td>183</td>
</tr>
<tr>
<td>Utility return and abend codes</td>
<td>185</td>
</tr>
<tr>
<td>BCF Purge utility</td>
<td>186</td>
</tr>
</tbody>
</table>

**Chapter 6  Using BCF reports**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of BCF reports</td>
<td>191</td>
</tr>
<tr>
<td>Report types</td>
<td>191</td>
</tr>
<tr>
<td>Overview of saving statistics</td>
<td>192</td>
</tr>
<tr>
<td>Accessing reports with the BCF ISPF interface</td>
<td>194</td>
</tr>
<tr>
<td>Jobstep reports</td>
<td>197</td>
</tr>
<tr>
<td>Jobstep Summary</td>
<td>198</td>
</tr>
<tr>
<td>DL/I Call Summary</td>
<td>199</td>
</tr>
<tr>
<td>Logging Performance Data</td>
<td>200</td>
</tr>
<tr>
<td>Checkwrite Statistics</td>
<td>201</td>
</tr>
<tr>
<td>Logging Delay Detail</td>
<td>202</td>
</tr>
<tr>
<td>Dynamic Log Usage Detail</td>
<td>203</td>
</tr>
<tr>
<td>Application Checkpoint Detail</td>
<td>204</td>
</tr>
<tr>
<td>Time Between Checkpoints</td>
<td>205</td>
</tr>
<tr>
<td>Time Between Checkpoints Distribution</td>
<td>206</td>
</tr>
<tr>
<td>Number of DL/I Calls Between Checkpoints</td>
<td>207</td>
</tr>
<tr>
<td>Number of DL/I Calls Between Checkpoints Distribution</td>
<td>208</td>
</tr>
<tr>
<td>Log data set information</td>
<td>209</td>
</tr>
<tr>
<td>Primary Log</td>
<td>209</td>
</tr>
<tr>
<td>Secondary Log</td>
<td>211</td>
</tr>
<tr>
<td>Log Data Set Record Detail report</td>
<td>211</td>
</tr>
<tr>
<td>Database Activity reports</td>
<td>212</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Jobstep Database Data Set Detail</td>
<td>212</td>
</tr>
<tr>
<td>Jobstep Database Detail</td>
<td>213</td>
</tr>
<tr>
<td>Database Buffer Pool reports</td>
<td>214</td>
</tr>
<tr>
<td>OSAM report</td>
<td>214</td>
</tr>
<tr>
<td>VSAM report</td>
<td>217</td>
</tr>
<tr>
<td>DL/I Activity reports</td>
<td>221</td>
</tr>
<tr>
<td>Data Set IWAIT Summary</td>
<td>221</td>
</tr>
<tr>
<td>Data Set IWAIT Distribution</td>
<td>222</td>
</tr>
<tr>
<td>Data Set IWAIT Detail</td>
<td>223</td>
</tr>
<tr>
<td>Background Write IWAIT Summary</td>
<td>223</td>
</tr>
<tr>
<td>Background Write IWAIT Distribution</td>
<td>224</td>
</tr>
<tr>
<td>DL/I Call reports</td>
<td>225</td>
</tr>
<tr>
<td>DL/I Call IWAIT Summary</td>
<td>226</td>
</tr>
<tr>
<td>DL/I Call Timing Summary</td>
<td>227</td>
</tr>
<tr>
<td>Execution Options Summary report</td>
<td>229</td>
</tr>
<tr>
<td>Chapter 7 Using the log utilities</td>
<td>233</td>
</tr>
<tr>
<td>BCF Log utility</td>
<td>233</td>
</tr>
<tr>
<td>Control statements</td>
<td>234</td>
</tr>
<tr>
<td>Overview of units of work (UOWs)</td>
<td>236</td>
</tr>
<tr>
<td>Control statement formats</td>
<td>237</td>
</tr>
<tr>
<td>Use of the statistics function</td>
<td>239</td>
</tr>
<tr>
<td>Overview of the Logical Compare function</td>
<td>241</td>
</tr>
<tr>
<td>Overview of the Batch Backout Log Create function</td>
<td>242</td>
</tr>
<tr>
<td>Overview of the Online Backout Log Create function</td>
<td>243</td>
</tr>
<tr>
<td>Use of the Find Online BMP Executions function</td>
<td>246</td>
</tr>
<tr>
<td>Utility return codes</td>
<td>247</td>
</tr>
<tr>
<td>Database Changed Block Reconciliation utility</td>
<td>248</td>
</tr>
<tr>
<td>Utility control information and JCL</td>
<td>248</td>
</tr>
<tr>
<td>Basic program function</td>
<td>249</td>
</tr>
<tr>
<td>Utility input</td>
<td>249</td>
</tr>
<tr>
<td>Utility output</td>
<td>249</td>
</tr>
<tr>
<td>Program report</td>
<td>250</td>
</tr>
<tr>
<td>Appendix A Contacting BMC Software product support</td>
<td>253</td>
</tr>
<tr>
<td>How to expedite resolution of your problem</td>
<td>253</td>
</tr>
<tr>
<td>BCFPRINT data set</td>
<td>253</td>
</tr>
<tr>
<td>BCFTRACE DD</td>
<td>254</td>
</tr>
<tr>
<td>BCFBOTRC DD</td>
<td>254</td>
</tr>
<tr>
<td>SYSUDUMP DD</td>
<td>254</td>
</tr>
</tbody>
</table>
About this book

This manual provides information about using the BMC Software BATCH CONTROL FACILITY (BCF) component of the APPLICATION RESTART CONTROL (AR/CTL) product. The manual is written for IBM IMS system programmers, database administrators, and application programmers who want to use the BCF logging and execution control services. It assumes that you are familiar with IMS terms and concepts.

Like most BMC documentation, this book is available in printed and online formats. To request printed books or to view online books and notices (such as release notes and technical bulletins), see the support website at http://www.bmc.com/support.

Note

Online books are formatted as PDF or HTML files. To view, print, or copy PDF books, use the free Adobe Reader from Adobe Systems. If your product installation does not install the reader, you can obtain the reader at http://www.adobe.com.

The software also offers online Help. To access Help, press F1 within any product or click the Help button in graphical user interfaces (GUls).

Related publications

From the BMC Support Central website, you can use the following methods to access related publications that support your product or solution:


- View Quick Course videos (short overviews of selected product concepts, tasks, or features), which are available from the following locations:
  — Documentation Center (primary center and secured center)
— Support Central (at http://www.bmc.com/support/mainframe-demonstrations)

— BMC Mainframe YouTube channel (https://www.youtube.com/user/BMSCoftwareMainframe)


You can order hardcopy documentation from your BMC sales representative or from the support site. You can also subscribe to proactive alerts to receive e-mail messages when notices are issued.

**Tip**

You can access the BMC Support Central site at http://www.bmc.com/support.

---

**Conventions**

This document uses the following special conventions:

- All syntax, operating system terms, and literal examples are presented in this typeface.

- Variable text in path names, system messages, or syntax is displayed in italic text: `testsystest/instance/fileName`

- Menu sequences use a symbol to convey the sequence. For example, **Actions => Create Test** instructs you to choose the **Create Test** command from the **Actions** menu.

---

**Syntax statements**

This topic explains conventions for showing syntax statements.

A sample statement follows:

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

The following table explains conventions for syntax statements and provides examples:
<table>
<thead>
<tr>
<th><strong>Convention</strong></th>
<th><strong>Example</strong></th>
</tr>
</thead>
</table>
| Items in italic type represent variables that you must replace with a name or value. If a variable is represented by two or more words, initial capitals distinguish the second and subsequent words. | *alias*
  *databaseDirectory*
  *serverHostName*
| Brackets indicate optional items. 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. | 
  `{tableName, columnName, field}
  [-full, -incremental, -level]`
| Braces indicate that at least one of the enclosed items is required. Do not type the braces when you enter the item. | 
  `{DBDName | tableName}
  UNLOAD device={disk | tape, fileName | deviceName}
  {-a | -c}`
| A vertical bar means that you can choose only one of the listed items. In the example, you would choose either `commit` or `cancel`. | 
  `{commit | cancel}`
| An ellipsis indicates that you can repeat the previous item or items as many times as necessary. | `columnName...`
Introduction

These topics introduce the BMC Software BATCH CONTROL FACILITY (BCF) component of the APPLICATION RESTART CONTROL (AR/CTL) product.

Note
The collective term AR/CTL products refers to AR/CTL for IMS, AR/CTL for DB2, AR/CTL for VSAM, the BCF component of AR/CTL, and the common AR/CTL panels and programs that support these products.

BCF environment

The BCF environment includes the BCF component of AR/CTL, AR/CTL services and facilities, and the AR/CTL products.

The products work under the IBM z/OS operating system.

BATCH CONTROL FACILITY component

The BCF component of AR/CTL assists in the management of batch DL/I jobs, improves the performance of the batch logging process, and provides useful statistics for application performance tuning.

Although BCF works with other restart products, it is not by itself a restart product. It does not supply the IMSLOGR DD statement and the restart checkpoint ID for a job that is restarted.

APPLICATION RESTART CONTROL services and facilities

The BMC Software AR/CTL products share a set of common services and facilities.

The AR/CTL application supervisor handles communication and coordination among the products in the application environment, calling AR/CTL products as
appropriate for the application situation. AR/CTL modules also perform general functions, such as ISPF common services and communication with BMC Software subsystems for management of control information.

For more information about these services, see the APPLICATION RESTART CONTROL Administrator Guide.

APPLICATION RESTART CONTROL

BCF works closely with AR/CTL for IMS.

AR/CTL provides full checkpoint/restart support with less manual intervention than the IMS facility requires. It provides a full replacement for the IMS generalized sequential access method (GSAM) with the AR/CTL application sequential access method (ASAM). The product also works in applications that use DB2, VSAM, and sequential files. AR/CTL and BCF work together to ensure correct backout and restart for IMS batch applications. For more information about AR/CTL, see the APPLICATION RESTART CONTROL User Guide.

BCF components

BCF consists of the following components:

- BMC Software subsystems
- REGISET and REGISET records
- AR/CTL application supervisor
- BCF execution controller
- BCF Physical Logger
- Enhanced processing (Cancel Intercept and Abend Filter)
- History data set and statistical reports
- BCF ISPF interface
Batch utilities

Figure 1: Components of BCF

BMC Software subsystems

BCF uses BMC Software standard subsystems for resource management.

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

- The BMCP establishes supervisory services for the BCSS and allows interception of open, close, attach, link, and load requests in the system. (AR/CTL enhanced processing is used for cancel intercept and abend filter processing for BCF.)
The BCSS manages I/O to the REGISET and manages APF-authorized functions. Three AR/CTL components are available with the BCSS: the IMS component, the non-IMS component, and the VSAM component. The IMS component must be enabled for BCF to operate. (The other components are for AR/CTL functions.) You can use the AR/CTL Status Check utility (program AESUVBCS) to ensure that required AR/CTL products, components, and functions are available for application program execution.

For more information about the BMC Software subsystems, see the APPLICATION RESTART CONTROL Administrator Guide.

Registration data sets and REGISET

A registration data set is a VSAM KSDS that contains the information necessary for central control over various phases of application program processing with BCF and AR/CTL (if active).

During configuration of AR/CTL or other BMC Software products that use the BCSS, you define a set of registration data sets. The BCSS manages the registration data sets.

You can define two types of registration data sets—primary and duplex—and you can define multiple data sets of each type. Each copy contains the same records as the other copies contain, and the BCSS keeps the records and their contents in sync. All primary registration data sets and duplex copies make up the REGISET. Each BCSS uses only one REGISET, but a REGISET can be shared by more than one BCSS.

If the active primary registration data set fails during processing, normal processing can continue if another primary registration data set in the REGISET is available; otherwise, the BCSS stops. You can restore a primary registration data set by using another primary registration data set, a duplex registration data set, or a backup copy of the primary registration data set.

The REGISET contains records that define information about the following elements for BCF and AR/CTL:

- Environment registration
- Program registration
- Program exclusion
- Processing options
- Dynamic allocation of log data sets
- Execution control
- Log volume
If you use AR/CTL with BCF, AR/CTL and BCF must share the REGISET. BCF can share the REGISET with other BMC Software products that use the BCSS (such as DATA ACCELERATOR and the DATA PACKER products). For more information about the REGISET, see the APPLICATION RESTART CONTROL Administrator Guide.

**Note**
An AR/CTL password will enable the AR/CTL products and the BCF component of those products.

If you have a BCF password, you can use that password to run BCF without AR/CTL.

You can also disable AR/CTL with the APPLICATION RESTART CONTROL Active flag in the AR/CTL general processing options record. See the APPLICATION RESTART CONTROL Reference Manual.

### AR/CTL application supervisor

BCF and AR/CTL (if active) execute under the AR/CTL application supervisor, which manages coordination and communication among the products.

The application program also executes under the AR/CTL application supervisor.

### BCF execution controller

The BCF execution controller provides an environment where recovery from an abending batch application program is performed automatically before the job step terminates. Recovery from a system failure occurs when the job step is resubmitted. The execution controller requires services provided by the BCF Physical Logger.

The execution controller offers many features that can improve and enhance the execution environment and performance of batch DL/I jobs:

- Job surveillance during execution
- Automatic batch backout
- Automatic log recovery
- Exchange of log device types

### Job surveillance during execution

The execution controller causes the application to execute as an MVS subtask so that BCF can monitor and record the program execution and termination status.
By tracking the status of the job, BCF can determine whether the job step requires BCF services during abnormal termination of the job step, during application restart after a system failure, and when the job step is using the application reattach services of AR/CTL.

Throughout the execution of the job step, the execution controller notifies you of major processing events, such as invocation of the IMS Batch Backout utility, by sending informational messages to the JES job log data set. You can set a BCF processing option to bypass the notification of certain classes of messages.

The execution controller creates and updates an execution control record in the REGISET to track the status of the job step. BCF uses the execution control record as follows:

- If the application terminates abnormally, the execution controller restores the database environment to a point ready for application restart. Possible actions include invoking the IMS Batch Backout utility and IMS Log Recovery utility. When it has restored the environment, the execution controller deletes the execution control record from the REGISET to prepare for job step restart.

- If the application terminates normally, the execution controller simply terminates the execution environment and deletes the execution control record and log volume records from the REGISET to prepare for the next execution of the job.

If the REGISET does not contain an execution control record at the start of a job step, BCF considers all functions from the previous execution, as well as the application, to be ready for restart or normal execution. If the REGISET already contains an execution control record, BCF determines that a prior execution of the job step has abnormally terminated and BCF has not recovered the environment. Most likely, a previous system failure or an MVS cancel has occurred.

The execution controller enables the automatic restart of jobs, but only from the standpoint of the batch backout and log recovery processes. BCF does not duplicate or restrict the functions provided by AR/CTL for batch processing. (See the APPLICATION RESTART CONTROL User Guide for complete information.)

### Automatic batch backout

The execution controller provides a more efficient, reliable way to handle the batch backout process:

- If, during application processing, the job step abnormally terminates after updating a database, the execution controller automatically invokes the standard IMS Batch Backout utility to back out any uncommitted database updates. The batch backout process is done before termination processing completes.

- If the system fails during the execution of the job step and the application has updated a database, the execution control record in the REGISET indicates
whether the job step requires batch backout. The execution controller automatically invokes the IMS Log Recovery and Batch Backout utilities when the job step is resubmitted for execution.

- When an application updates both IMS databases and DB2 tables, automatic batch backout ensures that they remain synchronized. Without automatic batch backout, backout of uncommitted changes to a DB2 table or to an IMS database may occur at different times.

When the execution controller is active, the normal method for handling a job step or system failure is to resubmit the job after the problem is corrected. The time-consuming and error-prone manual intervention process is no longer necessary. With BCF, it may be feasible to implement automatic job scheduling and operations in the batch application environment.

**Automatic log recovery at job restart**

The execution controller provides a more efficient way to handle the closure of log data sets.

If IMS cannot close the log data sets, the execution controller can automatically invoke the IMS Log Recovery utility and then the IMS Batch Backout utility during abnormal termination of the job step. If a job step is restarted and the previous batch log data set was not closed when the job step last terminated, the execution controller automatically invokes the IMS Log Recovery utility and closes the batch system log data set (SLDS).

**Exchange of log device types**

BCF can reduce JCL changes when you decide to use discard logging and/or migrate to tape logging from DASD logging.

The execution controller can deallocate the log data sets coded in the application program JCL and dynamically allocate replacement log data sets. You have full control over this process by specifying BCF processing options and data set allocation information in the REGISET.

**BCF Physical Logger**

The BCF Physical Logger replaces the IMS routines that perform physical logging.

It can provide significant performance improvements and reduce overall resource usage during the physical logging process. It can be used independently of the execution controller.
The BCF Physical Logger provides all the services requested by the IMS Logical Logger:

- Opening the log data set
- Closing the log data set
- Reading the log data set
- Writing the log data set
- Forcing the end-of-volume (EOV) on the log data set
- Terminating operations in case of emergency (ESTAE)

The BCF Physical Logger incorporates the following features:

- Buffered device support
- Discard logging support
- IMS dynamic backout with tape logging support
- SLDS duality
- Execution statistics

The performance improvements provided by the BCF Physical Logger can be lessened by factors such as database buffers, application design, log block size, and other factors that are specific to an installation.

**Buffered device support**

The BCF Physical Logger removes the IMS restriction that buffered devices (3480 and 3490 tape drives) must use tape write immediate (TWI) mode when used as batch logging devices. This restriction prevents you from taking advantage of the performance improvements offered by buffered tape devices. BCF can write to these devices in buffered mode, yet still ensure data integrity. Through device monitoring and synchronization techniques, the Physical Logger provides the IMS Database Manager with write ahead log tape (WALT) support for database integrity.

**Discard logging support**

For some batch jobs, it may be desirable to bypass the logging of database changes.

However, DBRC requires batch logging if the database is registered and DBRC is active. Without BCF, to bypass logging you must turn off DBRC during the batch job — opening a window for database integrity exposures. With BCF, the BCF Physical Logger can discard the log blocks received from the IMS Logical Logger but inform IMS that the log block was written normally. It also issues the notifications to DBRC that help ensure data integrity.
**Dynamic backout log**

The BCF Physical Logger can create an in-memory dynamic log for use with the IMS dynamic backout process.

The IMS dynamic backout process can handle certain classes of DL/I pseudo abends. It is particularly useful in a block-level sharing environment.

The dynamic log data set written by BCF contains only the database log records that have been written since the last application checkpoint. BCF stores the dynamic log data set in extended private memory or in an MVS/ESA data space.

When using the BCF dynamic log, you can take advantage of the IMS dynamic backout capability while writing the log data sets to tape media. Without BCF, the logs must be written to DASD for the IMS dynamic backout process to work.

When using BCF dynamic log and writing the log data sets to DASD, BCF can write the log blocks in true variable blocked format. It is not necessary to pad the log blocks to the maximum block size to emulate the fixed blocked format required by the IMS dynamic batch backout process. Writing the log blocks in variable blocked format can result in reduced DASD usage for batch job steps that log to DASD and request dynamic backout.

**SLDS duality support**

If the job step JCL must request dual logging, the BCF Physical Logger can attempt to provide SLDS duality. It writes a specified number of blocks to each SLDS prior to forcing an end-of-volume condition on both log data sets. This process is consistent with the process used by the IMS Archival utility.

**Execution statistics**

The BCF Physical Logger collects and reports statistics to help you monitor and tune the performance of batch programs.

The BCF Physical Logger can analyze these statistics and notify you when you can tune the job’s database buffer pools to improve logging to buffered tape devices. In some instances, this feature can also allow you to bypass application performance tuning unless it is really necessary.

**Enhanced processing**

BCF can provide specialized processing that usually is not allowed for batch jobs.
This added capability is available through the enhanced processing routines, which are installed on your system as supervisor call (SVC) routines. The AESENH00 maintenance program, used to install and remove the enhanced processing routines, must reside on the MVS image where you plan to run batch jobs that use enhanced processing, and the program must be executed after each IPL. When the AESENH00 maintenance program is executed, the SVC routines are dynamically loaded in the common area of your system.

**Note**

An IPL is not required each time you install or update BCF. When an IPL is required, BMC explicitly states as such in the associated release notes, technical bulletin, flash, or PTF.

During job step initialization, BCF determines whether the enhanced processing routines are installed. If enhanced processing is not installed, you cannot use cancel intercept or abend filter processing. Once loaded, BCF uses the routines to enhance or enable cancel intercept and abend filter processing when needed.

**Cancel intercept processing**

Cancel intercept is an optional feature of enhanced processing that enables BCF, under certain conditions, to intercept a JES `CANCEL` command and to transform it into a user abend.

By transforming a `CANCEL` command into a user abend, this feature allows BCF to perform batch backout for the application before the job step terminates. Without cancel intercept support, the job step is terminated immediately and batch backout will not occur until the job step is resubmitted.

**Note**

Cancel intercept support is provided for JES `CANCEL` commands only. BCF cannot intercept MVS `CANCEL` commands; you can use an MVS `CANCEL` command when you want an immediate cancel without any additional processing by BCF.

Cancel intercept support is active for a BCF-controlled job step only when an IMS task is executing. IMS tasks include tasks that are controlled by IMS, by the IMS Log Recovery utility, and by the IMS Batch Backout utility. If no IMS task is executing when a JES `CANCEL` command is issued, cancel intercept will not intercept the command and will not transform the command into a user abend.

The following processing rules explain cancel intercept execution:

- If no active IMS task is present for the job step, the `CANCEL` command executes normally (it is not intercepted).
- If IMS is active, the **CANCEL** command is transformed into a U1556 abend.

- If the IMS Log Recovery utility or the IMS Batch Backout utility is executing, cancel intercept requests permission to cancel the job step through message BMC48071A. If the operator allows either of the IMS utilities to abend, the job step will require manual intervention. When the necessary recovery actions have been completed, the job step must be restarted.

- Under unusual conditions, when the processing of a **CANCEL** command could cause unpredictable results, cancel intercept temporarily postpones execution and issues message BMC48077I. The message informs the operator about the cancel intercept processing delay and advises the operator to try again later. Usually, after a few seconds, a **CANCEL** command can be reissued successfully.

- If cancel intercept processing terminates abnormally, the resulting abend is associated with the job step being cancelled, and the job step terminates with the abend associated with the cancel intercept failure. If this happens, the actions that BCF takes are determined by the status of the job step when the abend occurs, and manual intervention may be required.

- If BCF cannot issue a U1556 abend, the job step is terminated with an SB22 abend, which emulates the original JES **CANCEL** command.

To terminate a BCF-controlled job step while cancel intercept is active, issue two successive JES **CANCEL** commands or issue an MVS **CANCEL** command.

To determine the status of a job before you issue a JES **CANCEL** command, use the BATCH CONTROL FACILITY Active Job Display panel.

Cancel intercept support can be installed with or without the other enhanced processing functions available with BCF.

**Abend filter processing**

Abend filter processing allows BCF to perform batch backout for applications that have a step parameter (**,STEP**) coded in an abend macro. Without this function, BCF cannot perform batch backout for these applications before job step termination.

**History data set and statistical reports**

BCF can use a history data set to maintain information about log data sets and generated statistics from executions of application programs. The history data set is VSAM KSDS. It contains statistics records and, if you choose to keep them, log
volume records that provide information about the IMS log data sets created during job step execution.

During the execution of the job step, BCF collects and reports many statistics that are difficult or impossible to obtain without BCF. It can write the statistics to the history data set, where you can view them using the BCF ISPF interface. BCF can also write the statistics to a sequential data set that can be printed or used as input to a statistical analysis package.

For more information, see “Using BCF reports” on page 191.

**BCF ISPF interface**

BCF provides an ISPF interface for accomplishing most user tasks.

The BCF ISPF interface consists of a set of CUA-compliant ISPF panels and their supporting dialogs. You can use the BCF ISPF interface to perform the following tasks:

- register environments to work with BCF
- register programs for BCF participation
- exclude programs from BCF participation
- set processing options at global, system, and job step levels
- provide dynamic allocation skeleton information
- view and modify execution control information
- view reports
- access AR/CTL common services

“Getting started with the BCF ISPF interface” on page 45 provides more information about the structure and use of the BCF ISPF interface.

**Batch utilities**

BCF provides batch utilities to help you handle specific situations that may arise in the batch backout and logging environment.

For more information about these utilities, see “Using the log utilities” on page 233.

**BCF Log utility**

The BCF Log utility helps you analyze BCF operations.
You can also use it as an *emergency* solution to some recovery requirements, for example, to run the IMS Batch Backout utility and back out the complete execution of a job step when the application has used checkpoint calls but did not begin execution with a checkpoint call. You can also use it to perform backout for a batch message processing (BMP) execution because the IMS Batch Backout utility does not support complete backout for a BMP that has participated in sync point processing.

The BCF Log utility provides the following functions:

- statistics (*STATS*)
- logical compare (*COMPARE*)
- batch backout log create (*BCREATE*)
- online backout log create (*OCREATE*)
- find online BMP executions (*FIND*)

**Database Changed Block Reconciliation utility**

The Database Changed Block Reconciliation utility is an extension of the BCF Log utility.

You can use to identify database blocks that have been changed by more than one unit of work. The utility produces a report that lets you determine whether executing the Batch Backout utility using the log data set created by the *OCREATE* function of the BCF Log utility is likely to result in a damaged database.

**Other features of BCF**

This section describes other BCF features that were not covered elsewhere.

**Conditional backout step bypass**

The batch job JCL may contain conditional batch backout steps to be executed if an IMS application fails.

Since BCF provides batch backout prior to step termination, usually the conditional step should not be allowed to execute.

If BCF is used for the conditional backout step, BCF can bypass the conditional backout step when the environment has been recovered. You do not need to make JCL changes to obtain this support, and you can return to your prior execution practices if BCF is disabled.
During batch backout processing, BCF determines whether the job contains any subsequent steps marked as COND=ONLY. If no COND=ONLY step exists, processing continues normally. If BCF finds a COND=ONLY step, BCF converts the final step termination code to a U628 (or U1574) abend if batch backout is successful. If batch backout is not successful, BCF converts the code to a U629 (or U1577) abend. The abend code depends on the value you specified for the Use Abend Code Series option. For more information, see “Use Abend Code Series” on page 108.

When the execution controller initializes for a conditional batch backout step, BCF bypasses the step if the job has abended with a U628 (or U1574) abend or a U629 (or U1577) abend. The logic for conditional step bypass includes abends U1550 and U1552.

### Dummy logging environment support

Dummy logging environment support allows you to gather statistics for IMS batch jobs that do not execute in a valid IMS logging environment, such as jobs with log data sets specified as DUMMY.

Using this feature, you can create a dummy logging environment for IMS so that some, but not all, of the logging functions are performed by IMS. This feature enables BCF to provide run-time execution statistics for read-only jobs. It also enables BCF to detect database updates in nonlogging environments and to notify users when they occur.

**Note**

DUMMY logging is not supported for utility (ULU) regions.

### Batch Backout Assist feature

BCF offers the Batch Backout Assist feature to improve the performance of read-backward processing for log data sets.

It enhances any type of IMS batch backout operation, whether the log is on tape or DASD, but is most helpful when buffered tape with the IDRC recording mode is used for the batch log data set on 3480/3490 tape devices.

When you use IDRC recording on 3480/3490 devices, a standard read-backward operation is not possible. Instead, the hardware and MVS software emulate a read-backward operation by backspacing blocks and doing a forward read. The Batch Backout Assist feature removes the requirement to read the tape data set backward by buffering the input log data set blocks in an MVS/ESA data space during the
forward read. When the read-backward operation starts, the records are retrieved from the data space that is emulating the read-backward operation.

IMS performs batch backout by reading one block at a time (no matter what the size of the block), and then reads forward to find the beginning of the log records. BCF, through the Batch Backout Assist feature, allocates MVS/ESA data spaces (64 KB each) to create sets of prechained buffers. While IMS is reading records from the log, BCF is reading blocks into the buffers, improving performance; after the first forward read into the data space, no I/O is required.

The blocks remain in the buffers until an application checkpoint log record is read. At a checkpoint, BCF resets the buffers to be reused.

The Batch Backout Assist feature requires that MVS/ESA data spaces be available for processing batch. Batch Backout Assist obtains multiple data spaces as needed to provide for the buffering of log records during backout. The size and number of data spaces allowed for these batch jobs is installation-dependent.

The amount of storage required is a function of the following:

- the size of the IMS log data set supplied as input to the Batch Backout utility
- the number of IMS checkpoint records on the log data set

BCF obtains unrestricted MVS/ESA data space storage, which allows BCF to operate at maximum effectiveness when providing backout assistance for large batch backouts. In some cases, BCF requires authorization provided by the BCSS to obtain enough data space storage for Batch Backout assist. If MVS/ESA denies a request for data space storage, the Batch Backout utility terminates abnormally.

You can disable the Batch Backout Assist feature for problem determination or for performance comparisons by including the `BCFBAIGN DD DUMMY` statement in the job step JCL.

Long-running batch jobs that use logging but do not use application checkpoints may cause problems if used with the Batch Backout Assist feature by obtaining too much storage. To prevent these problems, you can turn off the feature with the DD statement, use application checkpoints, or use the BCF Discard Logging option (back up the database before the job, and recover the data base from the image copy in case of an abend). If you use AR/CTL, you may be able to implement checkpoints without program or JCL changes; see the APPLICATION RESTART CONTROL User Guide for details.

**SVC99 retry processing**

SVC99 retry processing enables BCF to issue a WTOR when the requested number of devices are not available, preventing dynamic allocation of log data sets.
When a WTOR is issued, the operator has the option of waiting until the requested number of devices are available. Without SVC 99 retry processing, dynamic allocation automatically fails when the requested number of devices is unavailable.

**GDG support**

BCF supports the use of generation data groups (GDGs) for log data sets.

If you use GDGs for log data sets, you need to understand how BCF handles dynamic allocation of new GDG log data sets during execution. If you use GDGs to provide for unique log data set names, you may want to substitute the BCF techniques for providing unique log data set names. For more information, see “Generation data groups” on page 137.

**Fast OSAM Buffer Purge**

The Fast OSAM Buffer Purge feature can improve performance for batch update jobs that use OSAM databases.

Fast OSAM Buffer Purge dynamically monitors the OSAM buffer stealing process. When the buffer steal routines detect the need to steal an altered buffer, Fast OSAM Buffer Purge causes the entire OSAM buffer pool to be written back to the database using the OSAM queued write process. Fast OSAM Buffer Purge invokes standard IMS routines to cause the buffer purging.

The most benefit will be achieved for those batch jobs that do not issue IMS checkpoints or for jobs where the checkpoints are not issued often enough to prevent stealing of OSAM buffers. If checkpoints have been added to an application strictly to cause purging of altered OSAM database buffers, it may be possible to remove the checkpoints because Fast OSAM Buffer Purge causes buffer purging dynamically as needed. Using application checkpoints to purge OSAM buffers can result in added overhead without maximum benefit. Fast OSAM Buffer Purge does not create an application checkpoint.

Fast OSAM Buffer Purge is activated only if the necessary intercept points can be established in existing IMS modules. If the required intercept points cannot be established, Fast OSAM Buffer Purge is not activated and message BMC48063I is issued, allowing the job to continue processing without Fast OSAM Buffer Purge.
Summary of benefits

BCF provides many benefits related to three critical goals of data management:

- Data integrity ("Data integrity" on page 31)
- Data availability ("Data availability" on page 32)
- Resource conservation ("Resource conservation" on page 33)

Data integrity

BCF helps to preserve data integrity as follows:

- BCF automates the error-prone batch backout process. The execution controller ensures that the IMS Batch Backout utility is always executed when necessary and helps reduce the possibility of executing batch backout when it is not necessary. It also ensures that the correct log data sets are used as input to the IMS Batch Backout utility.

- BCF allows you to force DBRC usage in an IMS system without forcing batch applications to use logging. The BCF Physical Logger can discard the log records and handle the necessary notifications to DBRC to provide data integrity. Forcing DBRC usage provides a high level of data integrity, and BCF discard logging helps improve the performance of long-running batch applications.

- BCF usually accomplishes the batch backout process before an abending job step terminates and releases the databases. Therefore, it reduces the chance of another job step accessing the incorrectly changed database and helps preserve data integrity.

- BCF improves synchronization between IMS databases and DB2 tables that are updated by the same batch application. DB2 always backs out uncommitted changes to the DB2 table immediately, but IMS does not do so for a batch program. Therefore, the IMS database and DB2 table remain out of sync until the IMS batch backout process is executed. By automatically performing IMS batch backout at the same time DB2 backs out table changes, BCF reduces or eliminates out-of-sync conditions.

- BCF provides easy and flexible ways to change log devices without JCL changes. Using the Exchange Log Device Type option, you can switch from DASD to tape logging or vice versa without JCL changes. This feature can help ensure that batch update jobs do not execute without log data sets.
**Data availability**

BCF improves data availability as follows:

- BCF makes the implementation of DBRC easier because it can ensure that no IMS DL/I batch update jobs execute without log data sets. This includes jobs that are actually read-only jobs but have incorrectly indicated update intent in the program specification block (PSB). The prevention of abends for DL/I batch jobs significantly reduces the implementation effort for DBRC migration.

- The Batch Backout Assist feature reduces the time needed to perform batch backout. It allows the use of 3480/3490 IDRC recording mode for IMS log data sets. The Batch Backout Assist feature removes the need for the IMS Batch Backout utility to read a tape data set backwards during the backout process. BCF also allows controlling the use of IDRC using a product option regardless of current JCL specifications.

- BCF includes a comprehensive utility to perform analysis and recovery operations using the IMS log data set. The utility provides you with some alternative recovery and repair options because it allows backout of BMPs and transaction updates even if they have completed normally.

- BCF allows logging to buffered (3480 and 3490) tape devices in the faster and more efficient buffered mode. Faster logging improves the performance of the batch application program. Since the application program finishes faster, the database is available for other processing sooner.

- BCF allows you to bypass logging in a DBRC=FORCE environment. The performance of the batch application is improved without sacrificing the integrity that DBRC ensures.

- BCF automates the batch backout and log closure processes to bypass the delays inherent in manually performing these processes. In installations without BCF, an operator often has to contact the DBA or application programmer who, in turn, must determine whether to run the batch backout and/or log recovery processes. Then the DBA or application programmer must determine which logs to use as input and set up the JCL to execute it. BCF automates these time-consuming steps so that the database is available for other processing more quickly, and the failed job step can simply be resubmitted.

- BCF automatically detects situations when the performance of the batch job is degraded by logging characteristic and issues a message to report the situation. You can then investigate tuning the application to improve performance.
Resource conservation

BCF conserves valuable data processing resources with the following benefits:

- BCF reduces DASD use for IMS DL/I batch jobs by allowing tape logging with acceptable performance.

- BCF creates a near-BMP environment for the execution of IMS DL/I batch jobs. The execution environment includes automatic log recovery and batch backout. These recovery actions are provided for abended job steps and for restarts following a system failure. Therefore, BCF allows the choice of BMP versus DL/I batch to be made on the basis of resource consumption and data availability and not on operational considerations. The DL/I batch environment often uses considerably fewer processor resources than does the BMP environment.

- BCF can reduce the cost of managing test databases for application and maintenance development. Since BCF can automate backout for abending test executions, you can avoid total recoveries or restores following testing abends.

- BCF permits using buffered tape devices in a buffered mode, allowing these devices to operate in the most efficient manner.

- BCF permits the IMS dynamic batch backout process to work when the log records are written to a tape device. Without this feature, IMS requires the log data sets to reside on DASD for dynamic backout purposes (from which they must be archived to tape at additional expense). With this feature, IMS can perform dynamic batch backout for DL/I pseudo abends, which are common in a database block-level data sharing environment.

- BCF aids in application development by reducing the work load of DBAs, application programmers, and operators who must perform the batch backout process manually without BCF. It can also reduce the duration of the application program development cycle. Without BCF, programmers spend a substantial amount of time restoring test databases during the coding and testing of new applications. Since BCF automatically restores the environment when an application abends, the programmer can concentrate on finding and correcting the cause of the abend.

- BCF provides easy and flexible ways to change log devices without JCL changes. Using the Exchange Log Device Type option, you can switch from DASD to tape logging or vice versa without JCL changes. This feature conserves resources that would be spent coding JCL.

- BCF provides an ISPF interface and a batch utility that can be used to automate the recovery and restart of IMS DL/I batch job steps that were active when a system failure occurred. The functions provided include automatically cataloging IMS log data sets that were not cataloged because of a system failure and resetting the job step status in the REGISET to an appropriate status so that BCF can restart...
the job. BCF also provides a means to accomplish log recovery and batch backout for interrupted jobs without requiring the applications to be restarted.

Considerations for using BCF

This section describes considerations for using BCF in your environment.

For a discussion of the system resources required for BCF, see the Installation System documentation.

General considerations

The following general considerations apply to BCF.

**BMP RESTART**

BCF does not support the BMC Software BMP RESTART product.

BMP RESTART was replaced by AR/CTL. BCF fully supports AR/CTL.

IMS support

BCF supports IMS as follows.

**IMS requirements**

Your installation must have a licensed copy of the IMS Database Manager.

BCF is supported for use with all IBM-supported versions of IMS. For complete information about BCF support for operating systems, hardware, and database management systems, you can use the Product Availability and Compatibility (PAC) utility on the BMC Support Central website at http://www.bmc.com/support.

BCF supports all PSB PROCOPTs and DL/I databases.
DBRC

BCF fully supports DBRC. It does not replace or enhance any existing function of DBRC. The BCF Physical Logger provides the same level of DBRC support as the current IMS logging function provides.

Region types

BCF can participate in the execution of IMS batch programs that use these types of IMS regions:

- DLI
- DBB
- ULU

In a utility (ULU) environment that requires logging, the BCF Physical Logger can service the logging requests. When BCF is processing in a ULU environment, the execution controller initializes but does not perform automatic batch backout or exchange log device types. DUMMY logging is not supported for ULU regions.

BCF does not participate in BMP or MPP executions.

BCF participation in manual batch backout

When the IMS Batch Backout utility is executing in a DLI environment, the execution controller initializes but does not perform automatic batch backout or exchange log device types.

However, the Batch Backout Assist function remains active. If the IMS Batch Backout utility abends, the execution controller issues message BMC44242I with reason code 18 to indicate that manual intervention is required. It then deletes the execution control record and the log volume record(s) from the REGISET.

**WARNING**

For the execution controller to recognize the IMS Batch Backout utility, the program name DFSBBO00 must be specified on the EXEC statement.

Uncataloged data sets in STEPLIB

Uncataloged or temporary data sets that are included in the STEPLIB concatenation are not supported.

AR/CTL products are excluded from execution when the STEPLIB contains an uncataloged or temporary data set.
Automated operations

One of the primary objectives of BCF is to allow you to increase the level of automation in the DL/I environment.

The design of BCF, and in particular the Automatic Batch Backout option, focuses on reducing the amount of manual intervention necessary to recover from application abends or system failures.

The wide range of error, exception, and informational messages issued by the execution controller can also enhance automated operations. Products that specialize in automated job scheduling and restart can intercept and interpret these messages for more effective automated control over batch processing. The following features of BCF messages aid in automated operations:

- The messages attempt to identify resource usage and product functions so that the messages are more easily interpreted by automated methods.

- When reporting error or exception conditions that can have multiple meanings, the messages contain reason codes to help identify the reason for the message. This feature is intended to allow for the automatic notification of support personnel or submission of recovery jobs based on specific messages.

- In addition to encoded reason codes, several error conditions have a variety of expanded error messages providing variable text concerning the reason for the error.

- You can implement automated checking of BCF messages without concern that BCF maintenance and enhancements will adversely affect them. Text and return code offsets in existing messages will not be changed if at all possible.

Since the needs of automated operations environments are varied and unique, feel free to contact BMC Software to communicate your needs and requirements in this area.
Getting started

This section describes how to get started using the BMC Software BATCH CONTROL FACILITY (BCF) component of APPLICATION RESTART CONTROL (AR/CTL).

Overview of BCF component tasks

IMS DBAs, IMS system programmers, and IMS application programmers can be involved in installing, implementing, using, and maintaining BCF.

Table 1 on page 37 surveys the tasks these people can perform and indicates the location of complete information about these tasks.

Table 1: BCF component tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
</table>
| install AR/CTL, and process CPU ID passwords | ■ Use the Installation System (the Installation System was formerly called the OS/390 and z/OS Installer) to install AR/CTL.  
■ Use the AR/CTL batch interface to process a CPU ID password that authorizes the product to run on your processor. | the Installation System documentation |
| configure AR/CTL                           | Use the AR/CTL Installation System to configure AR/CTL.                      | APPLICATION RESTART CONTROL Customization Guide |
| understand the BCF component              | You should understand BCF processing and how BCF can change your operational procedures. | “Introduction” on page 15                      |
| get started with BCF                      | You must use BCF components to set up the product to run in your environment; this setup includes providing access to the BCF component execution modules so that they are available during the execution of the job step. | “Getting started” on page 37                    |
Getting started with BCF installation

This section summarizes the tasks required to install BCF.

The BCF installation process includes the following tasks:

- create the BCF component and AR/CTL product libraries by using the Runtime Enablement (RTE) process during the installation of the AR/CTL products
- install the BMC Software subsystems
- allocate the registration data sets (collectively called the REGISET) which contain BCF control and execution information, and initialize this data set with global records
- enable users to access and use the BCF and AR/CTL ISPF interfaces
- install AR/CTL enhanced processing (optional)
- install the CPU ID authorization password issued by BMC Software to permit product execution
- execute installation verification procedures (IVPs)

See the Installation System documentation and the APPLICATION RESTART CONTROL Customization Guide for more information.
Working with BMC Software subsystems

AR/CTL products use the BMC Primary Subsystem (BMCP) and the BMC Consolidated Subsystem (BCSS).

These subsystems manage resources for AR/CTL and other BMC Software products.

For more information about BMC Software subsystems and the topics discussed in this section, see the APPLICATION RESTART CONTROL Administrator Guide.

**BMCP**

The BMCP establishes supervisory services for the BCSS and many BMC Software products.

It allows interception of open, close, attach, link, and load requests in the system. (AR/CTL enhanced processing is used for cancel intercept and abend filter processing.) Because BMC Software products share the BMCP, you need only one copy of the BMCP on your system. All associated BMC Software products continue to operate normally even if the BMCP terminates; however, BMC Software recommends that you leave the BMCP running at all times. During configuration of AR/CTL or other products that use the BMCP, the BMCP is set up to start at each IPL.

**BCSS**

The BCSS manages I/O to the registration data sets (collectively called the REGISET), manages APF authorized functions, and performs processing for intercepted open, close, attach, link, and load requests.

The BCSS must be active on the MVS system where you want to execute application programs that use AR/CTL products and where you want to access records in the REGISET through the ISPF interface. During configuration of AR/CTL or other products that use the BCSS, the BCSS is set up to start at each IPL.

You can use the AR/CTL Status Check utility (program AESUVBCS) to ensure that required AR/CTL products, components, and functions are available for application program execution.

More than one BCSS can be active on an MVS system. One must be defined as the public BCSS; it is typically used for production work. Each of the others is defined as a private BCSS; a private BCSS is typically used for testing new releases of AR/CTL.
A default BCSS can be designated for each MVS. You use the BCSS identifier (BCSID) to identify the subsystem (and REGISET) you want to use.

**Working with the REGISET**

This section describes the REGISET and the records it contains.

It also explains how to use wildcard characters in the record keys.

For more information about the REGISET, see the *APPLICATION RESTART CONTROL Administrator Guide*.

**The REGISET**

The REGISET (the collection of registration data sets defined during the AR/CTL configuration process) is a repository for BCF information about executing application programs and how you want BCF to handle them.

A BCSS can use only one REGISET, but the same REGISET can be used by multiple BCSSs.

The REGISET can consist of as few as 2 (or as many as 22) registration data sets, each of which is a VSAM key-sequenced data set (KSDS). You can have two types of registration data sets—primary and duplex:

- **BCSS uses a primary registration data set** when providing AR/CTL products with responses to REGISET queries. You must have at least one primary registration data set; you can have as many as 11. If you have more than one and the active primary registration data set fails, the next primary registration data set is available and normal processing continues. You can recover the failing data set at a more convenient time, and recovery of the failing data set does not require restoration. If you encounter a failure in a primary registration data set and no other primary registration data set is available, the BCSS stops.

- **BCSS uses a duplex registration data set** to maintain an active backup copy of the active primary registration data set. Use of a duplex registration data set is optional; you can define as many as 11. Duplex registration data sets are used only for recovering primary registration data sets.

You can restore a primary registration data set by using another primary registration data set in the REGISET, a duplex registration data set in the REGISET, or a backup copy of the primary registration data set.
Types of records

When you set up BCF, you create records in the REGISET.

These records contain information about how you want BCF to work in your environment and during the execution of an application program. Table 2 on page 41 describes the types of records BCF uses.

Table 2: BCF record types

<table>
<thead>
<tr>
<th>Record type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR/CTL global options</td>
<td>Contains the values that control the use of AR/CTL products. The AR/CTL global options record is created during AR/CTL product configuration. You can use the ISPF interface to update the record. This record is required to be in the REGISET for BCF to work.</td>
</tr>
<tr>
<td>IMS environment registration</td>
<td>Contains the values that provide access to AR/CTL product load modules for application programs that use IMS. You can use this type of record to avoid changes to application program JCL. The global IMS environment registration record is created during AR/CTL product configuration. You can use the ISPF interface to update the global record and to create and update other records of this type.</td>
</tr>
<tr>
<td>IMS program registration</td>
<td>Contains the values that control AR/CTL product participation in the execution of an application program that uses IMS. You can use this type of record to avoid changes to application program JCL. The global IMS program registration record is created during AR/CTL product configuration. You can use the ISPF interface to update the global record and to create and update other records of this type.</td>
</tr>
<tr>
<td>IMS program exclusion</td>
<td>Prevents AR/CTL products from participating in the execution of an application program that uses IMS. You create and update this type of record through the ISPF interface.</td>
</tr>
<tr>
<td>global options record</td>
<td>Contains the values for BCF options that you want to use as default values if no other options record overrides the values. The global options record is created during AR/CTL configuration. You can use the BCF ISPF interface to update the global options record. This record is required to be in the REGISET for BCF to work.</td>
</tr>
<tr>
<td>system options record</td>
<td>Contains the values for BCF options that you want to use for a given IMS subsystem. The IMS subsystem is defined by the IMSID associated with the batch IMS system. At execution time, BCF merges the option values stored in this record with the options values stored in the global options record. You can use the BCF ISPF interface to create and update system options records.</td>
</tr>
<tr>
<td>job step options record</td>
<td>Contains the values for BCF options that you want to use for a job step, as identified by the job name, the step name, and the PSB name or program name. At execution time, BCF merges the option values stored in this record with the options values stored in the global options record and in the system options record (if any). You can use the BCF ISPF interface to create and update job step options records.</td>
</tr>
<tr>
<td>Record type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>execution control record</td>
<td>Contains information used by the execution controller to track information from an individual batch job step execution. The execution controller creates the record when the job step makes the first database update and updates it at processing milestones such as log closure, completed batch backout, and job termination without error. If the job step completes normally or if the execution controller successfully performs the batch backout process for an abending job step, the execution controller deletes the execution control record. If a system failure occurs or if the execution controller could not successfully perform the batch backout process, BCF does not delete the execution control record. When the job step is submitted again, the presence of the execution record tells the execution controller to perform the batch backout process. BCF uses the information in the execution control record to determine which actions are necessary to restore the databases and logs and to restart the job step.</td>
</tr>
<tr>
<td>primary log volume record</td>
<td>Contains the information about the IMS primary log data set created by the job step. BCF creates a primary log volume record for each job step at the same time it creates an execution control record. BCF updates the record during execution of the job step when changes occur to the log data set—during log end-of-volume processing, log closure, and log error processing. BCF uses this record to provide log volume information during job step restart after a processor or system error and to report log volume information (in the Log Data Set Volume report). BCF maintains and logically pairs the primary log volume record with the statistics record. BCF deletes the log volume record from the REGISET when the record is no longer needed for recovery or backout. If the Delete Volume Records option is set to N, BCF writes the log volume record to the history data set before deleting the record.</td>
</tr>
<tr>
<td>secondary log volume record</td>
<td>BCF creates the secondary log volume record at the same time it creates the execution control record. This record contains information about the IMS secondary log data set created by the job step. BCF maintains and uses this record in the same way as the primary log volume record.</td>
</tr>
<tr>
<td>dynamic allocation record</td>
<td>You create allocation records through the BCF ISPF interface. Each record is associated with a ddname (BCF supports only IEFRDER and IEFRDER2) and with an IMSID and/or a job name, step name, and program or PSB name. The record contains allocation information that BCF uses when dynamically allocating a data set during execution of a job step. At least one record that applies to the job step is required to be in the REGISET for BCF to work. You can create allocation records at the global level, the system level, and the job step level. At the system level, you can create allocation records for specific device classes (DASD, standard tape, buffer tape, and so on) and for a default class. If multiple records exist for the same ddname, BCF gets the dynamic allocation information from the record specified at the lowest level (the job step or system level). BCF does not merge information from records at different levels, as it does for processing options.</td>
</tr>
</tbody>
</table>
Record key qualifiers

The record key of each record in the REGISET contains fields that determine whether BCF will select the record to include in a limited list of records and during execution of the application program.

When you create the record, you fill in the fields with values that will match information from the application program execution. This information includes *qualifiers*, such as the MVS system ID, the job name from the JCL, and the program name.

A record type may not use all possible qualifiers in its record key. An execution may not have a value for a qualifier; for example, an execution that does not use a JCL procedure will have a blank procedure name qualifier.

Most BCF record keys can contain some or all of the qualifiers listed in Table 3 on page 43.

### Table 3: Valid qualifiers for record keys

<table>
<thead>
<tr>
<th>Qualifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDNAME</td>
<td>For system dynamic allocation records, the ddname, which can be IEFRDER or IEFRDER2. For more information, see “Understanding BCF dynamic allocation” on page 113.</td>
</tr>
<tr>
<td>PROGNAME</td>
<td>The program name (1 to 8 bytes) of the executing application program.</td>
</tr>
<tr>
<td>PROCSTEP</td>
<td>The procedure step name (1 to 8 bytes) from the JCL procedure. If the execution does not use a JCL procedure, this qualifier is blank.</td>
</tr>
<tr>
<td>JOBSTEP</td>
<td>The jobstep name (1 to 8 bytes) from the EXEC statement in the JCL to execute the jobstep.</td>
</tr>
<tr>
<td>JOBNAME</td>
<td>The job name (1 to 8 bytes) from the JOB statement in the JCL to execute the application job.</td>
</tr>
<tr>
<td>PSBNAME</td>
<td>The name (1 to 8 bytes) of the program specification block (PSB) used by the application program. A PSB is an IMS control block that is external to the application and is used to exchange information between the application and IMS or BCF. A PSB consists of one or more PCBs and is built during the PSBGEN process.</td>
</tr>
<tr>
<td>PSB/PGM</td>
<td>The name (1 to 8 bytes) of the PSB used by the application program or the name of the application program. Whether this qualifier contains the PSB name or the program name depends on the value of the Qualify with Program option. In a REGISET, all records must use the same Qualify with Program value. For more information, see “Qualify with Program” on page 83.</td>
</tr>
<tr>
<td>IMSID</td>
<td>The IMSID (1 to 4 bytes) set during the SYSGEN of the IMS system or the IMSID set on the EXEC statement in the job step JCL.</td>
</tr>
<tr>
<td>MVSID</td>
<td>The SMF system ID (1 to 4 bytes) of the MVS system where the application program executes.</td>
</tr>
</tbody>
</table>
Wildcard characters in record keys

To reduce the number of records you need to set up and maintain, BCF supports wildcard characters in some qualifiers of the REGISET record keys.

BCF uses the following symbols for wildcard characters:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>asterisk - trailing multiple character match</td>
</tr>
<tr>
<td>%</td>
<td>percent sign - leading or imbedded single character match</td>
</tr>
</tbody>
</table>

You can use the wildcard characters alone or in combination with valid specific characters, according to these rules:

- For BCF records, you can enter a percent sign in any position of the qualifier. For AR/CTL records, you can enter a percent sign in any position of the qualifier except the last position, and you cannot immediately follow the percent sign with an asterisk. BCF accepts any character in the position occupied by a percent sign when determining which records to select. One or more percent signs at the end of a specific string causes a match for the qualifier that contains only the specific string, plus those that start with the specific string followed by as many characters or blanks as the number of percent signs.

- You can enter an asterisk in the trailing position after a specific string or in the first position of a qualifier; no specific characters or percent signs may follow an asterisk. The asterisk is a match for any character. One or more asterisks at the end of a specific string makes BCF accept any length of qualifier as long as the specific string matches.

- If you enter a string containing only wildcards, any value in the qualifier is a match.

- On a panel that limits the list of BCF records to display, you can enter a single asterisk in each qualifier field. BCF displays all records of the selected type in the REGISET. You can enter an asterisk in some qualifier fields and a specific string in other qualifier fields. BCF displays all records (of the selected type) that match specific qualifiers. You cannot use wildcards and specific strings in the same qualifier on a panel that limits the list of BCF records to display.

- On a panel that limits the list of AR/CTL records to display, you can enter a string containing only wildcards for all qualifiers. BCF displays all records of the selected type in the REGISET. You can use wildcards and specific strings in the same qualifier on a panel that limits the list of AR/CTL records to display.

Table 4 on page 45 shows examples of wildcard usage for the 1- to 4-byte IMSID qualifier. The same principles apply to longer qualifiers.
Table 4: Wildcard usage examples

<table>
<thead>
<tr>
<th>Example</th>
<th>Discussion</th>
</tr>
</thead>
<tbody>
<tr>
<td>%ABC</td>
<td>matches all 4-byte IMSIDs that contain ABC in positions 2 through 4</td>
</tr>
<tr>
<td>A%B</td>
<td>matches all 3-byte IMSIDs that start with A, contain any character except blank in position two, and end with B</td>
</tr>
<tr>
<td>AB%</td>
<td>matches IMSID AB and all 3-byte MVSIDs that start with AB</td>
</tr>
<tr>
<td>ABC%</td>
<td>not valid in AR/CTL records because percent sign may not be in last position; specify ABC* instead; valid in BCF records</td>
</tr>
<tr>
<td>A*</td>
<td>matches all IMSIDs of any length that start with A; BCF displays the qualifier as A*** after padding the end with asterisks</td>
</tr>
<tr>
<td>ABC*</td>
<td>matches all IMSIDs that start with ABC</td>
</tr>
<tr>
<td>*AB</td>
<td>not valid because asterisk may not immediately precede specific characters or percent sign</td>
</tr>
<tr>
<td>A*B</td>
<td>not valid in AR/CTL records because an asterisk may not follow a percent sign; valid in BCF records</td>
</tr>
<tr>
<td>A**%</td>
<td>matches all IMSIDs</td>
</tr>
<tr>
<td>****</td>
<td>matches all 3- or 4-byte IMSIDs that start with A and have B in position 3</td>
</tr>
<tr>
<td>% % A*</td>
<td>matches all 3- or 4-byte IMSIDs that have A in position 3</td>
</tr>
<tr>
<td>ABCD</td>
<td>matches the IMSID ABCD only</td>
</tr>
</tbody>
</table>

Getting started with the BCF ISPF interface

The BCF ISPF interface provides an easy way to view statistical reports, set product options, define data set allocation information, and maintain the product components. This section provides overview information about the ISPF interface.

Setup requirements for your ISPF session

To use the ISPF interface, your ISPF session must have access to the AR/CTL product libraries containing panel dialogs and messages.

The libraries were created by the Runtime Enablement (RTE) process during the installation of AR/CTL products. You can include these libraries in your logon procedure, or you can use the ISPF LIBDEF facilities to access them. The data set
names, and methods for accessing them, are documented in the Installation System documentation.

Most functions of the ISPF interface access the REGISET. The BCSS must be active to access the REGISET. The ISPF interface also makes sure that the REGISET contains required records created by the REGISET Population utility (AESURPOP). If the required records are not present, the ISPF interface issues a message and prevents access to the REGISET. For more information about AESURPOP, see the APPLICATION RESTART CONTROL Administrator Guide.

The first time you invoke the ISPF interface (from each TSO user ID you use), the interface requires you to set values for environment-specific ISPF display options (that hide or display elements which specifically relate to IMS, DB2, or VSAM) and for the temporary file unit name. You can also customize the appearance of the panels with the Application Enhancement Series Profile Options panel. For information about these topics, see the APPLICATION RESTART CONTROL Administrator Guide.

Using the Application Enhancement Series primary menu

Use the Application Enhancement Series primary menu to access AR/CTL options, the BATCH CONTROL FACILITY primary menu, the APPLICATION RESTART CONTROL primary menu, and the AR/CTL common utilities.

Figure 2: Application Enhancement Series primary menu

<table>
<thead>
<tr>
<th>AESPRIM</th>
<th>Application Enhancement Series V2.2.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt; _________________________________________________________________</td>
<td></td>
</tr>
<tr>
<td>Select an option. Then press Enter.</td>
<td></td>
</tr>
<tr>
<td>Application Enhancement Series (AES)</td>
<td></td>
</tr>
<tr>
<td>1. AES records</td>
<td></td>
</tr>
<tr>
<td>2. BATCH CONTROL FACILITY (BCF)</td>
<td></td>
</tr>
<tr>
<td>3. APPLICATION RESTART CONTROL (AR/CTL)</td>
<td></td>
</tr>
<tr>
<td>AES Common Options</td>
<td></td>
</tr>
<tr>
<td>11. Display, print, jobcard, allocation, and profile options</td>
<td></td>
</tr>
<tr>
<td>12. Messages</td>
<td></td>
</tr>
<tr>
<td>13. Security</td>
<td></td>
</tr>
<tr>
<td>14. Product authorization</td>
<td></td>
</tr>
<tr>
<td>15. REGISET record copy utility</td>
<td></td>
</tr>
<tr>
<td>16. Exit</td>
<td></td>
</tr>
<tr>
<td>99. New features and changes for the AES family of products</td>
<td></td>
</tr>
</tbody>
</table>

To use the Application Enhancement Series primary menu

1. Type the number corresponding to one of the options in Table 5 on page 47, and press Enter.
Table 5: Application Enhancement Series primary menu options

<table>
<thead>
<tr>
<th>Menu item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Access the AR/CTL options, including AR/CTL global options, environment registration, program registration, and program exclusion. For more information, see “Working with IMS environment registration records” on page 53, “Working with IMS program registration records” on page 55, and “Working with IMS program exclusion records” on page 59.</td>
</tr>
<tr>
<td>2</td>
<td>Access the BATCH CONTROL FACILITY primary menu. See “Using the BATCH CONTROL FACILITY primary menu” on page 47.</td>
</tr>
<tr>
<td>3</td>
<td>Access the APPLICATION RESTART CONTROL (AR/CTL) primary menu. See the APPLICATION RESTART CONTROL Reference Manual for more information.</td>
</tr>
<tr>
<td>11</td>
<td>View information about the operating environment, and set up ISPF and other environment defaults for AR/CTL products. See the APPLICATION RESTART CONTROL Administrator Guide for more information.</td>
</tr>
<tr>
<td>12</td>
<td>View information about AR/CTL product messages. See the APPLICATION RESTART CONTROL messages in the BMC Documentation Center for more information.</td>
</tr>
<tr>
<td>13</td>
<td>Set up the security environment for the AR/CTL products. See the APPLICATION RESTART CONTROL Administrator Guide for more information.</td>
</tr>
<tr>
<td>14</td>
<td>Process CPU ID authorization passwords for the AR/CTL products. See the Installation System documentation for more information.</td>
</tr>
<tr>
<td>15</td>
<td>Use the REGISET Record Copy utility to copy records for AR/CTL products. For more information, see the APPLICATION RESTART CONTROL Administrator Guide.</td>
</tr>
<tr>
<td>16</td>
<td>Exit from the Application Enhancement Series primary menu.</td>
</tr>
<tr>
<td>99</td>
<td>View information about new features and changes to the AR/CTL family of products.</td>
</tr>
</tbody>
</table>

Using the BATCH CONTROL FACILITY primary menu

Use the BATCH CONTROL FACILITY primary menu to select the BMC Consolidated Subsystem (BCSS) and BCF option that you want to work with.
To use the BATCH CONTROL FACILITY primary menu

1 Type or verify the ID of the BCSS to use in the **BMC consolidated subsystem ID** field. If you don’t know which BCSS to use, or if you want to know which ones are active, type and enter \? or blanks for a list of active subsystems; BCF displays the Active BMC Consolidated Subsystems panel or a message stating that no subsystems are active.

---

**Note**

If you receive message AESM031A SSCT NOT FOUND or message AESM031A SSVT NOT FOUND, the BCSS is not active. The subsystem must be started before you can connect to it.

The **Current BCSS type** field displays the type and version of the BCSS. The version of the BCSS usually does not correspond to the level of BCF.

2 Type the number corresponding to one of the following options, and press **Enter**:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Access the BCF statistical reports that have been saved in the history data set. For more information, see “Using BCF reports” on page 191.</td>
</tr>
<tr>
<td>2</td>
<td>Access the log volume records that have been saved in the history data set. For more information, see “Using BCF reports” on page 191.</td>
</tr>
<tr>
<td>3</td>
<td>Access the execution control records for active job steps (usually necessary only when manual intervention is required). For more information, see “Using BCF execution control records” on page 151.</td>
</tr>
<tr>
<td>4</td>
<td>Set, display, and change values for BCF processing options at the global level. For more information, see “Using BCF options” on page 61.</td>
</tr>
<tr>
<td>5</td>
<td>Set, display, and change values for BCF processing options at the system (IMSID) level. For more information, see “Using BCF options” on page 61.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>6</td>
<td>Set, display, and change values for BCF processing options at the job step level. For more information, see “Using BCF options” on page 61.</td>
</tr>
<tr>
<td>7</td>
<td>Set, display, and change dynamic allocation information for the primary and secondary log data sets. For more information, see “Preparation for dynamic allocation of log data sets” on page 113</td>
</tr>
<tr>
<td>U</td>
<td>Access the AR/CTL common utilities. For more information about these utilities, see the APPLICATION RESTART CONTROL Administrator Guide.</td>
</tr>
<tr>
<td>X</td>
<td>Exit from the BATCH CONTROL FACILITY primary menu.</td>
</tr>
</tbody>
</table>

### Using the Active BMC Consolidated Subsystems panel

Use this panel to select the BCSS (and REGISET) that you want to work with.

1. Enter ? or blanks in the BCSS identifier (BCSID) field on the BATCH CONTROL FACILITY primary menu.

BCSS displays the Active BMC Consolidated Subsystems panel (Figure 4 on page 49).

---

**Note**

If no BCSS is active, BCF displays a message instead of the Active BMC Consolidated Subsystems panel.

---

For the MVS system where your TSO session is running, the panel lists all BCSSs that have been started since the last IPL and that are still active. The types PRIVATE and PUBLIC are defined when the subsystem is installed.

The BCSS must be started (active) before you can connect to it. For more information about starting the BCSS, see the APPLICATION RESTART CONTROL Administrator Guide.

---

**Figure 4: Active BMC Consolidated Subsystems panel**

<table>
<thead>
<tr>
<th>Command ===&gt;</th>
<th>Scroll ===&gt; PAGE</th>
</tr>
</thead>
</table>

Select one BCSID. Then press enter.

<table>
<thead>
<tr>
<th>S</th>
<th>BCSID</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AESQ</td>
<td>PRIVATE</td>
<td></td>
</tr>
<tr>
<td>ARCQ</td>
<td>PRIVATE</td>
<td></td>
</tr>
<tr>
<td>ARDO</td>
<td>PRIVATE</td>
<td></td>
</tr>
<tr>
<td>BCFQ</td>
<td>PRIVATE</td>
<td></td>
</tr>
<tr>
<td>MRFP</td>
<td>PRIVATE</td>
<td></td>
</tr>
</tbody>
</table>

****************************************************************************** BOTTOM OF DATA ******************************************************************************

---

Chapter 2  Getting started  49
Limiting the list of records

When you select a type of record from the BATCH CONTROL FACILITY primary menu or from another BCF panel, BCF might display a panel to allow you to limit the list of record names BCF displays for selection.

Figure 5 on page 50 is an example of this type of panel. If the REGISET contains numerous records of this type, a limited list helps prevent excessive scrolling when selecting records. A limited list can also help reduce confusion by omitting records that do not apply to the parameters you enter, such as the job name, jobstep name, PSB name, and ddname.

Figure 5: Limit List of Records panel

AESPIMVM                     Limit List of Records
Command ===> _________________________________________________________________
BCSID   : BCFQ      Record type : IMS environment registration

To limit the list of records, type positional masking characters
and wildcard characters (*). Then press Enter

Record ID
    IMSID  . . . . . .  ****
    MVSID  . . . . . .  ****

To limit the list of records

1 Enter a specific name or an asterisk in any field to tailor the list.

When you press Enter, BCF displays the list of records of the selected type that match the selection criteria; it omits those that do not match. If no records match the criteria, the list is empty, and you can add new records.

Listing records

After you select a BCF record type or limit the list of records, BCF displays a panel listing the records of the selected type that match your limiting qualifiers (if any).
See Figure 6 on page 51 for an example of this type of panel. From this panel, you can add a record, delete a record, or select a record to view or update.

**Figure 6: List Records panel**

<table>
<thead>
<tr>
<th>Action code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>Display the contents of this record (and possibly make changes).</td>
</tr>
<tr>
<td>D</td>
<td>Delete the record. BCF displays a panel to request confirmation of the deletion; for more information, see “Deleting a record” on page 52.</td>
</tr>
</tbody>
</table>

**To create a new record**

1. Enter the ADD command.

BCF displays a panel to request the qualifiers for the record key; for more information, see “Adding a record” on page 51.

You can use the following action codes:

**Adding a record**

Use this procedure to add a record.

1. Enter the ADD command from a record selection list.
BCF displays a panel to request the record qualifiers; see Figure 7 on page 52 for an example of this type of panel.

**Figure 7: Add Record panel**

<table>
<thead>
<tr>
<th>Command ===&gt;</th>
<th>Add Record</th>
</tr>
</thead>
</table>

**BCSID** : BCFQ  
**Record type** : IMS program registration

Type positional masking characters and wildcard characters (*) in key fields. Then press Enter.

<table>
<thead>
<tr>
<th>Record ID</th>
<th>PROGNAME</th>
<th>PROCSTEP</th>
<th>JOBSTEP</th>
<th>JOBNAME</th>
<th>PSBNNAME</th>
<th>IMSID</th>
<th>MVSID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>. . . . .</td>
<td>. . . . .</td>
<td>. . . . .</td>
<td>. . . . .</td>
<td>. . . . .</td>
<td>. . .</td>
<td>. . .</td>
</tr>
</tbody>
</table>

2. Enter specific and generic qualifiers in any combination in the record key.

For information about using wildcard characters in the qualifiers, see “Wildcard characters in record keys” on page 44.

3. Press Enter.

**Deleting a record**

Use this procedure to delete a record.

1. Select a record for deletion.

BCF displays a panel so that you can confirm the delete request; see Figure 8 on page 53 for an example of this type of panel. To help you identify the record completely, the panel displays the last update (or creation) information for the record.
Note

Most valid ISPF commands are available on this panel; however, because you should give careful consideration to the delete request, the END command is not available. The panel does not provide a default value; you must choose an option before continuing.

Figure 8: Delete Record panel

AESPIMPD Delete Record
Command ===> _________________________________________________________________

BCSID : BCFQ       Record type : IMS program registration
Rec ID  : PAGTEST*  ********  ********  PAGJOB**  ********  PAG1      ****

Select an option. Then press Enter.
   _  1. Delete record
   _  2. Cancel delete request

2 Type one of the following options, and press Enter:

■ 1—Delete this record.

   WARNING
   Be careful when using this option. Depending on the record and the record key, deleting the record can affect multiple application programs in the environment. Be sure that the record is not needed before deleting it.

■ 2—Cancel the delete request.

Implementing BCF in application programs

To provide access to the AR/CTL and BCF load modules during execution of an application program, you must create an IMS environment registration record. To cause BCF to participate in program execution, you must create an IMS program registration record or include the AES$ssid DD statement in the JCL for executing the application program.

Working with IMS environment registration records

An IMS environment registration record provides access to the AR/CTL and BCF load modules in the environment.
This record contains the data set names of the AR/CTL and BCF load module libraries.

If you provide an IMS program registration record (discussed in the next section), the libraries from the IMS environment registration record are used instead of any AR/CTL or BCF libraries specified in the STEPLIB concatenation in the JCL to execute the application program.

**Note**
A global-level IMS environment registration record is created during configuration of AR/CTL. This record should be appropriate for most application executions; typically you do not need to create other IMS environment records.

All of the steps in the following procedure are required. For details about this task, see the APPLICATION RESTART CONTROL Administrator Guide.

**To create an IMS environment registration record**

1. Access the Application Enhancement Series primary menu (see “Using the Application Enhancement Series primary menu” on page 46).

2. Type 1 (AES records) in the choice entry field.

3. Press Enter.

   AR/CTL displays the Application Enhancement Series Options panel.

4. Type or verify the BCSID.

5. Type 5 (IMS environment registration) in the choice entry field.

6. Press Enter.

   AR/CTL displays the Limit List of Records panel, which you can use to limit the list of records (see “Limiting the list of records” on page 50).

7. Type or verify the qualifiers for limiting the list of records. You can use specific and wildcard characters as explained in “Wildcard characters in record keys” on page 44.

8. Press Enter.

   AR/CTL displays the List Records panel, which shows a list of records matching your selection criteria. (See “Listing records” on page 50.)

9. On the List Records panel, enter the ADD command.
10 On the Add Record panel, type the qualifiers to use in the record key. (See “Adding a record” on page 51.)

11 Type or verify the data set names of the AES execution load module library, the AR/CTL execution load module library, and the BCF component execution load module library.

The AES, AR/CTL, and BCF component execution load module libraries should reside in the same data set. Type the same data set name in each field.

12 Press Enter to validate the information. Enter the END command to save the record and exit from the panel.

**Working with IMS program registration records**

To cause BCF to participate in the execution of an application program, you can create an IMS program registration record.

If you do not create an appropriate IMS program registration record in the REGISET, you must include the AES$ssid DD statement in the JCL for executing the application program.

**Tip**

BMC Software recommends the IMS program registration record method for implementing BCF in most environments.

All of the steps in the following procedure are required. For details about this task, see the APPLICATION RESTART CONTROL Administrator Guide.

**To create an IMS program registration record**

1 Access the Application Enhancement Series primary menu (see “Using the Application Enhancement Series primary menu” on page 46).

2 Type 1 (AES records) in the choice entry field.

3 Press Enter.

AR/CTL displays the Application Enhancement Series Options panel.

4 Type or verify the BCSID.

5 Type 6 (IMS environment registration) in the choice entry field.
6 Press Enter. AR/CTL displays a panel that you can use to limit the list of records (see “Limiting the list of records” on page 50).

7 Type or verify the qualifiers for limiting the list of records. You can use specific and wildcard characters as explained in “Wildcard characters in record keys” on page 44.

8 Press Enter. AR/CTL displays a panel that shows a list of records matching your selection criteria. (See “Listing records” on page 50.)

9 Enter the ADD command to create a new record.

10 On the Add Record panel (see “Adding a record” on page 51) type or verify the qualifiers to use in the record ID.

11 Make sure that the default N is set for the Replace IMS option; this value is required for BCF executions. The Replace IMS option is for use with AR/CTL; for more information, see the APPLICATION RESTART CONTROL Reference Manual.

12 Press Enter to validate the information. Enter the END command to save the record and exit from the panel.

**Modification of JCL**

This section explains the AES$ssid DD statement and other JCL changes that you can make for BCF.

To implement BCF without using a program registration record, you must modify the application program execution JCL or JCL procedure to include the AES$ssid DD statement. This statement identifies the BCSS to use for the job step. Whether you use a program registration record or the AES$ssid DD statement, you can make other JCL changes for BCF data sets.

**BCF modifications**

The example in this topic shows JCL for executing an IMS batch application program with some possible BCF modifications (pointed out in the comments).

*All changes are optional; however, if you do not create a program registration record that applies to the application program execution, the AES$ssid DD statement is required.*
<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES$ssid DD</td>
<td>You can insert a DD DUMMY statement that identifies the subsystem ID of the BCSS to use for the job step. The ddname of the statement is <strong>AES$ssid</strong>, where <strong>AES$</strong> is a constant and <strong>ssid</strong> is the subsystem ID of the BCSS. The AES$ssid DD statement works as follows:</td>
</tr>
<tr>
<td></td>
<td>- If an AES$ssid DD statement is present and the REGISET does not contain a program registration record, AR/CTL is activated in the execution even if a program exclusion record is present.</td>
</tr>
<tr>
<td></td>
<td>- If an AES$ssid DD statement is present, a program registration record is present, and a program exclusion record is present, the exclusion record takes effect and AR/CTL is deactivated.</td>
</tr>
<tr>
<td></td>
<td>- If no AES$ssid DD statement is present, BCF uses the subsystem associated with a REGISET that contains an applicable IMS program registration record.</td>
</tr>
<tr>
<td>STEPLIB DD</td>
<td>The IMS environment registration record controls which libraries are used during execution.</td>
</tr>
<tr>
<td>BCFSTATS DD</td>
<td>In addition to writing statistics to the history data set, BCF can write them to the output data set defined by the BCFSTATS DD statement. If you provide the BCFSTATS data set, you can define it as <code>SYSOUT=*</code> to send the output to the same destination as the job messages, or you can define it as a DASD data set. Do not code this statement as <strong>DUMMY</strong>. For more information about the BCF reports, see “Using BCF reports” on page 191. If you omit the BCFSTATS DD statement, BCF writes the statistical reports to the history data set only.</td>
</tr>
<tr>
<td>BCFPRINT DD</td>
<td>BCF uses the BCFPRINT data set to report the option values used for execution, the dynamic allocation text unit information from the execution, and errors detected in the BCFSYSIN control statements. BCF dynamically allocates this data set as <code>SYSOUT=*</code> (sending the output to the same destination as the job messages). You can include the BCFPRINT DD statement to write the output to a DASD data set. The statement should define a sequential data set.</td>
</tr>
<tr>
<td>BCFSYSIN DD</td>
<td>You can code the BCFSYSIN DD statement to describe the BCF control statement data set. The control statements define BCF processing options to merge with the options specified at the global, system, and/or job step level. The BCFSYSIN data set can be a sequential data set or a member of a PDS, or it can be specified instream. The record length must be defined as 80 bytes. For more information about specifying the control statements in the BCFSYSIN data set, see “Working with options at the execution (BCFSYSIN) level” on page 109.</td>
</tr>
<tr>
<td>BCFTRACE DD and BCFBOTRC DD</td>
<td>“Contacting BMC Software product support” on page 253 provides descriptions of the BCFTRACE DD and BCFBOTRC DD statements as well as other information on resolving problems.</td>
</tr>
<tr>
<td>BCFHIST DD</td>
<td>You can include the BCFHIST DD statement to override the default history data set name value entered on the Processing Options panel (see “History Data Set Name” on page 80). This statement provides an easy way to change the name of the history data set without changing the default.</td>
</tr>
</tbody>
</table>
At times, it may be necessary to disable the Batch Backout Assist feature for problem determination or for performance comparisons. You can disable the feature by including the BCFBAIGN DD statement in the job step JCL. Specify the statement as DUMMY.

**BCF use of special DD names**

The following ddnames are used by BCF for special functions:
Excluding BCF from program execution

BCF provides several methods for excluding job steps from BCF participation. You can use these methods to phase in use of BCF after installation, to prevent BCF from executing in selected job steps, or to switch off BCF in emergency situations.

The following methods are available:

- IMS program exclusion record
- BCF, Active Option (see “How BCF merges option values” on page 62)

The Force BCF Execution option (set at the AR/CTL level) is used only if BCF encounters an initialization failure. It has no effect on intentional methods for deactivating BCF.

Working with IMS program exclusion records

To exclude BCF from execution, create an IMS program exclusion record for the program.

This record excludes all AR/CTL products from participation—not just the BCF component, but also AR/CTL for IMS, AR/CTL for DB2, and AR/CTL for VSAM.

All of the steps in the following procedure are required. For details about this task, see the APPLICATION RESTART CONTROL Administrator Guide.

<table>
<thead>
<tr>
<th>ddname</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBOSYSIN</td>
<td>This temporary data is used during the execution of the IMS Batch Backout utility. The data control block (DCB) name for SYSIN is changed dynamically within the batch backout utility so it uses BBOSYSIN instead of SYSIN. This preserves the original SYSIN data set, which may have been specified in the job step JCL for use by the application.</td>
</tr>
<tr>
<td>BCFRDER</td>
<td>To avoid potential conflicts with the IEFRDER ddname, BCF uses the BCFRDER ddname to alter the ddnames used for log recovery and batch backout utilities when they are invoked during abnormal termination or restart processing. During log recovery, BCF dynamically allocates the input log data set to BCFRDER. During restart processing, BCF dynamically allocates the output log data set for batch backout to BCFRDER.</td>
</tr>
<tr>
<td>BCFRDER2</td>
<td>To avoid potential conflicts with the IEFRDER2 ddname, BCF uses the BCFRDER2 ddname to alter the ddnames used for log recovery and batch backout utilities when they are invoked during abnormal termination or restart processing. During log recovery, BCF dynamically allocates the input log data set to BCFRDER2. During restart processing, BCF dynamically allocates the output log data set for batch backout to this special ddname.</td>
</tr>
</tbody>
</table>
To create an IMS program exclusion record

1. Access the Application Enhancement Series primary menu (see “Using the Application Enhancement Series primary menu” on page 46).

2. Type 1 (AES records) in the choice entry field.

3. Press Enter. AR/CTL displays the Application Enhancement Series Options panel.

4. Type or verify the BCSID.

5. Type 7 (IMS Program exclusion) in the choice entry field.

6. Press Enter. AR/CTL displays a panel that you can use to limit the list of records (see “Limiting the list of records” on page 50).

7. Type or verify the qualifiers for limiting the list of records. You can use specific and wildcard characters as explained in “Wildcard characters in record keys” on page 44.

8. Press Enter. AR/CTL displays a panel (see “Listing records” on page 50) that shows a list of records matching your selection criteria.

9. Enter the ADD command to create a new record.

10. On the Add Record panel (see “Adding a record” on page 51), type or verify the qualifiers to use in the record ID.

11. Press Enter to validate the information, and enter the END command to save the record and exit from the panel.
Using BCF options

The BMC Software BATCH CONTROL FACILITY (BCF) component of APPLICATION RESTART CONTROL (AR/CTL) allows you to control many of the actions it takes during job step processing.

By setting the appropriate values for the BCF component processing options, you can tailor BCF processing to meet the needs of the installation, the system, the application program job step, or the execution of the job step.

Overview of the options

This section defines the option levels, describes how BCF merges the option values at execution time, and discusses how to choose the optimum level or levels at which to set the option values. This section also provides overview information about BCF processing options.

Option levels

You can specify BCF processing options at four levels:

- Global
- System (IMSID)
- Job step
- Execution

Global level

Option values set at the global level apply to all executions of all batch application programs using BCF.

Global-level values are stored in the global processing options record in the REGISET. Each REGISET can contain only one global record, and this record cannot
be deleted. This record contains values for all BCF options. Options set at the other levels override the global-level options.

**System level**

Option values set at the system level override the global values. They apply to all executions of all application programs in a particular IMS system, as identified by the IMS system ID.

System-level values are stored in a system options record of the REGISER. You can create multiple system options records. Each record contains values for only the options you choose to specify for the IMS system, rather than values for all BCF options. The values you do not specify are taken from the global processing options record.

**Job step level**

Option values set at the job step level override system and global values. They apply to all executions of a particular job step.

The job step is identified by a unique combination of job name, step name, and program specification block (PSB) or program name. Job step values are stored in a job step options record of the REGISER. You can create multiple job step options records. Each record contains values for only the options you choose to specify for the uniquely qualified job step. The values you do not specify are taken from the global or system processing options record.

**Execution level**

Option values set at the execution level override job step, system, and global values. They apply to a single execution of a specific job step.

You set execution values in the job step JCL by coding keywords in the control statement data set (BCFSYSIN). The values for any options that you do not specify are taken from the global, system, or job step options record.

**How BCF merges option values**

At execution, BCF merges option values as shown in the figure below.
By merging the options, BCF lets you easily change values at the global and system levels without having to modify every existing job step options record.

**Figure 10: Merging the option values at execution time**

BCF merges option values from all applicable sources into a copy of the BECDIR module in main memory as follows; the original BECDIR module in the BCF execution library is not changed during the merge process:

1. BCF loads the BECDIR directory module from the BCF execution library into main memory. Then BCF reads the option values from the global options record in the REGISET and loads these values into the BECDIR copy in memory.

2. BCF searches the REGISET for a system options record associated with the IMS system in which the application program is executing. The record is associated with the IMS system if the record qualifier contains the specific IMS ID or generic (wildcard) characters that match the IMS ID. If it finds the system options record, BCF replaces the global values in the loaded copy of the BECDIR module with the
values specified in the system options record. Any values that are not specified in the system options record remain as the values set in the global options record.

If BCF does not find an applicable system options record, it does not update any option values set at the global level.

3 BCF searches the REGISET for a record associated with the job step, as uniquely qualified by the job name, step name, and PSB or program name. The record is associated with a job step if the record qualifiers contain the specific job name, step name, and PSB or program name of the job step or generic (wildcard) characters that match these qualifiers. If it finds the job step record, BCF replaces the values in the loaded copy of the BECDIR module with the values specified in the job step options record. Any values that are not specified in the job step options record remain as the values set in the global options record or system options record.

If BCF does not find an applicable job step options record, it does not update any option values set at previous levels.

4 BCF searches the job step JCL for the presence of the BCFSYSIN DD statement, which defines BCF control statement data set. If it finds this statement, BCF replaces the values in the loaded copy of the BECDIR module with the keyword values coded in the BCFSYSIN data set. Any values that are not specified with keywords remain as the values set in the BECDIR module, system options record, or job step options record.

If BCF does not find a BCFSYSIN DD statement, it does not update any option values set at previous levels.

Selecting the appropriate option level

BCF provides multiple option levels to give you maximum control over processing for a particular job while requiring minimum effort to control most jobs. Using a combination of the levels, you can quickly and easily set and change the option values; for example you can use the levels as follows:

- Set the option values you want to use for the majority of the job steps at your installation in the global processing options record. If you later decide to change these values, you can make those changes most easily at the global level. Some BCF option values can be set only at the global level.

- To use a different value for an option in a particular IMS system (such as a test system), set it in a system options record. You can also set a value at the system level to prevent it from changing if you reset that value at the global level.

- If you always want to use a different option value for a job step, set it in a job step options record. You can also set a value at the job step level to prevent that value from changing if you reset it at the global or system level.
For a particular execution of a job step, you can set a different value for an option in the BCFSYSIN data set. You may use the execution level to temporarily bypass a problem or to see how changing the option value affects processing.

Methods for setting options

To set option values at the global level, system level, and job step level, use the BCF ISPF interface. The option you select on the BATCH CONTROL FACILITY primary menu determines the level at which you access the processing options.

The Processing Options panel, on which you actually set the processing option values, is the same for all levels. The title on the Processing Options panel indicates the level you have accessed, and the identifier field to the left of each option field indicates the level at which an option value is set.

To set options at the execution level, use keywords in the BCFSYSIN control statement data set in the job step JCL.

Option descriptions

In this section, each option is described in a separate topic that is titled with the option name.

The opening paragraphs of each section discuss the following subjects (if applicable):

- Meanings of option values
- How the option affects BCF processing and the environment
- How other BCF processing options can affect the use of this option
- Suggestions for using the option in certain environments

Most options can be set with the BCF ISPF interface or with keywords in the BCFSYSIN control statement data set. For reference purposes, each section also contains the following information:

Panel field

field name

On the Processing Options panel, you use this field to set the option value.

Keyword

KEYWORD=value
You can code this keyword in the BCFSYSIN control statement data set. For options that you can set only at levels higher than the execution level, this notation is omitted. Some BCFSYSIN keywords can be abbreviated; for example, you can code RECORD=value or REC=value.

Values

**VALUE**

This brief explanation of the accepted values describes the results of setting a value. Use the value for the panel and the keyword.

**WARNING**

Avoid using *NONE* as a keyword value unless you are certain of the results its use will produce. BCF interprets *NONE* in a restricted manner and takes special actions.

Default

**VALUE**

The initial default value for the option is used when the global options record is created during configuration. You can change this initial default value to set a different global default for your site.

**Summary of the options**

The following table summarizes the information about BCF processing options. The table is arranged alphabetically by the option names listed in the ISPF panel fields.

**Table 6: Options summary**

<table>
<thead>
<tr>
<th>Option name and reference</th>
<th>BCFSYSIN keyword</th>
<th>Valid values</th>
<th>Initial default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Alert User ID” on page 102</td>
<td>ALERT</td>
<td>TSO user ID or NONE</td>
<td>no default</td>
</tr>
<tr>
<td>“Automatic Batch Backout” on page 84</td>
<td>AUTObble</td>
<td>Y, N</td>
<td>Y</td>
</tr>
<tr>
<td>“BATCH CONTROL FACILITY Active” on page 73</td>
<td>BCFactive</td>
<td>Y, N, C (not available in BCFSYSIN)</td>
<td>Y</td>
</tr>
<tr>
<td>“Backout Variable (BKO)” on page 104</td>
<td>BKO</td>
<td>xxxxxxxxx</td>
<td>BACKOUT</td>
</tr>
<tr>
<td>Option name and reference</td>
<td>BCFSYSIN keyword</td>
<td>Valid values</td>
<td>Initial default value</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>------------------</td>
<td>--------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>“Buffered Device Support” on page 86</td>
<td>BUFDEV</td>
<td>BUFFER TWI</td>
<td>BUFFER</td>
</tr>
<tr>
<td>“Buffered Tape Unit Name” on page 78</td>
<td>BUFTAPE</td>
<td>unittype</td>
<td>3400</td>
</tr>
<tr>
<td>“DASD Unit Name” on page 76</td>
<td>DASD</td>
<td>unittype</td>
<td>3380</td>
</tr>
<tr>
<td>“Delete Volume Records” on page 82</td>
<td>DELVOL</td>
<td>N Y</td>
<td>N</td>
</tr>
<tr>
<td>“Department Variable (DEPT)” on page 105</td>
<td>DEPT</td>
<td>xxxxxxxxxx DEPT</td>
<td></td>
</tr>
<tr>
<td>“Discard Logging Active” on page 88</td>
<td>DISLOG</td>
<td>N Y</td>
<td>N</td>
</tr>
<tr>
<td>“DL/I Activity Statistics” on page 98</td>
<td>DLSTAT</td>
<td>N IWAIT FULL</td>
<td>N</td>
</tr>
<tr>
<td>“Dynamic Log Option” on page 93</td>
<td>DYNAMic</td>
<td>N nnnnn</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1-32767)</td>
<td></td>
</tr>
<tr>
<td>“Dynamic Log Type” on page 95</td>
<td>DTYPE</td>
<td>MEMORY DSPACE</td>
<td>MEMORY</td>
</tr>
<tr>
<td>“Exchange Log Device Type” on page 91</td>
<td>EXCHange</td>
<td>N Y STATS MATCH UPDATE DBRC</td>
<td>N</td>
</tr>
<tr>
<td>“History Data Set Name” on page 80</td>
<td>none</td>
<td>data set name</td>
<td>none</td>
</tr>
<tr>
<td>“Ignore Checkpoints” on page 107</td>
<td>IGNorckp</td>
<td>N Y Log</td>
<td>N</td>
</tr>
<tr>
<td>“JOBLOG Messages” on page 79</td>
<td>JOBLOG</td>
<td>FULL LIMITED</td>
<td>FULL</td>
</tr>
<tr>
<td>“Logger Perf % Threshold” on page 101</td>
<td>PERF</td>
<td>nnn</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0-99)</td>
<td></td>
</tr>
<tr>
<td>“Manual Intervention Override” on page 106</td>
<td>MANintov</td>
<td>N Y</td>
<td>N</td>
</tr>
<tr>
<td>“Max SLDS Block Count” on page 97</td>
<td>MSLDS</td>
<td>nnnnn</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0-65535)</td>
<td></td>
</tr>
<tr>
<td>Option name and reference</td>
<td>BCFSYSIN keyword</td>
<td>Valid values</td>
<td>Initial default value</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------------</td>
<td>-----------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>“Notify Option for DBRC” on page 90</td>
<td>NOTIFY</td>
<td>BACKOUT, RECOVER, NONE</td>
<td>BACKOUT</td>
</tr>
<tr>
<td>“Number of Statistics Sets” on page 100</td>
<td>STATS</td>
<td>nnn (0-999)</td>
<td>0</td>
</tr>
<tr>
<td>“Physical Logger Error Option” on page 81</td>
<td>LERR</td>
<td>IMS, ABEND</td>
<td>IMS</td>
</tr>
<tr>
<td>“Qualify with Program” on page 83</td>
<td>none</td>
<td>N, Y</td>
<td>N</td>
</tr>
<tr>
<td>“Record Activity Stats To” on page 100</td>
<td>RECord</td>
<td>N, HIST</td>
<td>N</td>
</tr>
<tr>
<td>“Standard Tape Unit Name” on page 77</td>
<td>TAPE</td>
<td>unittype</td>
<td>3400</td>
</tr>
<tr>
<td>“SYSOUT Class” on page 96</td>
<td>SYSOUT</td>
<td>xxxx (A-Z, 1-9, *)</td>
<td>*</td>
</tr>
<tr>
<td>“Tape BKO” on page 96</td>
<td>TBKO</td>
<td>N, Y</td>
<td>N</td>
</tr>
<tr>
<td>“Use Abend Code Series” on page 108</td>
<td>ABNcode</td>
<td>OLD, NEW</td>
<td>OLD</td>
</tr>
<tr>
<td>“Use Execution Controller” on page 74</td>
<td>BECactive</td>
<td>Y, N</td>
<td>Y</td>
</tr>
<tr>
<td>“Use Log Compaction” on page 105</td>
<td>COMP</td>
<td>Y, N</td>
<td>Y</td>
</tr>
<tr>
<td>“Use Physical Logger” on page 75</td>
<td>BLPactive</td>
<td>Y, N</td>
<td>Y</td>
</tr>
</tbody>
</table>

Working with processing option records in ISPF

This section describes how to access processing option records with the BCF ISPF interface and how to use the Processing Options panel.
Accessing global processing option records

To access processing option records at the global level through the BCF ISPF interface, perform the following steps:

1. Access the BATCH CONTROL FACILITY primary menu.
2. Type 4 (Set GLOBAL default options) in the choice entry field.
3. Press Enter. BCF displays the first page of the Processing Options panel. See “Processing Options panel” on page 70.

Only one global processing options record is allowed in the REGISET. This record cannot be deleted.

Accessing system processing option records

To access processing option records at the system (IMS) level through the BCF ISPF interface, perform the following steps:

1. Access the BATCH CONTROL FACILITY primary menu.
2. Type 5 (Set SYSTEM level default options) in the choice entry field.
3. Press Enter. BCF displays a panel listing the system processing options records in the REGISET. (For generic information about this panel, see “Listing records” on page 50.) Select a record to display (and possibly change), delete an existing record, or use the ADD command to create a new record.
4. Press Enter.

- If you selected a record to display, BCF displays the first page of the Processing Options panel. See “Processing Options panel” on page 70.

- If you entered the ADD command, BCF displays a panel to allow you to enter the record qualifiers. (For generic information about this panel, see “Adding a record” on page 51.) Type the record key values to use for the new record, and press Enter. BCF displays the first page of the Processing Options panel.

- If you selected a record to delete, BCF displays a panel so that you can confirm or cancel the delete action. (For generic information about this panel, see “Deleting a record” on page 52.)
Accessing job step processing option records

To access processing option records at the job step level through the BCF ISPF interface, perform the following steps:

1. Access the BATCH CONTROL FACILITY primary menu.
2. Type 6 (Set JOBSTEP level default options) in the choice entry field.
3. Press Enter.

BCF displays a panel you can use to limit the list of records. (For generic information about this panel, see “Limiting the list of records” on page 50.)

4. Type or verify the qualifiers.
5. Press Enter.

BCF displays a panel showing a list of records that match your selection criteria. (For generic information about this panel, see “Listing records” on page 50.) Select an existing record to display (and possibly change), delete an existing record, or use the ADD command to create a new processing option record.

6. Press Enter.

- If you selected a record to display, BCF displays the first page of the Processing Options panel. (See the figure in “Options on page 1 of the Processing Options panel” on page 72. See also “Processing Options panel” on page 70.)

- If you entered the ADD command, BCF displays a panel to allow you to enter the record qualifiers. (For generic information about this panel, see “Adding a record” on page 51.) Type the record key values to use for the new record, and press Enter. BCF displays the first page of the Processing Options panel (see the figure in “Options on page 1 of the Processing Options panel” on page 72).

- If you selected a record to delete, BCF displays a panel so that you can confirm or cancel the delete action. (For generic information about this panel, see “Deleting a record” on page 52.)

Processing Options panel

When you select a processing options record, BCF displays the first page of the Processing Options panel at the level you selected (GLOBAL, SYSTEM, or JOBSTEP).
The Processing Options panel consists of four pages. The general way you interact with the panel is the same for all pages. The panel contains the following elements:

Panel title

The panel title shows the level (GLOBAL, SYSTEM, or JOBSTEP) of options you are working with. The contents and functionality of the panel are the same, regardless of the level.

Command

You can enter the following commands:

- **CAN or CANCEL**
  
  Exit this panel without saving changes or creating the new record. BCF loses any changes that you made on the panel and redisplays the Select (level) Options List panel.

- **SAVE**
  
  Save the changes made to the record or create the record, and display the panel defined in the Next Page field.

- **END**
  
  Save the changes made to the member or create the record, and return to the Select (level) Options List panel. This means you must type **END** on the Command line and press **Enter**, or press an equivalent function key (F3 in most cases).

Page n of 4 Next Page

The **Page n of 4** field identifies the sequence number of the current page. The **Next Page** field contains the sequence number of the page BCF displays if you press **Enter**. The initial value in this field is the default next page. You can type over the displayed value to go directly to a particular panel.

Identifier

The identifier field to the left of the option name indicates the level at which the displayed option value is defined. The following identifiers are possible:

- **G**—global level
- **S**—system level
- **J**—job step level

You cannot modify the values in this field. If you change the value of an option, the BCF ISPF interface replaces the value in this field with G, S, or J, depending on the level at which you are defining options. The level is indicated by the panel title.
Option name

The option name identifies BCF processing option that is controlled by the value you specify in the option value field.

Option value

To set a value for the option, type the desired value in the field. If the field already contains a value, type over the displayed value.

To reset an option value that is set at the job step level to the value set at the system or global level, erase the displayed value (with the Erase EOF key) or type spaces in the field. When you press Enter, BCF resets the field to the system value if it is defined, or to the global value if a system value is not defined.

To reset an option value that is set at the system level to the value set at the global level, erase the displayed value (with the Erase EOF key) or type spaces in the field. When you press Enter, BCF resets the field to the global value.

If a field is blank at the global level, BCF requires you to type a valid value before proceeding to the next panel.

If you need further information about any option, type a question mark (?) in the field and press Enter. BCF displays a help panel to explain the option and the values you can specify. You can enter the desired value directly on the help panel.

When you change any value on the SYSTEM or JOBSTEP Processing Options panel or on a supporting help panel, BCF highlights the value and displays the following message to notify you of the change:

BCFM032A VALUE CHANGED, PRESS ENTER TO CONTINUE

Press Enter to display the page identified in the Next Page field.

Options on page 1 of the Processing Options panel

This section describes how to set values for the processing options on page 1 of the Processing Options panel (the figure below).
### BATCH CONTROL FACILITY Active

Use this option to activate or deactivate overall BCF processing in the environment. You can use it in an emergency if execution problems make it advisable to disable BCF.

Setting this value to **Y** at the global level indicates that you want the initial status of BCF to be active. You can still bypass the use of BCF by specifying this value as **N** at a lower level.

When you specify a value of **N** at the global level, BCF does not look at additional option levels for overrides. The execution controller and BCF Physical Logger are not used. IMS native routines are used instead.

If you specify the value **C**, the execution controller merges all available options and the status of BCF is determined by the final setting of this option. If, after the merge, the option value remains **C**, the execution controller and BCF Physical Logger are not used. IMS native routines are used instead.

The option values **Y** and **C** allow for conditional activation of BCF in environments where it is desirable for BCF to be installed in a common library and its activation staged by system (IMSID) or job step. If you want BCF to be active for all jobs unless specifically deactivated, set the option value to **Y** at the global level. Then specify the option value as **N** for any system or job step option members where BCF is not desired. If you want BCF to be inactive for all jobs unless specifically activated, set the value of this option at the global as **C**. Then specify the value **Y** for any system or job step level option member where BCF is desired.

This option lets you suspend and resume BCF usage without making JCL changes; however, the execution controller and the BCF Physical Logger perform some initialization processing before interrogating this option and determining its final value. More action may be required to remove BCF completely, such as removing the option values from all levels of option member.
the BCF execution library from the STEPLIB concatenation or deleting the program registration record.

---

**WARNING**
The job step restart procedures in the environment may require the execution controller to perform the batch backout process. If you reset this value from Y to N, you must ensure that the batch backout process is performed manually if required.

---

Panel field

**BATCH CONTROL FACILITY Active**

Keyword

`BCFactive=value`

Values

Set one of the following values:

- **Y**—Activate BCF
- **N**—Do *not* activate BCF. The execution controller does not initialize. The native IMS Physical Logger is used instead of the BCF Physical Logger.
- **C**—Determine the active status of BCF after the execution controller has merged all available options. If, after the merge, the value for this option is C, IMS native routines are used instead of the execution controller and BCF Physical Logger. This value is not available when you use the BCFSYSIN keyword.

Default

**Y**

---

**Use Execution Controller**

Use this option to activate or deactivate the execution controller in the environment. You can use it as an execution switch in an emergency if execution problems make it advisable to disable the execution controller.

If the value of the **BATCH CONTROL FACILITY Active** option is **N** at the global level, BCF does not check this option and control passes to the IMS region controller.

This option applies to the execution controller *only*. You can still use the BCF Physical Logger if this option is set to **N**.
This option lets you suspend and resume usage of the execution controller without making JCL changes; however, the execution controller performs some initialization processing before interrogating this option. More action may be required to remove the execution controller completely, such as removing the BCF execution library from the STEPLIB concatenation or deleting the program registration record.

**WARNING**
The job step restart procedures in the environment may require the execution controller to perform the batch backout process. If you reset this value from Y to N, you must ensure that the batch backout process is performed manually if required.

**Panel field**

Use Execution Controller

**Keyword**

BECactive=value

**Values**

Set one of the following values:

- **Y**—Activate the execution controller. The BCF Physical Logger must also be active. If an error occurs during the initialization of the execution controller, the action taken usually depends on the value of the Initialization Error Option, which is set at the AR/CTL global level.

- **N**—Do not activate the execution controller. Use the standard IMS execution environment. Some functions of the BCF Physical Logger may be unavailable if the execution controller is not active.

**Default**

Y

**Use Physical Logger**

Use this option to activate or deactivate the BCF Physical Logger in the environment. You can use it in an emergency if execution problems make it advisable to disable the BCF Physical Logger.

If the value of the BATCH CONTROL FACILITY Active option is N at the global level, BCF does not check this option and the IMS Physical Logger handles all logging requests.
If you set this option to **N**, the BCF Physical Logger does not initialize and the IMS Physical Logger services all batch logging requests. Because the execution controller requires the BCF Physical Logger to be active, disabling the BCF Physical Logger also disables the execution controller.

This option lets you suspend and resume usage of the BCF Physical Logger without making JCL changes; however, the Physical Logger performs some initialization processing before interrogating this option. More action may be required to remove the Physical Logger completely, such as removing the BCF execution library from the STEPLIB concatenation or deleting the program registration record.

**WARNING**
The job step restart procedures in the environment may require the execution controller to perform the batch backout process. If you reset the value of this option from **Y** to **N**, you not only disable the BCF Physical Logger, you also disable the execution controller. Therefore, you must ensure that the batch backout process is performed manually if required.

**Panel field**

Use Physical Logger

**Keyword**

BLPactive=value

**Values**

Set one of the following values:

- **Y**—Activate the BCF Physical Logger to replace the IMS Physical Logger for batch logging requests. If initialization of the BCF Physical Logger is not possible, the action taken usually depends on the value of the Physical Logger Error Option. For more information, see “Physical Logger Error Option” on page 81.

- **N**—Use the IMS Physical Logger.

**Default**

**Y**

**DASD Unit Name**

Use this option to provide the unit name BCF should use when it dynamically allocates an existing IMS log data set residing on DASD as input to the IMS Batch Backout utility or IMS Log Recovery utility.
If the Automatic Batch Backout option is set to Y and a job step or system failure occurs, the execution controller invokes the IMS Batch Backout utility (and, if necessary, the IMS Log Recovery utility) to prepare the environment for job step restart. BCF dynamically allocates the log data set that was originally allocated in the job step JCL as input to these utilities. If the log data set is not cataloged, the specific device unit name is not available when BCF dynamically allocates the data set as input to the IMS Batch Backout utility or IMS Log Recovery utility.

This option provides the specific device unit name to use when the log data set is not cataloged and it resides on a DASD device.

BCF uses this option only when the log data set was originally allocated through a job step JCL statement. If BCF dynamically allocated the log data set through the Exchange Log Device Type option, it uses the same unit name to allocate the log for batch backout or log recovery as it used in the original dynamic allocation.

If DBRC is active for the job step, the value of the DASD Unit Name option should be consistent with the DASDUNIT parameter on the INIT.RECON control statement.

**Panel field**

<table>
<thead>
<tr>
<th>DASD Unit Name</th>
</tr>
</thead>
</table>

**Keyword**

DASD=value

**Values**

Specify the unit type (1 to 8 characters) to which DASD log data sets are allocated.

**Default**

3380

**Standard Tape Unit Name**

Use this option to provide the unit name BCF should use when it dynamically allocates an existing IMS log data set that resides on a standard or unbuffered tape device as input to the IMS Batch Backout utility or IMS Log Recovery utility.

If the Automatic Batch Backout option is set to Y and a job step or system failure occurs, the execution controller invokes the IMS Batch Backout utility (and, if necessary, the IMS Log Recovery utility) to prepare the environment for job step restart. BCF dynamically allocates the log data set that was originally allocated in the job step JCL as input to these utilities. If the log data set is not cataloged, the specific
device unit name is not available when BCF dynamically allocates the data set as input to the IMS Batch Backout utility or IMS Log Recovery utility.

This option provides the specific device unit name to use when the log data set is not cataloged and it resides on a standard or unbuffered tape device.

BCF uses this option only when the log data set was originally allocated through a job step JCL statement. If BCF dynamically allocated the log data set through the Exchange Log Device Type option, it uses the same unit name to allocate the log for batch backout or log recovery as it used in the original dynamic allocation.

If DBRC is active for the job step, the value of the Standard Tape Unit Name option should be consistent with the TAPEUNIT parameter on the INIT.RECON control statement.

**Panel field**

**Standard Tape Unit Name**

**Keyword**

TAPE=value

**Values**

Specify the unit type (1 to 8 characters) to which standard tape log data sets are allocated.

**Default**

3400

---

**Buffered Tape Unit Name**

Use this option to provide the unit name BCF should use when it dynamically allocates an existing IMS log data set that resides on a buffered tape device as input to the IMS Batch Backout utility or IMS Log Recovery utility.

If the Automatic Batch Backout option is set to Y and a job step or system failure occurs, the execution controller invokes the IMS Batch Backout utility (and, if necessary, the IMS Log Recovery utility) to prepare the environment for job step restart. BCF dynamically allocates the log data set that was originally allocated in the job step JCL as input to these utilities. If the log data set is not cataloged, the specific device unit name is not available when BCF dynamically allocates the data set as input to the IMS Batch Backout utility or IMS Log Recovery utility.
This option provides the specific device unit name to use when the log data set is not
cataloged and it resides on a buffered tape device.

BCF uses this option only when the log data set was originally allocated through a
job step JCL statement. If BCF dynamically allocated the log data set through the
Exchange Log Device Type option, it uses the same unit name to allocate the log for
batch backout or log recovery as it used in the original dynamic allocation.

Because DBRC does not differentiate between standard and buffered tape units, the
value of the Buffered Tape Unit Name option should be consistent with the
TAPEUNIT parameter on the INIT.RECON control statement if DBRC is active for
the job step.

Panel field
  Buffered Tape Unit Name

Keyword
  BUFFTAPE=value

Values
  Specify the unit type (1 to 8 characters) to which buffered tape log data sets
  are allocated.

Default
  3400

**JOBLOG Messages**

Use this option to limit the types of messages BCF writes to the job log during the
processing of the job step.

BCF always writes error messages; however, you can control whether it writes all or
only product-selected informational messages.

The full range of error and informational messages can help you identify the major
phases and actions that the execution controller has taken during job step execution.
Setting this option to **FULL** is useful for diagnosing problems and for determining
which actions are necessary for manual intervention.

If you do not want BCF to provide this audit trail of messages, set this option to
**LIMITED** to reduce the volume of messages written to the job log data set. All error
messages and product-selected informational messages are still written to the job
log.
Panel field

JOBLOG Messages

Keyword

JOBLOG=value

Values

Set one of the following values:

- FULL—Write all informational and error messages to the job log.
- LIMITED—Write only product-selected informational messages and all error messages to the job log.

Default

FULL

History Data Set Name

This option provides the name of the history data set to use during job step execution.

BCF uses the history data set to store statistical reports about the job step execution and about the primary and secondary log data sets.

Because BCF does not supply a default value for this option, you must provide a value for it. If you have migrated from a previous version of BCF, a migration utility may have already set this value.

If the history data set is not available at execution time, job step processing continues without BCF recording statistics.

Note

You can specify this option at the global level and the system level. At the execution level, you can use the BCFHIST DD statement.

Panel field

History Data Set Name
Values

Specify the name of the history data set. This data set must be an existing VSAM data set. The name can be a maximum of 44 characters, including the periods that separate the qualifiers. For more information about allocating the history data set, see the APPLICATION RESTART CONTROL Customization Guide.

Default

none

Options on page 2 of the Processing Options panel

This section describes how to set values for the processing options on page 2 of the Processing Options panel (see the figure below).

**Figure 12: Processing Options panel (Page 2)**

<table>
<thead>
<tr>
<th>Command</th>
<th>GLOBAL Processing Options</th>
<th>PAGE 2 OF 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Next Page .. 3</td>
</tr>
</tbody>
</table>

To store or change a value in the member, over type or change the value. To remove, blank or erase the value.

- **G Physical Logger Error Option** . . . IMS  (IMS/ABEND)
- **G Delete Volume Records** . . . . . . . N  (N/Y)
- **G Qualify with Program** . . . . . . . N  (N/Y)
- **G Automatic Batch Backout** . . . . . . Y  (Y/N)
- **G Buffered Device Support** . . . . BUFFER  (BUFFER/TWI)
- **G Discard Logging Active** . . . . . . N  (N/Y)
  - If Discard Logging Active is Y:  
    - **G Notify Option for DBRC** . . . . BACKOUT  (BACKOUT/RECOVER/NONE)
    - **G Exchange Log Device Type** . . . . N  (N/Y/STATS/MATCH/UPDATE/DBRC)
    - **G Dynamic Log Option** . . . . . . . N  (N/1-32767)
    - **G Dynamic Log Type** . . . . . . . MEMORY  (MEMORY/DSPACE)

Place a ? in any option for functional description.

Physical Logger Error Option

Use this option to specify the action BCF takes if an error occurs during the initialization of the BCF Physical Logger.

When this option is set to IMS and the BCF Physical Logger cannot initialize, the job step continues with the IMS Physical Logger performing the physical logging. You can use this value if you are implementing BCF in phases and your site does not fully depend on BCF to provide logging enhancements.
When this option is set to ABEND and the BCF Physical Logger cannot initialize, BCF issues an abnormal termination for the job step. You should set this option to ABEND if the job step requires one or more BCF logging enhancements.

Because BCF is designed to let IMS try to continue if a BCF error occurs, the initial default value is IMS. However, if your operational procedures depend on the BCF Physical Logger for performance, dynamic backout support, or discard logging support, BMC Software recommends using the value ABEND.

**Panel field**

Physical Logger Error Option

**Keyword**

LERR=value

**Values**

Set one of the following values:

- IMS—Continue job step processing using the IMS Physical Logger to service logging requests.
- ABEND—Abend the job step.

**Default**

IMS

---

**Delete Volume Records**

Use this option to control whether the execution controller copies the log volume record (or records) to the history data set or deletes them without copying them first.

The execution controller uses log volume records in the REGISET to contain the information about the log data sets that were created during job step execution. If the execution controller determines that batch backout is needed for the job step, it uses the information in the log volume records to allocate the log data sets to use as input to the IMS Batch Backout utility (and, if necessary, the IMS Log Recovery utility). BCF also uses the data in the log volume records to create the Log Data Set Information reports.

Set this option to Y if you want the execution controller to delete the log volume records from the REGISET without copying them to the history data set. Set this option to N if you want the execution controller to delete the log volume records after copying them to the history data set. When the records are in the history data
set, they are available for the Log Data Set Information reports and for input to programs that audit and track log data sets.

When you no longer want to keep the log volume records in the history data set, you can access them through option 2 on the BATCH CONTROL FACILITY primary menu and manually select them for deletion, as explained in “Log volume records” on page 152. Or you can use the BCFXPURG utility, as explained in “BCF Purge utility” on page 186.

**Panel field**

Delete Volume Records

**Keyword**

DELVOL=value

**Values**

Set one of the following values:

- Y—Delete the log volume records from the REGISET without copying them to the history data set first.
- N—Copy the log volume records to the history data set before deleting them from the REGISET.

**Default**

N

**Qualify with Program**

Use this option to control whether BCF uses the program specification block (PSB) name or the program name as a REGISET record key qualifier.

Set this option to the value that provides unique identification in your environment.

BCF must be able to identify each unique batch application job step in the environment. It uses a combination of the job name, step name, and PSB or program name as the job step identifier. The job step identifier is part of the record key in the following types of REGISET and history data set records:

- Job step processing options
- Execution control
- Primary log volume
Secondary log volume
Job step dynamic allocation
Statistics

All members in the REGISET must use a consistent qualification method.

**Note**
You can specify this option only at the global level.

**Panel field**

**Qualify with Program**

**WARNING**
Do not change the value of this option without carefully evaluating the effect of the change on existing records in the REGISET. If the REGISET contains execution control records, job step options records, or job step dynamic allocation records, changing the value of this option can cause unpredictable results and can lead to data integrity exposures.

**Values**
Set one of the following values:

- **N**—Use the PSB name in the job step identifier.
- **Y**—Use the program name in the job step identifier.

**Default**

N

**Automatic Batch Backout**

Use this option to control whether the execution controller automatically invokes the IMS Batch Backout utility and, if necessary, the IMS Log Recovery utility, when needed to restore the environment for job step restart.

If you set this option to **Y**, the execution controller attempts to provide an environment in which the normal procedure for handling job step restart after a program abend or system failure is to resubmit the job step. BCF performs the following processing:

- If a program abend occurs during processing and the application program has made database updates, the execution controller automatically invokes the IMS
Batch Backout utility and, if necessary, the IMS Log Recovery utility, to back out the database changes before job step termination. The job step can be resubmitted whenever the problem that caused the abend is corrected.

- If the execution controller cannot accomplish the necessary backout processing at job step termination, such as when a system failure occurs, the job step can still be resubmitted. The execution controller recognizes that the previous execution of the job step requires backout. The execution controller automatically invokes the IMS Batch Backout utility (and, if necessary, the IMS Log Recovery utility); and it ensures that the batch backout process completes successfully before allowing the application program to execute.

If the execution controller cannot ensure that all necessary batch backout processing has been done, it requests manual intervention. For more information, see “Manual intervention” on page 174.

During the automatic batch backout process, BCF dynamically allocates log data sets to use with the IMS Batch Backout utility (and, if necessary, the IMS Log Recovery utility). To provide the allocation information BCF needs, you must create at least one dynamic allocation record in the REGISET. “Preparation for dynamic allocation of log data sets” on page 113 contains complete information about this topic.

If this option is set to **N**, the execution controller does not attempt to invoke batch backout processing, and it does not attempt to prevent the restart of the job step when the batch backout process has not completed successfully. During restart processing, BCF determines the setting of the Automatic Batch Backout flag from the options used during the original job step execution.

You may set this option to **N** for unique situations where it is valid to restart the job step without performing batch backout processing. The following examples illustrate these situations:

- When the BCF Discard Logging option is set to **Y**, no log records are available to use as input to the IMS Batch Backout utility. The proper way to handle a program abend or system failure during the job step is to recover the database from an image copy data set.

- If the application program does replace-type operations only, it can be valid to re-execute the program without backout.

- If operational procedures for handling the batch backout process are already in place and you want to implement BCF gradually, you can get the benefits of BCF improvements in the batch logging area without immediately changing your operational procedures.

When the Automatic Batch Backout option is set to **N** and a program abend occurs, the execution controller issues message BMC44251I during job step termination. This message notifies you that batch backout may be required. Because this option value
indicates that you want to perform backout processing manually, the execution controller also deletes the execution control record from the REGISET.

When the Automatic Batch Backout option is set to N and a system failure occurs, the execution controller does not issue a message, and it deletes the execution control record from the REGISET during restart.

If DBRC is in use, you must ensure that batch backout is done or that the subsystem record is deleted before attempting job restart—just as you would normally do if BCF were not installed.

Panel field

Automatic Batch Backout

Keyword

AUTObbo=value

Values

Set one of the following values:

- Y—Automatically invoke the IMS Batch Backout utility (and, if necessary, the IMS Log Recovery utility) when needed to accomplish batch backout.

- N—Do not automatically invoke the IMS Batch Backout utility (and, if applicable, the IMS Log Recovery utility), and do not prevent the restart of a job step if the batch backout process has not completed successfully. Perform no automatic restoration of the environment.

Default

Y

Buffered Device Support

Use this option to control whether the BCF Physical Logger writes log records to buffered (3480 and 3490) tape devices in buffered mode or tape write immediate (TWI) mode.

Both modes provide the write ahead log tape (WALT) support required by the IMS Database Manager. However, for most environments, buffered mode provides significant performance improvements over TWI mode.

To allow the BCF Physical Logger to write log records to buffered tape devices in buffered mode, set this option to BUFFER. In buffered mode, the I/O operation
completes when the data has been transferred from main storage to the buffer of the tape control unit. The tape control unit does not necessarily write the transferred data to the tape media at this time but retains the data in a device buffer until multiple buffers can be written to the tape media in a single transfer operation. There are two advantages of buffered mode:

- The application program is not delayed while the log records are written to tape.
- The buffered device can achieve speeds closer to its rated performance.

Through a combination of device monitoring and synchronization techniques, BCF provides full WALT support when writing in buffered mode. BCF can achieve significant performance improvements in batch logging by using buffered tape devices in buffered mode.

To force the BCF Physical Logger to use buffered tape devices in a manner similar to that used by the IMS Physical Logger, set this option to TWI. In TWI mode, the I/O operation does not complete until the tape control unit actually writes the physical block to the tape media. TWI mode provides WALT support by ensuring that the data is recorded on the tape media before the write operation is completed. Depending on the rate of logging requests and the size and number of IMS log buffers, TWI mode can result in application program delays and a logging rate for the device that is far short of the rates achieved in buffered mode.

You should set this option to TWI mainly for the following reasons:

- If a problem occurs with buffered device support, temporarily setting this option to TWI can allow you to use other BCF features while maintaining log integrity.
- You can compare the performance of the job step when using buffered mode with performance when using TWI mode. The detailed statistics provided by BCF can assist you in tuning the job for maximum performance, so you can make a realistic evaluation of the benefit of buffered support for log data sets written to 3480 and 3490 devices.

When this option is set to TWI, BCF issues message BMC44212I to the TSO user ID identified with the Alert User ID option.

Panel field

Buffered Device Support

Keyword

BUFDEV=value

Values

Set one of the following values:
- BUFFER—Use buffered mode to write to buffered tape devices.
- TWI—Use TWI mode.

**Default**

BUFFER

**Discard Logging Active**

Use this option to allow the job step to bypass the writing of log records but still use DBRC to ensure database integrity.

You can also use this option to allow BCF to collect statistics for job steps that do not log database changes.

BCF support for buffered devices can let you implement logging for a job step that must otherwise execute without logging to achieve acceptable performance; however, some job steps may still need to be executed without logging. DBRC requires all job steps that update databases to log the database changes. Without BCF, you must turn off DBRC for a job step that does not log changes; data integrity exposures, therefore, become possible. This option allows DBRC to be active for a job step that does not log database changes.

If you set this option to Y, the BCF Physical Logger discards the log records passed to it by the IMS Logical Logger. To the IMS Database Manager and to DBRC, it appears that a normal log data set is being created. In fact, BCF does not open the log data set and writes no records to it.

If the processing option (PROCOPT) of the program communication block (PCB) indicates update intent, DBRC is active, and the database is registered for a job step using discard logging, BCF issues all notifications to DBRC to ensure data integrity. When the job step makes the first update to a database, BCF notifies DBRC that backout (or recovery) is needed for the database. However, if the PROCOPT specifies L or LS and the Notify Option for DBRC option is set to NONE, BCF does not notify DBRC at the first update and database integrity is your responsibility.

Further DBRC notifications depend on whether the job step completes successfully or fails:

- If the job step completes successfully, BCF notifies DBRC that backout is no longer needed. Because this job step did not create the log data sets that would be necessary for a valid recovery, BCF notifies DBRC that a database reorganization has been performed. The REORG notification prevents an invalid recovery from occurring if the image copy that is taken after the execution is found to be in error. Finally, BCF notifies DBRC that an image copy is needed for the database.
If the job step does not complete successfully, the Notify Option for DBRC option controls how the notification is handled; for more information, see “Notify Option for DBRC” on page 90.

Because no log data set is created, the IMS Batch Backout utility cannot be executed to restore the database if the job step does not complete successfully.

BCF does not attempt to invoke the utility as it usually does if the Automatic Batch Backout option is set to Y. Instead, BCF issues message BMC44251I with reason code 5 to indicate that a potential integrity exposure outside of BCF control has occurred. If DBRC was active for the job step, a GENJCL RECOV or equivalent recovery process can restore the database to the required status.

When all processing has been done to restore the database, you (or an automated program) must notify DBRC that backout is no longer required for the database. If the execution controller is active for job step control, you must also delete the execution control record from the REGISET to allow job step restart.

You must take an image copy of the database before and after executing the job step. Allowing multiple consecutive job steps to access a database that was updated without logging and without an image copy is inconsistent with the use of DBRC. If your site needs to allow multiple consecutive access, do not use DBRC for the job step execution.

If DBRC is not active for the job step or if the database is not registered with DBRC, you must coordinate the recovery and/or image copy actions necessary to ensure data integrity.

BCF does not support discard logging if the IRLM is being used and the PSB PROCOPT is UPDATE. This environment is assumed to be a block level data sharing environment, and discard logging is inappropriate in this case.

## Note

If the BCF Physical Logger fails to initialize because the IERFDER DD statement is specified as DUMMY or NULLFILE, the job step continues processing regardless of the value specified for the Physical Logger Error option.

To use discard logging, the IERFDER DD statement cannot be specified as DUMMY. For job steps that specify DUMMY, it may be possible to reduce JCL changes by using the Exchange Log Device Type option, which substitutes the log data set defined in the JCL with a data set that BCF dynamically allocates. For more information, see “Exchange Log Device Type” on page 91.

**Panel field**

Discard Logging Active
Keyword

DISLOG=value

Values

Set one of the following values:

- N—The BCF Physical Logger writes the log records to the log data sets.
- Y—The BCF Physical Logger discards all log records.

Default

N

Notify Option for DBRC

Use this option to control how BCF notifies DBRC when discard logging is active, the job step terminates abnormally, and the PROCOPT of the PCB indicates update intent.

This option is active only if the Discard Logging Active option is set to Y.

If a job step using discard logging does not complete successfully, the database remains marked with the notification you set with this option. If you set this option to BACKOUT or RECOVER, DBRC prevents access to the database until the required processing to restore the database has been performed. If you set this option to NONE, BCF does not attempt to notify DBRC.

Panel field

Notify Option for DBRC

Keyword

NOTIFY=value

Values

Set one of the following values:

- BACKOUT—You can use this value if the PROCOPT indicates update intent with any of these values: A, I, R, D, L, or LS. BCF notifies DBRC of Backout Needed. This value represents the way that previous levels of BCF handled notification.
- **RECOVER**—You can use this value if the PROCOPT indicates update intent with any of these values: A, I, R, D, L, or LS. BCF notifies DBRC of Recovery Needed.

- **NONE**—You can use this value if the PROCOPT indicates update intent with either of these values: L or LS. BCF attempts no DBRC notification. If this option is set to NONE with other PROCOPT values, BCF automatically changes the value to BACKOUT.

**Default**

BACKOUT

**Exchange Log Device Type**

Use this option to control whether the execution controller deallocates a log data set defined in the job step JCL and dynamically allocates it more appropriately for the BCF options in use.

You can also use this option to implement other BCF options such as the Dynamic Log Type, Discard Logging Active, and Buffered Device Support without modifying existing JCL, as the following examples show:

- Your site may have converted to DASD logging to obtain better performance. Because BCF can write the logs to a buffered tape device in buffered mode, you may be able to improve job step performance by writing the log data sets to a buffered tape device.

- Your site may have converted to DASD logging to support IMS dynamic backout for DL/I pseudo abends. If you use BCF dynamic log, the physical log data set no longer needs to reside on DASD to support dynamic backout.

- To achieve acceptable performance, your site may execute certain job steps without logging database changes. For these job steps, DBRC is turned off and the IERDRER and IERDRER2 DD statements are specified as DUMMY. BCF provides the ability to discard log records while DBRC is active. Setting the Exchange Log Device Type option to Y and the Discard Logging Active option to Y enables BCF to discard the log records.

In each of these examples, you can set this option to Y. The execution controller deallocates all log data sets specified in the job step JCL. Then it dynamically allocates new log data sets using the data set allocation information you have provided in the REGISET. For more information, see “Preparation for dynamic allocation of log data sets” on page 113.

You will probably find this option most useful during the implementation phase of BCF. While you are testing BCF and determining whether to use various processing
options, you can set this option to Y for individual job steps to allow BCF to exchange the log device type. After testing is complete, you will probably want to change the log data set statements in the JCL.

BMC Software recommends that you build a system-level DEFAULT dynamic allocation record for IEFRDER because, in special cases, BCF may need this member, even if you are not using the Exchange Log Device Type option. If you plan to use dual logging, you must also build a system-level DEFAULT dynamic allocation record for IEFRDER2.

Panel field

Exchange Log Device Type

Keyword

EXCHange=value

Values

Set one of the following values:

- **N**—Do not exchange the log data set device type. Leave the log data sets as allocated in the job step JCL.

- **Y**—Exchange the log data set device type.

- **STATS**—Perform exchange only if the input log data sets are specified as DUMMY or NULLFILE. Additional processing takes place when the device exchange occurs for the job step. If the PSB PROCOPT is not update sensitive, the BCF Physical Logger activates the discard logging feature so that execution characteristics or statistics can be obtained for read-only jobs without log data sets actually being created. If discard logging is invoked for the job step, BCF deletes the log data set during termination processing. If the PSB PROCOPT indicates update processing, BCF does not invoke discard logging for the job step and retains the created log data set.

- **MATCH**—Perform exchange based on the device type of the log data set specified in the job step JCL. If a job step dynamic allocation record is present, it is used. If a job step dynamic allocation record is not present, BCF attempts to use a system dynamic allocation record, if present, that matches the device type associated with the log data set specified in the job step JCL. If a system dynamic allocation record specified as DASD, TAPE, or BUFTAPE is present, it is selected based on the device type of the log data set specified in the job step JCL. If the log data set DD statement in the JCL is defined as DUMMY or NULLFILE, BCF uses a default skeleton type of DASD. If the DASD skeleton type is not available, BCF uses the DEFAULT skeleton type.
UPDATE— Perform exchange only if the PSB PROCOPT associated with the execution has UPDATE intent. If the intent of the PROCOPT cannot be determined or the PSB cannot be located, the exchange is performed as if the intent of the PROCOPT is UPDATE.

DBRC— Perform exchange only if the PSB PROCOPT associated with the execution has UPDATE intent and the log data sets specified in the job step JCL are specified as DUMMY or NULLFILE. This exchange option is intended to ensure that update-sensitive jobs do not execute without log data sets. DBRC does not have to be active before the exchange is performed. If the intent of the PROCOPT cannot be determined or the PSB cannot be located, the exchange is performed as if the intent of the PROCOPT is UPDATE.

Default

N

Dynamic Log Option

Use this option to control whether the BCF Physical Logger creates a dynamic log to use with the IMS dynamic backout process.

IMS provides dynamic backout support for DL/I pseudo abends; however, to provide this support, IMS imposes two significant restrictions on the batch logging process:

- The log data set must be written to DASD. Because log data sets are typically retained over time to permit recovery of the database, most installations must archive the DASD log to the less expensive tape media for long-term storage.

- The log data set must be written to emulate fixed length blocks so that IMS can read it with OSAM and convert the log block number to a DASD TTR. The blocks must be padded to the required maximum length at the expense of storage space.

BCF can circumvent these IMS restrictions by creating a separate dynamic log to satisfy dynamic backout read requests. Because the physical log data set is no longer used for the dynamic backout process, the log data set can be written to tape. Also, it is no longer necessary to pad the log blocks to a fixed length; therefore, less space may be required for the log data sets. (The amount of DASD savings, if any, depends on application design, number of database buffers, and logging parameters.)

When you set this option to a value other than N, the BCF Physical Logger creates and maintains a dynamic log in extended private memory or in an MVS/ESA data space. As BCF writes the log blocks to the physical log data set, it also copies them to the dynamic log. The log records remain in the dynamic log until the application
program issues a checkpoint call to commit the database changes; then BCF clears the log records from the dynamic log.

If a DL/I pseudo abend occurs and IMS dynamic backout is requested, BCF obtains the requested log block from BCF dynamic log. BCF dynamic log is expected to be used by applications that issue checkpoints calls. If the number of log blocks that are written between checkpoints exceeds the size of the dynamic log, BCF suspends writing records to the dynamic log. It resumes writing the records when the application program commits the unit of work by issuing a checkpoint call. While the dynamic log is suspended, any request to read a log block that does not exist in the dynamic log results in a dynamic backout error.

If an IMS dynamic backout error occurs and the Automatic Batch Backout option is set to Y, the execution controller invokes the IMS Batch Backout utility to accomplish batch backout. If IMS dynamic backout fails and the Automatic Batch Backout option is set to N, the execution controller does not invoke the IMS Batch Backout utility and job step recovery must be performed manually.

A numeric value for the Dynamic Log Option sets the number of blocks to use for the dynamic log. The size of each block is equal to the log block size. The value of the Dynamic Log Type option determines whether the dynamic log resides in extended private memory or in an MVS/ESA data space.

Panel field

Dynamic Log Option

Keyword

DYNAMic=value

Values

Set one of the following values:

- N— Do not create a dynamic log.
- nnnnn—Create the dynamic log. This value sets the number of blocks to use for the dynamic log (maximum 32767).

**WARNING**

If the dynamic log type is MEMORY (see “Dynamic Log Type” on page 95), the value specified for the dynamic log affects the region size.

Default

N
Dynamic Log Type

Use this option to control whether the BCF Physical Logger creates the dynamic log (requested by setting the Dynamic Log Option to a value other than N) in extended private memory or in an MVS/ESA data space.

If the request for memory or a data space fails, the dynamic log is not initialized.

Panel field

Dynamic Log Type

Keyword

DTYPE=value

Values

Set one of the following values:

- MEMORY—Create the dynamic log in extended private memory.
- DSPACE—Create the dynamic log using an MVS/ESA data space.

Default

MEMORY

Options on page 3 of the Processing Options panel

This section describes how to set values for the processing options on page 3 of the Processing Options panel.

Figure 13: Processing Options panel (Page 3)

<table>
<thead>
<tr>
<th>BCFOPY03</th>
<th>level</th>
<th>Processing Options</th>
<th>PAGE 3 OF 4</th>
</tr>
</thead>
</table>
| Command ===>

To store or change a value in the member, over type or change the value. To remove, blank or erase the value.

- G SYSOUT Class. .......................... * (A-Z,1-9,*"
- G Tape BKO. .............................. N (N/Y)
- G Max SLDS Block Count. ............... 0 (0-65535)
- G DL/I Activity Statistics. ............ N (N/IWAIT/FULL)
- G Record Activity Stats To. ......... HIST (N/HIST)
- G Number of Statistics Sets .......... 0 (0-999)
- G Logger Perf % Threshold .............. 10 (0-99)
- G Alert User ID .......................... (no default)

Place a ? in any option for functional description.
SYSOUT Class

Use this option to set the SYSOUT class to use for dynamically allocated print data sets.

When the Automatic Batch Backout option is set to YES, BCF invokes the IMS Batch Backout utility (and, if necessary, the IMS Log Recovery utility) to prepare the environment for job step restart. BCF must dynamically allocate the print data sets used by these utilities for error messages and reports. Use this option to provide the output class to use for these print data sets.

Panel field

SYSOUT Class

Keyword

SYSOUT=value

Values

Set one of the following values:

- xxxx—The device class to use (1 to 4 characters: A to Z, 1 to 9).
- *—Use the same output class as the job message class.

Default

*  

Tape BKO

Use this option to control whether IMS dynamic backout is supported when the log data set is written to tape. This option works with the Dynamic Log Option to circumvent dynamic backout restrictions on the batch logging process.

Without BCF, when a job step requests IMS dynamic backout support, the IMS Logical Logger pads the log blocks to the maximum log block size. Padding the blocks emulates the fixed block format necessary to support the IMS dynamic backout process. When you set the Dynamic Log Option to a value other than N, BCF creates a dynamic log to satisfy requests from the IMS dynamic backout process. Therefore, the physical log blocks no longer need to be padded to emulate fixed block format.
BCF dynamic log is used if the physical log is written to tape and you set the Tape BKO option to Y. BCF dynamic log is also used if the physical log is written to DASD and you set the Tape BKO option to Y or the IMS parameter $BKO=Y$ is set in the job step JCL. The BCF Physical Logger removes the pad bytes from the log blocks and allows backout from the BCF dynamic log.

This option can provide a substantial reduction in the size of a log data set that was previously written to DASD and used for dynamic backout. The amount of space saved, if any, depends on application design, number of database buffers, and logging parameters.

If you set the Tape BKO option to N, the BCF Physical Logger does not support dynamic backout if the physical log data set is recorded to tape.

Panel field

Tape BKO

Keyword

TBKO=value

Values

Set one of the following values:

- N—Do not allow dynamic backout support when logging to tape data sets.
- Y—If BCF provides a dynamic log, allow backout from the dynamic log for tape and DASD log data sets.

Default

N

Max SLDS Block Count

Use this option to control whether BCF attempts to provide system log data set (SLDS) duality.

If you set this option to 0, BCF does not attempt to provide SLDS duality. Normally, when the job step requests dual logging (by including the IEFRDER DD statement and the IEFRDER2 DD statement) and the log data sets are written to tape, individual SLDS volumes cannot be used interchangeably. Although the dual log data sets contain the same records, the individual volumes that comprise each data set can begin and end at different records within the data set. When the dual logs are
later used as input to another process (such as database recovery), an error found on
a volume of the primary log data set usually requires the process to restart at the
beginning. The process cannot usually continue from the point that the error
occurred because the volumes of secondary log data set are not identical to the
volumes of the primary log data set.

If you set this option to a number greater than zero, BCF attempts to provide SLDS
duality by using the same technique as used by the IMS Log Archive utility. BCF can
issue a force-end-of-volume command for both data sets after it writes the maximum
number of blocks you specified with this option. If one of the volumes reaches the
tape end-of-reel marker before reaching the maximum block count, BCF does not
perform any special processing to maintain volume duality.

BCF statistical reports provide information you can use to adjust the value of this
option to the optimum number of blocks.

Panel field

Max SLDS Block Count

Keyword

MSLDS=value

Values

Set one of the following values:

- \textit{n}n\textit{nn}nn—Issue a force-end-of-volume command for each tape log data set
  after writing no more than this maximum number of log blocks. Specify
  only 1 through 65535.

- 0—Do not attempt to provide SLDS duality.

Default

0

**DL/I Activity Statistics**

Use this option to control whether BCF accumulates statistics to produce the
following reports:

- DL/I Activity reports:
  - Data Set IWAIT Summary
  - Data Set IWAIT Distribution
— Data Set IWAIT Detail
— Background Write IWAIT Summary
— Background Write IWAIT Distribution

■ DL/I Call reports:
— DL/I Call IWAIT Summary
— DL/I Call Timing Summary

You can use these statistics to monitor and evaluate the performance of the database and DL/I environment. They can also aid in capacity planning. See “DL/I Activity reports” on page 221 for examples and explanations of the database IWAIT statistics. See “DL/I Call reports” on page 225 for examples and explanations of the DL/I call activity statistics.

BCF can store these statistics in the history data set. It can also write them to hard copy if you provide the BCFSTATS DD statement in the job step JCL.

**Note**
A small amount of CPU overhead is required to collect the statistics.

**Panel field**

DL/I Activity Statistics

**Keyword**

DLSTAT=value

**Values**

Set one of the following values:


**Default**

N
Record Activity Stats To

Use this option to control whether BCF records the accumulated job step statistics in the history data set. (This option does not control whether BCF writes them to the data set described by the BCFSTATS DD statement.)

If you set the value of this option to HIST, BCF creates a set of statistics in the history data set each time the job step executes. The key of this statistics set contains the job name, step name, program or PSB name, and the date and time that the job step executed. The set contains the statistics that BCF accumulated during the job step execution. BCF can use these statistics to rebuild the BCF statistical reports, or you can extract them for input to a statistical analysis package.

You can use the BCF ISPF interface to display the statistics saved in the history data set, print the statistics, or delete them. “Using BCF reports” on page 191 describes the reports and how to access them with the BCF ISPF interface.

Panel field

Record Activity Stats To

Keyword

RECord=value

Values

Set one of the following values:
- N—Do not record job step statistics.
- HIST—Record the job step statistics in a statistics record of the history data set.

Default

N

Number of Statistics Sets

Use this option to control the number of statistics sets that BCF stores in the history data set for each unique job step. A statistics set is all of the reports generated during an execution of a job step.

When the value of this option is greater than 0 but less than 999, BCF retains the statistics sets for this job step until the number of sets is equal to this maximum value. The next time the job step executes, BCF deletes the oldest statistics set and saves the new set so that the number of statistics sets does not exceed this maximum value.
value. If the old statistics set was created less than 24 hours before the current job step execution, BCF postpones the deletion. The next time the job step executes after 24 hours have elapsed, BCF deletes the oldest set or sets.

For consistency, you should set this option at the global or system level. If you set it at the job step level or execution level, confusing results are possible. For example, if this option is set to 6 at the global level and to 3 for a job step, some statistics sets will probably be deleted unexpectedly every time the job step is executed.

**Panel field**

Number of Statistics Sets

**Keyword**

`STATS=value`

**Values**

Set one of the following values:

- `nnn`—Retain this maximum number (1 to 998) of statistics sets for the job step in the history data set.
- `0`—Do not record statistics in the history data set.
- `999`—Do not delete statistics sets. To manage the size of the history data set, you should check the history data set periodically and manually delete the statistics sets that are no longer of interest or execute the BCF Purge utility (as described in “BCF Purge utility” on page 186).

**Default**

0

**Logger Perf % Threshold**

Use this option to monitor the amount of work that the BCF Physical Logger did to provide log integrity for database updates.

BCF can significantly decrease the elapsed time for many job steps by providing buffered tape support for the log data sets. The amount of decrease, if any, partly depends on the amount of log synchronization that the BCF Physical Logger must do to provide log integrity for database updates. The relationship between the total blocks written and the number of blocks written as a function of WALT processing reflects the improvement in buffered tape logging performance. This relationship is called the *logger performance percentage*. 
When the BCF Physical Logger is writing the log data set to a buffered tape device and the value of the Buffered Device Support option is BUFFER, the BCF Physical Logger calculates the logger performance percentage and reports it in message BMC44207I.

A relatively low value reported for the logger performance percentage indicates efficient WALT processing. As the value of the logger performance percentage increases, it is more likely that you can improve logging performance by tuning the database buffer pools, logging environment, or application program.

This option allows surveillance of the logger performance percentage. If the logger performance percentage exceeds the value you set for this option, BCF sends message BCF44209I to the TSO user ID designated with the Alert User ID option and to the job log. The message alerts you to the possible need for tuning to improve logging performance.

If you set a value of 0 for this option, BCF performs no surveillance of the logger performance percentage.

Panel field

Logger Perf % Threshold

Keyword

PERF=value

Values

Set one of the following values:

- nn—The percentage level (1 to 99) for the logging performance threshold.
- 0—No surveillance of logging performance is done.

Default

10

Alert User ID

Use this option to identify a TSO user ID to which BCF can send Alert category messages.

BCF can employ the TSO Send facility to notify an identified TSO user about exception conditions detected by BCF during job step execution. These conditions typically merit further evaluation. Here are some examples:

- TWI mode is in use.
Batch backout may be required.

The performance percentage threshold set with the Logger Perf % Threshold option has been exceeded.

The APPLICATION RESTART CONTROL messages in the BMC Documentation Center list and explain the Alert category messages in detail.

**Note**

If you are running under TSO Version 1.4 or later, BCF may be unable to send messages to an inactive user ID on the system on which BCF is running.

BCF can send messages to a user ID that is not active if the Database Utilities subsystem (DBUSS) is installed and started in the MVS environment. The DBUSS is included at no extra charge with the BMC Software IMS Database Utilities. It provides high-speed database read access for UNLOAD PLUS, IMAGE COPY PLUS, and POINTER CHECKER PLUS. The DBUSS is not included with BCF. The *Database Products for IMS Customization Guide* provides complete information about the DBUSS.

If you do not have the DBUSS installed and active, you may want to set the Alert User ID value to a user ID that always remains active, such as is typically used for operations.

If you provide no value for this option, BCF bypasses the attempt to send messages to a TSO user ID; however, it still writes the Alert category messages to the job log data set along with the other BCF messages.

**Panel field**

Alert User ID

**Keyword**

ALERT=value

**Values**

Specify the TSO user ID to receive Alert category messages or specify **NONE**.

**Default**

no default
Options on page 4 of the Processing Options panel

This section describes how to set values for the processing options on page 4 of the Processing Options panel.

Figure 14: Processing Options panel (Page 4)

<table>
<thead>
<tr>
<th>BCFOPY04</th>
<th>level Processing Options</th>
<th>PAGE 4 OF 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===</td>
<td>Next Page . .</td>
<td>Commands: CANcel</td>
</tr>
<tr>
<td>To store or change a value in the member, over type or change the value. To remove, blank or erase the value.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G Backout Variable (BKO) . . . . . . . . . . BKO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G Department Variable (DEPT). . . . . DEPT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G Use Log Compaction . . . . . . . . Y (Y/N)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G Manual Intervention Override. . . . N (N/Y)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G Ignore Checkpoints. . . . . . . . . N (N/Y/LOG)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G Use ABEND Code Series . . . . . . . . OLD (OLD/NEW)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Place a ? in any option for functional description.

Backout Variable (BKO)

Use this option to set the value of the backout variable.

When BCF dynamically allocates a new log data set, it constructs the data set name from the skeleton you have supplied in a dynamic allocation record of the REGISET. If the backout variable (&BKO) is part of the skeleton and BCF is allocating the log data set for the IMS Batch Backout utility to use for logging its changes, BCF substitutes the value of this option in the data set name. “Data set name skeletons” on page 132 provides complete information about this topic.

Panel field

Backout Variable (BKO)

Keyword

BKO=value

Values

Specify a valid MVS data set name qualifier (1 to 8 characters). Valid characters are A to Z, 0 to 9, #, @, $. The qualifier cannot start with a number.

Default

BACKOUT
Department Variable (DEPT)

Use this option to set the value of the department variable.

When BCF dynamically allocates a new log data set, it constructs the data set name from the skeleton you have supplied in a dynamic allocation record of the REGISET. If the department variable (&DEPT) is part of the skeleton, BCF substitutes the value of this option in the data set name. “Data set name skeletons” on page 132 provides complete information about this topic.

Panel field

Department Variable (DEPT)

Keyword

DEPT=value

Values

Specify a valid MVS data set name qualifier (1 to 8 characters). Valid characters are A to Z, 0 to 9, #, @, $. The qualifier cannot start with a number.

Default

DEPT

Use Log Compaction

Use this option to control the IDRC hardware compaction feature of the IBM 3480/3490 tape device.

Set this option to Y unless your installation is not able to use BCF Backout Assist. You cannot use BCF Batch Backout Assist if use of data spaces must be limited. For more information, see “Batch Backout Assist feature” on page 28.

Set this option to N when you do not want the hardware IDRC compaction feature active for log creation. When the IDRC feature is inactive for log creation, more physical tape space may be needed for the log data set. If you do not use BCF Batch Backout Assist, writing the log data set with the IDRC feature inactive allows the IMS Batch Backout utility to physically read the log tape data set backwards during batch backout. This processing can result in better elapsed time for backout when compared with backout execution using a log tape written with the IDRC feature active.
Panel field

Use Log Compaction

Keyword

COMP=value

Values

Set one of the following values:
- Y—Use the IDRC feature to activate log compaction.
- N—Deactivate the IDRC feature during log creation.

Default

Y

Manual Intervention Override

Use this option to override the Manual Intervention Required flag in the execution control record without changing the execution control record in the REGISET. This option takes effect during the restart of a job step that requires manual intervention.

If you set this option to Y, BCF allows you to retry the operation that previously failed without updating the execution control record. The problem requiring manual intervention must be corrected before the job is restarted.

If this option is set to N, you can retry the operation that previously failed by updating the execution control record. Set the Retry Manual Intervention field to Y on the Active Job Status Modification panel.

Panel field

Manual Intervention Override

Keyword

MANintov=value

Values

Set one of the following values:
- Y—Retry the operation that previously failed without updating the Manual Intervention Required flag in the execution control record of the REGISET.
- **N**—Do not retry the operation unless the Manual Intervention Required flag has been reset in the execution control record of the REGISET.

| Default | N |

## Ignore Checkpoints

Use this option to ignore IMS checkpoints that an application program can request with an IMS `CHKP` call. This option takes effect only if APPLICATION RESTART CONTROL (AR/CTL) is not active for the job step.

You may want to use this option if the application issues checkpoints but you want to be able to perform batch backout to the beginning of the job. Or you may be converting an application from a BMP to a batch job and want to ignore checkpoints when the application is executing in the batch environment.

When this option is set to N, BCF allows normal IMS checkpoint processing to occur. When this option is set to Y, BCF ignores the checkpoint request and returns control to the application program. The PCB status code is set to blanks.

When this option is set to Log, BCF allows normal checkpoint processing to occur but does not record type `X'41'` records in the IMS batch log.

### Note

If you specify Log, BCF does not allow IMS Dynamic Backout. This restriction prevents the possibility of inconsistent behavior between IMS and BCF during IMS Dynamic Backout when the Ignore Checkpoints option is set to Log. During IMS Dynamic Backout, IMS backs out only the changes that occurred since the last IMS application checkpoint. If BCF is active during IMS Dynamic Backout and the Ignore Checkpoints option is set to Log, BCF backs out changes to the beginning of the job. Setting the value to Y or Log will cause BCF to perform batch backout to the beginning of job step execution.

If AR/CTL is active during BCF initialization, BCF automatically sets the Ignore Checkpoints option to N. If this happens, BCF indicates the event by specifying **AUTO** in the **SOURCE** column of the Execution Options Summary report (saved in the statistics record of the history data set and in the BCFSTATS data set). This setting is different from the setting in the Pre-Execution Options Summary report (saved in the BCFPRINT data set).
Panel field

Ignore Checkpoints

Keyword

IGNorckp=value

Values

Set one of the following values:
- N—Allow normal checkpoint processing.
- Y—Prevent IMS from executing the checkpoint, and return control to the application program.
- Log—Allow normal checkpoint processing, but prevent IMS from recording type X'41' records in the IMS batch log.

Default

N

Use Abend Code Series

Use this option to control certain BCF codes that conflict with IMS codes.

To enable compatibility with existing automated operator routines, this option allows BCF to continue issuing "old" codes that are also issued by IMS. To eliminate the confusion that these conflicting codes may cause, this option also allows BCF to replace the old codes with "new" codes that have the same meanings. The old abend codes range from U061 to U714, and the new abend codes range from U1557 to U1575.

Setting the option to OLD causes BCF to continue issuing the old series of abend codes. Setting the option to NEW causes BCF to issue a new series of abend codes that do not conflict with codes issued by IMS.

Panel field

Use ABEND Code Series

Keyword

ABNcode=value

Values

Set one of the following values:
OLD—Issue old BCF abend codes (that conflict with IMS codes).

NEW—Issue new BCF abend codes (that have the same meaning as the old codes but do not conflict with IMS codes).

Default

OLD

Working with options at the execution (BCFSYSIN) level

To set BCF processing options at the execution level, you can code keywords in the BCFSYSIN control statement data set. This section explains BCFSYSIN syntax and summarizes the uses of the keywords.

BCFSYSIN control statement syntax

BCF employs a free-form control statement syntax similar to TSO and IDCAMS syntax. The figure below shows some example control statements.

Figure 15: Control statement syntax

```
//BCFSYSIN DD *
AUTO=YES, DISLOG=Y, BUFFTAPE=3400,
DTYPE=DSpace STATS=6 JOBLOG=FULL
REC=PDX, TBKO=N ALERT=BMCO00
BUFDEV=TWI
```

The elements you can combine are keywords, their values, separators, and continuation characters:

**Keywords and values**

Keywords can begin in any column and can be coded in any order. An equal sign (=) follows the keyword, and the value follows the equal sign, with no intervening blanks. The table in “BCFSYSIN keyword summary” on page 110 lists the BCFSYSIN keywords and their valid values.

**Separators**

Use one or more blanks or a comma to separate the keywords. (Do *not* use a blank or comma between a keyword and its value.)
Continuation

To continue a statement to the next line, no continuation character is necessary. You must keep a keyword and its value on the same line. You can use only columns 1 to 72.

BCFSYSIN syntax errors

If BCF detects a syntax error when processing the BCFSYSIN data set, it ignores the error. The job step execution uses the options available from the REGISET or valid BCFSYSIN data set options. BCF writes error messages concerning the syntax errors to the BCFPRINT data set.

BCFSYSIN keyword summary

The table below lists the BCFSYSIN keywords and their valid values. The page numbers in the table refer to the sections that provide complete information.

Table 7: BCFSYSIN control statement keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Valid values</th>
<th>Initial default value</th>
<th>Description and location</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABNcode</td>
<td>OLD, NEW</td>
<td>OLD</td>
<td>“Use Abend Code Series” on page 108</td>
</tr>
<tr>
<td>ALERT</td>
<td>TSO user ID, NONE</td>
<td>none</td>
<td>“Alert User ID” on page 102</td>
</tr>
<tr>
<td>AUTOObbo</td>
<td>Y, N</td>
<td>Y</td>
<td>“Automatic Batch Backout” on page 84</td>
</tr>
<tr>
<td>BCFactive</td>
<td>Y, N (C is not available in BCFSYSIN)</td>
<td>Y</td>
<td>“BATCH CONTROL FACILITY Active” on page 73</td>
</tr>
<tr>
<td>BECactive</td>
<td>Y, N</td>
<td>Y</td>
<td>“Use Execution Controller” on page 74</td>
</tr>
<tr>
<td>BKO</td>
<td>xxxxxxxxx</td>
<td>BACKOUT</td>
<td>“Backout Variable (BKO)” on page 104</td>
</tr>
<tr>
<td>BLPactive</td>
<td>Y, N</td>
<td>Y</td>
<td>“Use Physical Logger” on page 75</td>
</tr>
<tr>
<td>Keyword</td>
<td>Valid values</td>
<td>Initial default value</td>
<td>Description and location</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------</td>
<td>-----------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>BUFDEV</td>
<td>BUFFER</td>
<td>BUFFER</td>
<td>“Buffered Device Support” on page 86</td>
</tr>
<tr>
<td></td>
<td>TWI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BUFFTAPE</td>
<td>unitName</td>
<td>3400</td>
<td>“Buffered Tape Unit Name” on page 78</td>
</tr>
<tr>
<td>COMP</td>
<td>Y</td>
<td>Y</td>
<td>“Use Log Compaction” on page 105</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DASD</td>
<td>unitType</td>
<td>3380</td>
<td>“DASD Unit Name” on page 76</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DELVOL</td>
<td>N</td>
<td>N</td>
<td>“Delete Volume Records” on page 82</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEPT</td>
<td>xxxxxxxx</td>
<td>DEPT</td>
<td>“Department Variable (DEPT)” on page 105</td>
</tr>
<tr>
<td>DISLOG</td>
<td>N</td>
<td>Y</td>
<td>“Discard Logging Active” on page 88</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLSTAT</td>
<td>N</td>
<td></td>
<td>“DL/I Activity Statistics” on page 98</td>
</tr>
<tr>
<td></td>
<td>IWAIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DTYPE</td>
<td>MEMORY</td>
<td>MEMORY</td>
<td>“Dynamic Log Type” on page 95</td>
</tr>
<tr>
<td></td>
<td>DSPACE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DYNAMIC</td>
<td>N</td>
<td>nnnnn</td>
<td>“Dynamic Log Option” on page 93</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1-32767)</td>
<td></td>
</tr>
<tr>
<td>EXCHANGE</td>
<td>N</td>
<td>Y</td>
<td>“Exchange Log Device Type” on page 91</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>STATS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MATCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UPDATE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DBRC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IGNORCKP</td>
<td>N</td>
<td>Y</td>
<td>“Ignore Checkpoints” on page 107</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LOG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOBLOG</td>
<td>FULL</td>
<td>FULL</td>
<td>“JOBLOG Messages” on page 79</td>
</tr>
<tr>
<td></td>
<td>LIMITED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LERR</td>
<td>IMS</td>
<td>IMS</td>
<td>“Physical Logger Error Option” on page 81</td>
</tr>
<tr>
<td></td>
<td>ABEND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MANINTOV</td>
<td>N</td>
<td>Y</td>
<td>“Manual Intervention Override” on page 106</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keyword</td>
<td>Valid values</td>
<td>Initial default value</td>
<td>Description and location</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------</td>
<td>-----------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>MSLDS</td>
<td>nnnnn (0-65535)</td>
<td>0</td>
<td>“Max SLDS Block Count” on page 97</td>
</tr>
<tr>
<td>NOTIFY</td>
<td>BACKOUT</td>
<td>BACKOUT</td>
<td>“Notify Option for DBRC” on page 90</td>
</tr>
<tr>
<td></td>
<td>RECOVER</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PERF</td>
<td>nn</td>
<td>10</td>
<td>“Logger Perf % Threshold” on page 101</td>
</tr>
<tr>
<td>RE Cord</td>
<td>N N</td>
<td>N</td>
<td>“Record Activity Stats To” on page 100</td>
</tr>
<tr>
<td></td>
<td>HIST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STATS</td>
<td>nnn (0-999)</td>
<td>0</td>
<td>“Number of Statistics Sets” on page 100</td>
</tr>
<tr>
<td>SY SOUT</td>
<td>xxxx (A-Z, 1-9, *)</td>
<td>*</td>
<td>“SYSOUT Class” on page 96</td>
</tr>
<tr>
<td>TAPE</td>
<td>unitName</td>
<td>3400</td>
<td>“Standard Tape Unit Name” on page 77</td>
</tr>
<tr>
<td>TBKO</td>
<td>N N</td>
<td>N</td>
<td>“Tape BKO” on page 96</td>
</tr>
</tbody>
</table>
Preparation for dynamic allocation of log data sets

This section discusses how the BMC Software BATCH CONTROL FACILITY (BCF) component of APPLICATION RESTART CONTROL (AR/CTL) dynamically allocates the new data sets it uses for the primary and secondary logs. It describes how to prepare for dynamic allocation by providing information in one or more dynamic allocation records in the REGISET and how to specify data set name skeletons. Finally, it discusses generation data groups and provides information and examples for using them with the BCF component.

Understanding BCF dynamic allocation

Before you perform the tasks that prepare for dynamic allocation, you should understand when, why, and how BCF dynamically allocates data sets. You should also be aware of the various ways you can provide dynamic allocation information so you can choose the best method for the situation.

BCF can use three types of records in the REGISET to handle log information:

- Log volume records
- Dynamic allocation records
- Options records

For overview information about these records, see “Types of records” on page 41.

Dynamic allocation uses

BCF performs dynamic allocation when any of the following processing functions is active:
Log exchange

During initialization of the batch job execution, if you are using the Exchange Log Device Type option, BCF dynamically allocates log data sets to exchange those from the JCL.

Recovery

Recovery occurs after an application has abended and log recovery or batch backout is needed. This phase completes before step termination occurs in the original execution.

Restart

Restart occurs when log recovery or batch backout is not performed during the recovery phase, when an application was cancelled without recovery, or when a system failure prevented recovery.

Application reattach

Application reattach can occur when an application fails while APPLICATION RESTART CONTROL (AR/CTL) is active for the job step. After IMS environmental recovery is performed, AR/CTL can reattach the application without termination.

Automatic log exchange (for application reattach) enables BCF to provide new log data sets, as needed, when performing application reattach. When the AR/CTL application supervisor issues a call to BCF to perform initialization for application reattach, BCF automatically invokes the Exchange Log Device Type option for the job step. This option allows BCF to provide new log data sets as needed.

When the log data sets are allocated, the Execution Options report indicates that the Exchange Log Device Type option was automatically invoked (AUTO) by BCF.

Conditions for dynamic allocation

BCF performs dynamic allocation of new log data sets when any of the following conditions occur:
- Normal restart processing occurs.
- Log recovery occurs during application recovery.
- Log recovery occurs during application restart.
- Batch backout occurs during application recovery.
- Batch backout occurs during application restart.
- BCF provides new log data sets for application reattach.
The Exchange Log Device Type option is set to Y, STATS, MATCH, UPDATE, or DBRC. For more information, see “Exchange Log Device Type” on page 91.

Preparation for dynamic allocation

To prepare for dynamic allocation of the log data sets, you provide the necessary allocation information in one or more dynamic allocation records in the REGISET.

You create and update these records with the BCF ISPF interface, as described in “Accessing dynamic allocation records with the ISPF interface” on page 123. Each record contains parameters to use when allocating the data set, such as data set status, normal and conditional disposition, and space allocation. “Dynamic allocation options panels” on page 126 describes these parameters in detail.

Dynamic allocation records can be defined at the global, system, or job step level.

Global dynamic allocation records

Global dynamic allocation records apply to all jobs executed at the installation. The DEFAULT record is used with the Exchange Log Device Type option only if a more specific record cannot be found.

System dynamic allocation records

System dynamic allocation records apply to all jobs executed in a particular IMS system. Since the allocation information you want to use often depends on the device type, the system dynamic allocation record also applies to a device class. BCF uses the system level when you have not created an applicable job step dynamic allocation record.

Job step dynamic allocation records

Job step dynamic allocation records apply only to a particular job step.

Device classes for dynamic allocation records

This section describes the device classes for dynamic allocation records that are available with BCF and when each type is used during dynamic allocation processing.
BCF uses the most specific job step device class (when present) for all dynamic allocation requests. When the job step device class is not present, BCF uses a system device class as follows:

**DEFAULT**

BCF uses the DEFAULT system dynamic allocation record when it allocates a data set for use with the Exchange Log Device Type option, and also when you have not created a more specific system dynamic allocation record (depending on the value specified for the Exchange Log Device Type option).

*Note*

BMC Software recommends that you build a DEFAULT dynamic allocation record for ddname IEFRDER because, in special cases, BCF may need this record even if you are not using the Exchange Log Device Type option. If you plan to use dual logging, you must also build a DEFAULT dynamic allocation record for ddname IEFRDER2.

**DASD**

BCF uses the DASD system dynamic allocation record (if present) to allocate a new data set for use with the Exchange Log Device Type option when the original log data set was allocated to a DASD device and a more specific dynamic allocation record does not exist. BCF also uses this record to allocate a new data set for the IMS Batch Backout utility or the IMS Log Recovery utility when the original log data set was allocated to a DASD device and a more specific dynamic allocation record does not exist.

**TAPE**

BCF uses the TAPE system dynamic allocation record (if present) to allocate a new data set for use with the Exchange Log Device Type option when the original log data set was allocated to a standard tape device and a more specific dynamic allocation record does not exist. BCF also uses this record to allocate a new data set for the IMS Batch Backout utility or the IMS Log Recovery utility when the original log data set was allocated to a standard tape device and a more specific dynamic allocation record does not exist.

**BUFTAPE**

BCF uses the BUFTAPE system dynamic allocation record (if present) to allocate a new data set for use with the Exchange Log Device Type option when the original log data set was allocated to a buffered tape device and a more specific dynamic allocation record does not exist. BCF also uses this record to allocate a new data set for the IMS Batch Backout utility or the IMS Log Recovery utility when the original log data set was allocated to a buffered tape device and a more specific dynamic allocation record does not exist.
STATS

BCF uses the STATS system dynamic allocation record (if present) when it allocates a new data set for use with the Exchange Log Device Type option. This occurs when a value of STATS has been specified for the option and the original log data set is specified as DUMMY or NULLFILE. BCF assumes that log data sets allocated with STATS will not be used. The data set will not be opened unless the processing option of the associated PCB is set to UPDATE. When the processing option of the PCB is set to READ, discard logging is automatically set and the log data set is not opened.

BACKOUT

BCF uses the BACKOUT system dynamic allocation record (if present) when it allocates a log device for batch backout execution. This device class allows you to allocate the log data set created for backout to a different device class than was used for the original log data set. Specifying BACKOUT may be useful when the original log data set resides on tape or buffered tape and the output log data set created by backout is not expected to be large.

RECOVERY

BCF uses the RECOVERY system dynamic allocation record (if present) when it allocates a log device for log recovery. This device class allows you to allocate the log data set created for log recovery to a different device class than was used for the original log data set.

UPDATE

BCF uses the UPDATE system dynamic allocation record (if present) when it allocates a new data set for use with the Exchange Log Device Type option. This occurs when a value of UPDATE has been specified for the option, the PSB PROCOPT intent is UPDATE, and a more specific dynamic allocation record does not exist.

DBRC

BCF uses the DBRC system dynamic allocation record (if present) when it allocates a new data set for use with the Exchange Log Device Type option. This occurs when a value of DBRC has been specified for the option, the PSB PROCOPT intent has been specified as UPDATE, and the original log data set is specified as DUMMY or NULLFILE.

U616

If AR/CTL is active, BCF uses the U616 system dynamic allocation record for a new log data set that is created when a new log is allocated before application reattach. This device class allows you to change the device type when an application terminates abnormally with a U616 abend. This abend is issued when a log data set is unavailable during application execution.
Because an X’37’ abend could have caused the log data set to be unavailable, this device class allows you to change the device type to TAPE before reattach processing is performed. If the U616 dynamic allocation record is not present, the RECOVERY dynamic allocation record is used.

Note
This device class is used only if AR/CTL is installed.

DISCARD

BCF uses the DISCARD system dynamic allocation record when it allocates a data set for use with the Exchange Log Device Type option and the Discard Logging Active option. This device class allows you to allocate a small data set on DASD instead of tape and ensures that the data set will not be opened.

Note
Discard logging is not supported for utility (ULU) regions.

Conditions for new log allocation

When new logs are needed, BCF dynamically allocates them using a dynamic allocation record in the REGISET.

This section outlines the circumstances involved in eight dynamic allocation scenarios when new log data sets are created.

Table 8 on page 118 describes the option or utility used, the purpose of the allocation, whether the transaction is conditional, and the record search order used.

### Table 8: Dynamic allocation scenarios

<table>
<thead>
<tr>
<th>Option/utility used</th>
<th>Purpose of transaction</th>
<th>Conditional</th>
<th>Record search order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exchange Log Device Type=Y</td>
<td>Exchange log data set device types or naming structures without changing JCL.</td>
<td>no</td>
<td>1. Job Step record&lt;br&gt;2. System DISCARD record (if Discard Logging Active=YES)&lt;br&gt;3. System DEFAULT record&lt;br&gt;4. Global DEFAULT record</td>
</tr>
<tr>
<td>Exchange Log Device Type=STATS</td>
<td>Exchange log data set device types to obtain execution statistics if the original log is DUMMY or NULLFILE.</td>
<td>yes</td>
<td>1. Job Step record&lt;br&gt;2. System DISCARD record (if Discard Logging Active=YES)&lt;br&gt;3. System STATS record&lt;br&gt;4. System DEFAULT record&lt;br&gt;5. Global DEFAULT record</td>
</tr>
<tr>
<td>Option/utility used</td>
<td>Purpose of transaction</td>
<td>Conditional</td>
<td>Record search order</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>-------------</td>
<td>---------------------------------------------------------</td>
</tr>
<tr>
<td>Exchange Log Device Type=MATCH</td>
<td>Exchange log data set device types or naming structures without changing JCL.</td>
<td>no</td>
<td>1. Job Step record&lt;br&gt;2. System DISCARD record (if Discard Logging Active=YES)&lt;br&gt;3. System device (DASD, TAPE, BUFTAPE) record&lt;br&gt;4. System DEFAULT record&lt;br&gt;5. Global DEFAULT record</td>
</tr>
<tr>
<td>Exchange Log Device Type=UPDATE</td>
<td>Exchange log data set device types or naming structures without changing JCL when PSB PROCOPT is UPDATE.</td>
<td>yes</td>
<td>1. Job Step record&lt;br&gt;2. System DISCARD record (if Discard Logging Active=YES)&lt;br&gt;3. System UPDATE record&lt;br&gt;4. System DEFAULT record&lt;br&gt;5. Global DEFAULT record</td>
</tr>
<tr>
<td>Exchange Log Device Type=DBRC</td>
<td>Exchange log data set device types or naming structures without changing JCL when PSB PROCOPT is UPDATE and original log data set is DUMMY or NULLFILE.</td>
<td>yes</td>
<td>1. Job Step record&lt;br&gt;2. System DISCARD record (if Discard Logging Active=YES)&lt;br&gt;3. System DBRC record&lt;br&gt;4. System DEFAULT record&lt;br&gt;5. Global DEFAULT record</td>
</tr>
<tr>
<td>Application reattach active</td>
<td>Allocate new log data set for application to be used after AR/CTL completes re-attach processing.</td>
<td>yes</td>
<td>1. Job Step record&lt;br&gt;2. System U616 record (if a U616 abend occurred before)&lt;br&gt;3. System RECOVERY record&lt;br&gt;4. System DEFAULT record&lt;br&gt;5. Global DEFAULT record</td>
</tr>
</tbody>
</table>

**Note:** Available only if AR/CTL is installed.
Dynamic allocation process

When the execution controller determines that it needs to allocate a new IMS log data set for the primary log (with the ddname IEFDRDER) and/or the secondary log (with the ddname IEFDRDER2), it uses the following procedure:

1. BCF searches the REGISET for a dynamic allocation record identified with the ddname, job name, step name, and program or PSB name as defined in the current job step. If BCF finds this record at the job step level, it uses the information in this record to allocate the data set.

2. If BCF does not find a job step dynamic allocation record, and if it is allocating the data set for the IMS Batch Backout utility or the IMS Log Recovery utility, it searches for a dynamic allocation record defined for the ddname, the IMS system, and the same device class as the original log data set. If BCF finds this record at the system and device class level, it uses the information in this record to allocate the data set.

3. If BCF performs dynamic allocation, it will happen in one of the following ways:
   - If dynamic allocation for batch backout is in progress and BCF finds a system level record for backout, it uses this record for the allocation. If BCF does not find a system level record for backout, it continues with step 4.
   - If dynamic allocation for log recovery is in progress and BCF finds a system level record for log recovery, it uses this record for the allocation. If BCF does not find a system level record for log recovery, it continues with step 4.

4. If BCF does not find a system dynamic allocation record for the device type or if it is allocating the data set for use with the Exchange Log Device Type option, it searches for a dynamic allocation record defined for the ddname, the IMS system, and the default device class.

5. If BCF does not find a system dynamic allocation record, it searches for a dynamic allocation record defined for the ddname and the installation. If BCF finds this record at the global level, it uses the information in this record to allocate the data set.

6. If BCF does not find an applicable dynamic allocation record, it issues one of the following messages:
   - BMC44246I (BCF could not locate a dynamic allocation record)
   - BMC44243I (a dynamic allocation error occurred)
   - BMC44242I (manual intervention required)

The job step abends with a U625 (or U1567) abend. The abend code displayed depends on the value specified for the Use Abend Code Series option. For more information, see “Use Abend Code Series” on page 108.
Functions and options for dynamic allocation support

This section describes functions and options that BCF provides for dynamic allocation support.

Data set names

To determine the data set name to allocate, BCF checks the applicable dynamic allocation record for a data set name or data set name skeleton. If you have provided this value, BCF allocates the data set with this name.

DDnames

The ddnames that BCF dynamically allocates for use by the IMS Batch Backout and IMS Log Recovery utilities are controlled internally.

These ddnames may not match the ddnames normally used by the utility.

During restart processing, if the log data sets specified in the JCL of the job step are not members of a generation data group (GDG), BCF uses special ddnames that are compatible with the IMS Batch Backout and Log Recovery utilities. When log data sets are needed for batch backout or log recovery, BCF allocates data sets to the BCFRDER and BCFRDER2 DD statements. During execution of the IMS utility, BCF uses these data sets instead of the log data sets specified by the IEFRDER and IEFRDER2 DD statements. If IEFRDER or IEFRDER2 are specified as DUMMY or NULLFILE, BCF allocates temporary data sets to the IEFRDER and IEFRDER2 DD statements.

If the log data sets are members of a GDG, BCF uses the original data sets for backout or recovery as needed. For more information, see “BCF use of special DD names” on page 58.

Copy Current Space Allocation option

When the Copy Current Space Allocation option is set to Y in a dynamic allocation record being used to allocate a log data set, BCF saves the space information from the original data set, if it is allocated on DASD, and uses it for a newly allocated log data set.
If you use the Exchange Log Device Type option to exchange the log data set, the new log data set uses the space information from the original data set.

If the record specifies space allocation information and the Copy Current Space Allocation option is set to Y, the space information from the record is used only if the original log data set did not contain space information.

If a log data set is exchanged for a data set with a device type that does not support space information (tape), any space information that has been saved by this option is discarded.

The Copy Current Space Allocation option must be set to Y for use of the Log Recovery % Space Increase option and the Batch Backout % Space Increase option.

**Automatic Space Increase options**

When BCF allocates a log data set for log recovery or batch backout and the dynamic allocation record BCF is using specifies Y for the Copy Current Space Allocation option, BCF can automatically increase the primary space allocation of the allocated log data set.

The allocation record must specify acceptable values (0 to 999) for the Log Recovery % Space Increase option and the Batch Backout % Space Increase option. The new space allocation value does not change the space allocation value specified in the IEFRDER DD statement in the JCL.

**Compaction option**

You can specify Y for the Compaction option in a dynamic allocation record that BCF uses to allocate a log data set for tape.

BCF requests compaction for all log data sets that it allocates with that record.

**Error recovery**

When certain dynamic allocation errors occur that prevent a data set from being allocated for a job, BCF allows you to correct these errors and avoid the associated abends.
Accessing dynamic allocation records with the ISPF interface

You can use the BCF ISPF interface to create, display, and change the dynamic allocation records that BCF uses when it dynamically allocates log data sets. This section describes how to work with dynamic allocation records at the global, system, and job step levels.

Accessing global dynamic allocation records

To access global dynamic allocation records, perform the following steps:

1. Access the BATCH CONTROL FACILITY primary menu.
2. Type 7 (Set ddname skeleton information) in the choice entry field.
3. Press Enter.

BCF displays the Set Dynamic Allocation Skeleton Information panel (the figure below).

BCFPDDS0          Set Dynamic Allocation Skeleton Information
Command ===>      
Select an option. Then press Enter.
Option ........... 1. Set GLOBAL Level Skeleton
                   2. Set SYSTEM Level Skeleton
                   3. Set JOBSTEP Level Skeleton

4. Type 1 in the Option field.
5. Press Enter.

BCF displays the Select GLOBAL Dynamic Allocation Skeletons panel. (For generic information about this panel, see “Listing records” on page 50.) Select a record to display (and possibly change), delete an existing record, or use the ADD command to create a new record.

6. Press Enter.

- If you selected a record to display, BCF displays the first page of the GLOBAL Dynamic Allocation Options panel. See “Dynamic allocation options panels” on page 126.
- If you entered the ADD command, BCF displays a panel to allow you to enter the record qualifiers. (For generic information about this panel, see “Adding a
record” on page 51.) Type the record key values to use for the new record, and press Enter. BCF displays the first page of the GLOBAL Dynamic Allocation Options panel. See “Dynamic allocation options panels” on page 126.

 ■ If you selected a record to delete, BCF displays a panel so that you can confirm or cancel the delete action. (For generic information about this panel, see “Deleting a record” on page 52.)

**Accessing system dynamic allocation records**

Use this procedure to access system dynamic allocation records.

---

**Note**

System dynamic allocation records support a variety of device classes. You will see the device class names on the ISPF panels you work with in this procedure. For more information about the device classes, see “Device classes for dynamic allocation records” on page 115.

---

**To access system dynamic allocation records**

1. Access the BATCH CONTROL FACILITY primary menu.

2. Type 7 (Set ddname skeleton information) in the choice entry field.

3. Press Enter.

   BCF displays the Set Dynamic Allocation Skeleton Information panel (see the figure in “Accessing global dynamic allocation records” on page 123).

4. Type 2 in the Option field.

5. Press Enter.

   BCF displays a panel you can use to limit the list of records. (For generic information about this panel, see “Limiting the list of records” on page 50.)

6. Press Enter.

   BCF displays the Select SYSTEM Dynamic Allocation Skeletons panel. (For generic information about this panel, see “Listing records” on page 50.)

7. Select a record to display (and possibly change), delete an existing record, or use the ADD command to create a new record. Press Enter.
If you selected a record to display, BCF displays the first page of the SYSTEM Dynamic Allocation Options panel. See “Dynamic allocation options panels” on page 126.

If you entered the ADD command, BCF displays a panel to allow you to enter the record qualifiers. (For generic information about this panel, see “Adding a record” on page 51.) Type the record key values to use for the new record, and press Enter. BCF displays the first page of the SYSTEM Dynamic Allocation Options panel. See “Dynamic allocation options panels” on page 126.

If you selected a record to delete, BCF displays a panel so that you can confirm or cancel the delete action. (For generic information about this panel, see “Deleting a record” on page 52.)

**Accessing job step dynamic allocation records**

To access job step dynamic allocation records, perform the following steps:

1. Access the BATCH CONTROL FACILITY primary menu.
2. Type 7 (skeleton information) in the choice entry field.
3. Press Enter.

   BCF displays the Set Dynamic Allocation Skeleton Information panel (see the figure in “Accessing global dynamic allocation records” on page 123).
4. Type 3 in the Option field.
5. Press Enter.

   BCF displays a panel you can use to limit the list of records. (For generic information about this panel, see “Limiting the list of records” on page 50.)
6. Press Enter.

   BCF displays the Select JOBSTEP Dynamic Allocation Skeletons panel. (For generic information about this panel, see “Listing records” on page 50.)
7. Select a record to display (and possibly change), delete an existing record, or use the ADD command to create a new record. Press Enter.

   If you selected a record to display, BCF displays the first page of the JOBSTEP Dynamic Allocation Options panel. See “Dynamic allocation options panels” on page 126.
If you entered the ADD command, BCF displays a panel to allow you to enter the record qualifiers. (For generic information about this panel, see “Adding a record” on page 51.) Type the record key values to use for the new record, and press Enter. BCF displays the first page of the JOBSTEP Dynamic Allocation Options panel. See “Dynamic allocation options panels” on page 126.

If you selected a record to delete, BCF displays a panel so that you can confirm or cancel the delete action. (For generic information about this panel, see “Deleting a record” on page 52.)

Dynamic allocation options panels

When you select a dynamic allocation options record, BCF displays the first page of the Dynamic Allocation Options panel at the level you selected (GLOBAL, SYSTEM, or JOBSTEP).

The Dynamic Allocation panel consists of two pages. The general way you interact with the panel is the same for all pages, and the panels are identical in format, content, and functionality, except for the panel title and the level at which BCF uses the information.

For more information about many of the options shown on the Dynamic Allocation Options panels, refer to the appropriate IBM JCL reference manual for your environment.

Note
 If the value in the field is blank (for character fields) or a zero (for numeric fields), BCF uses the value that was provided when the data set was originally allocated.

Setting dynamic allocation options on page 1

Use this procedure to set values for the dynamic allocation options on page 1 of the Dynamic Allocation Options panel.
To set values for the dynamic allocation options on page 1

1 Set the following values as needed:

**SMS Data Set**

*(optional)* Specify whether the data set should use System Managed Storage (SMS). Set one of these values; the default value is N:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>The data set should not use SMS.</td>
</tr>
<tr>
<td>Y</td>
<td>The data set should use SMS. The only required fields (besides the data set name) are Data Set Status and the two disposition fields (Normal Disposition and Conditional Disposition).</td>
</tr>
</tbody>
</table>

**DSN**

Required. Specify the data set name that BCF should allocate during processing of the ddname associated with this dynamic allocation record. You can supply an actual data set name, or you can construct a data set name skeleton with any combination of constants and symbolic keywords. See “Data set name skeletons” on page 132 for more information.

**Unit**

*(optional)* Specify the unit type (1 to 8 characters) to allocate for the data set. BCF has no default value.

**Unit Count**

*(optional)* Specify the unit count (0 to 59) to allow for the data set. The default value is 0 (BCF uses the value from the original allocation).
Data Set Status

This display-only field shows the data set status; the value is NEW.

Normal Disposition

(required) Specify the disposition of the data set if the step terminates normally; the default value is CATLG.

Note

If you code a normal disposition of DELETE (so that the log data set is deleted after normal step termination) and an application abend occurs that causes log recovery or batch backout to be attached, BCF changes the normal disposition to KEEP. This action ensures that the log data set is available for log recovery or batch backout. You must then handle the disposition of the log data set; the data set is not cataloged.

Conditional Disposition

(required) Specify the disposition of the data set if the step terminates abnormally; the default value is CATLG.

Space Allocation Type

(required) Specify the type of space to allocate. Set one of the following values; the default value is CYLS:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CYLS</td>
<td>Allocate the space in cylinders.</td>
</tr>
<tr>
<td>TRKS</td>
<td>Allocate the space in tracks.</td>
</tr>
</tbody>
</table>

Primary Quantity

(required) Specify the amount of space (1 to 9999) to allocate for the first extent. The default value is 1.

Secondary Quantity

(optional) Specify the amount of space (0 to 9999) to allocate for additional extents. The default value is 1.

Block Size

(optional) Specify the block size in bytes (0 to 32760) to use for the data set. The block size must be at least four bytes larger than the logical record length. The default value is 32760.
Logical Record Length

*(optional)* Specify the logical record length in bytes (0 to 32756) to use for the data set. The logical record length must be at least four bytes smaller than the block size. The default value is 32756.

Volume Serial

*(optional)* Specify the volume serial identifier (1 to 6 characters) to use for the data set. You can specify a generic or esoteric volume name. BCF has no default value.

Number of Buffers

*(optional)* Specify the number of buffers (0 to 255) to allocate for the data set. The default value is 0 (BCF uses the value from the original allocation).

Setting dynamic allocation options on page 2

Use this procedure to set values for the dynamic allocation options on page 2 of the Dynamic Allocation Options panel.

<table>
<thead>
<tr>
<th>Command</th>
<th>Level</th>
<th>Dynamic Allocation Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCFPSK02</td>
<td></td>
<td>PAGE 2 OF 2</td>
</tr>
</tbody>
</table>

**Commands:**
- CANcel
- SAVE

**Values:**
- DSN For GDG Model DSCB
- Label Type: (SL/NL)
- Expiration Date: (yyyddd)
- Retention Period: (0-9999)
- Release Space at Data Set Close: (N/Y)
- Compaction: (N/Y)
- Copy Current UNIT Type: (N/Y)
- Copy Current DCB: (N/Y)
- Copy Current Space Allocation: (N/Y)
- If Copy Current Space Allocation is Y (yes): Log Recovery % Space Increase: (0-999)
- Batch Backout % Space Increase: (0-999)

Place a ? in any option for functional description.

To set the dynamic allocation options on page 2

1. Set the following values as needed:

   **DSN For GDG Model DSCB**

   *(optional)* Specify a data set name that replaces the default model data set control block (SYS1.MODEL) for generation data groups.
Label Type

*(optional)* Specify the label type for the tape data set. Set one of the following values; the default value is **SL**:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL</td>
<td>Standard label. This value is required for IMS log data sets.</td>
</tr>
<tr>
<td>NL</td>
<td>No label.</td>
</tr>
</tbody>
</table>

Expiration Date

*(optional)* Specify the Julian expiration date for the tape data set in the format *yyyyddd*, where *yyyy* are the four digits of the year and *ddd* is the day (1 to 366). The default value, at execution, is the **EXPDT** parameter coded on the original IERDRER or IERDRER2 DD statement.

Retention Period

*(optional)* Specify the number of days (0 to 366) to retain the tape data set. The default value is the **RETPD** parameter coded on the original IERDRER or IERDRER2 DD statement.

Release Space at Data Set Close

*(optional)* Specify whether to release unused space when the data set is closed. Set one of the following values; the default value is **N**:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Do not release unused space.</td>
</tr>
<tr>
<td>Y</td>
<td>Release unused space.</td>
</tr>
</tbody>
</table>

Compaction

*(optional)* Specify whether to use the compaction option so that IDRC can be specified when tape log data sets are allocated. Set one of the following values; the default value is **N**:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Do not use the compaction option.</td>
</tr>
<tr>
<td>Y</td>
<td>Use the compaction option.</td>
</tr>
</tbody>
</table>

Copy Current UNIT Type

*(optional)* Specify whether, when BCF performs log device exchange, to copy the UNIT information from the original IMS log data set. If the log data set is
specified in the JCL or the dynamic allocation record without UNIT information, BCF uses the MVS default device type. Set one of the following values; the default value is N:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Do not copy the current UNIT information.</td>
</tr>
<tr>
<td>Y</td>
<td>Copy the current UNIT information.</td>
</tr>
</tbody>
</table>

**Copy Current DCB**

*(optional)* Specify whether, when BCF performs log device exchange, to copy the DCB information from the original IMS log data set. If the log data set is specified in the JCL without DCB information, the IMS default is used. If the log data set is specified in the dynamic allocation record without DCB information, BCF uses a 10 KB default. Set one of the following values; the default value is N:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Do not copy the current DCB information.</td>
</tr>
<tr>
<td>Y</td>
<td>Copy the current DCB information.</td>
</tr>
</tbody>
</table>

**Copy Current Space Allocation**

*(optional)* Specify whether, when BCF performs log device exchange, to copy the space allocation information from the original IMS log data set. Set one of the following values; the default value is N:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Do not copy the current space allocation information.</td>
</tr>
<tr>
<td>Y</td>
<td>Copy the current space allocation information. This value is required for use of the Log Recovery % Space Increase option and the Batch Backout % Space Increase option.</td>
</tr>
</tbody>
</table>

**Log Recovery % Space Increase**

*(optional)* Specify the percentage of space (of the original log data set) to increase the primary allocation amount and secondary allocation amount to use for log recovery data sets. Set the percentage of increase (0 to 999); the default value is 0. The Copy Current Space Allocation option must be set to Y for use of the Log Recovery % Space Increase option.

**Batch Backout % Space Increase**

*(optional)* Specify the percentage of space (of the original log data set) to increase the primary allocation amount and secondary allocation amount to use for batch backout data sets. Set the percentage of increase (0 to 999); the
default value is 0. The Copy Current Space Allocation option must be set to Y for use of the Batch Backout % Space Increase option.

Management Class

(optional) Specify the class to be used to obtain the data-management-related information (for example, migration, backup, and retention criteria) for allocation of an SMS-managed data set. BCF has no default value.

Storage Class

(optional) Specify the class to be used to obtain the storage-related information for allocation of an SMS-managed data set. BCF has no default value.

Data Class

(optional) Specify the class to be used to obtain the data-related information (for example, SPACE, LRECL, BLKSIZE, BUFFNO) for allocation of an SMS-managed data set. BCF has no default value.

Data set name skeletons

When you prepare for dynamic allocation, you can provide the actual data set name to allocate for the IEFRDER or IEFRDER2 log data set, or you can provide a data set name skeleton.

When dynamic allocation of a data set is required, BCF constructs the data set names from the provided skeleton. It processes the skeleton from left to right, replacing the substitution keywords with the appropriate values. BCF stops processing the skeleton when it has substituted the current values for all keywords, or when the length of the data set name has reached 44 characters. If the length of the constructed data set name is greater than 44 characters, BCF uses only the first 44 characters.

WARNING

BCF performs only limited validation of the data set name constructed from the substitution mask. If the data set name constructed is not a valid data set name, a JCL or dynamic allocation error may occur.
Restrictions on generation data groups

If you use generation data group (GDG) log data sets to provide unique log data set names, BMC Software recommends that you consider using the data set name skeleton keywords. These allow BCF to create unique non-GDG data set names.

If you use GDGs for other reasons, be aware of the restrictions that apply when using them with BCF, especially when using the conditional steps that create the GDG log data sets. For more information, see “Generation data groups” on page 137.

Syntax and substitution keywords

You can use literal constants and substitution keywords in a data set name skeleton.

You can concatenate the constants and substitution keywords in any manner, with or without periods as separators between them, as long as the resulting qualifier is not more than eight characters long. You can use a constant and a substitution keyword in the same qualifier (with no intervening periods), but the constant must come first; for example, GRP&IMSID is a valid construction, but &IMSIDGRP is not.

You cannot use embedded blanks in the skeleton; the first blank terminates the definition of the data set name.

Some of the values, such as for the &JOBN keyword and the &STEPN keyword, are determined at job step execution. Other values, such as for the &BKO keyword and the &DEPT keywords, are defined in the global, system, or job step processing options records.

**WARNING**

When you use the &BKO keyword for allocations other than batch backout execution, you must specify it in a way that the resulting data set name contains at least two qualifiers. For example, ABC.&BKO is an invalid construction. The resulting constructed data set name would be ABC if the skeleton were being used for the allocation of a data set other than for batch backout utility execution.

ABC.&JOBN.&BKO is an example of the correct use of the &BKO keyword.

The following rules apply to the use of the &COPY keyword:

- It cannot be used with any other nodes or substitution keywords.
- It can take the form of a GDG specification; for example, it can be &COPY (+1).
- If the skeleton contains only the &COPY keyword and the existing data set name is a GDG, BCF assumes a generation increment of 1 for the new allocation.
If no data set name is associated with the log DD statement being allocated, BCF uses the following default expression to form the data set name:

\&JOBNAME.COPY.&DAY.&TIME

---

**Note**

If you use the `&COPY` keyword to copy the data set name from a non-GDG log data set, duplicate data set names may be created.

---

Table 9 on page 134 lists the substitution keywords you can include in the data set name skeleton and summarizes the rules for their use.

### Table 9: Data set name skeleton substitution keywords

<table>
<thead>
<tr>
<th>Substitution keyword</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;JOBN</td>
<td>The job name from the JOB statement in the application program JCL.</td>
</tr>
<tr>
<td>&amp;STEPN</td>
<td>The job step name from the EXEC statement in the application program JCL.</td>
</tr>
<tr>
<td>&amp;PROCSN</td>
<td>The step name from the procedure being executed by the execution JCL. If the JCL does not execute a procedure, the substitution value is set to nulls and (if the data set name node contains the keyword &amp;PROCSN only) the entire node is omitted.</td>
</tr>
<tr>
<td>&amp;PSB</td>
<td>The PSB name from the PSB value of the PARM parameter on the EXEC statement in the application program JCL.</td>
</tr>
<tr>
<td>&amp;PROG</td>
<td>The program name from the PGM parameter on the EXEC statement in the job step JCL.</td>
</tr>
<tr>
<td>&amp;IMSID</td>
<td>The IMSID from the IMSID value of the PARM parameter on the EXEC statement in the application program JCL or from the IMSID specified during stage 1 of the IMS system generation (SYSGEN).</td>
</tr>
<tr>
<td>&amp;BKO</td>
<td>The value of the BKO keyword from the most specific BCF options record that applies to this execution. (Note that this keyword must be used with other keywords.) The BKO keyword is a conditional value that BCF uses when dynamically allocating a data set for creation by the IMS Batch Backout utility. See “Backout Variable (BKO)” on page 104 for complete information.</td>
</tr>
<tr>
<td>&amp;DEPT</td>
<td>The value of the DEPT keyword from the most specific BCF options record that applies to this execution. See “Department Variable (DEPT)” on page 105 for complete information.</td>
</tr>
<tr>
<td>&amp;SEQ</td>
<td>The value of the SEQ keyword within the job step. BCF initially sets the value to 1 and increments it each time the keyword is used in the job step. The value is not maintained between successive job steps for the same or different jobs. When BCF constructs the data set qualifier using this keyword, it prefixes the seven-position numeric portion with the character S.</td>
</tr>
<tr>
<td>&amp;DAY</td>
<td>The Julian date of the execution, in the format D_yyddd, where D is a constant, yy is the last two digits of the year, and ddd is the day of the year (1 to 366).</td>
</tr>
<tr>
<td>Substitution keyword</td>
<td>Meaning</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------</td>
</tr>
<tr>
<td>&amp;TIME</td>
<td>The time of the execution, in the format Thhmmssst, where T is a constant, hh is the hour (00 to 23), mm is the minute (00 to 59), ss is the second (00 to 59), and t is the tenth of a second (0 to 9).</td>
</tr>
<tr>
<td>&amp;USER</td>
<td>The value specified for the USER keyword of the job statement. If BCF cannot locate a valid USER value on the job statement or by system default, the job name is used for this keyword.</td>
</tr>
<tr>
<td>&amp;COPY</td>
<td>BCF copies the data set name associated with the log DD statement.</td>
</tr>
<tr>
<td>&amp;COPYn</td>
<td>BCF copies the current DSN node n of the data set name associated with the log DD statement, where n is the position of the node in the data set name.</td>
</tr>
<tr>
<td>&amp;COPY(+i)</td>
<td>BCF copies the data set name of the GDG associated with the log DD statement and increments the generation by i.</td>
</tr>
</tbody>
</table>

**Example skeletons**

The following examples show how to define data set name skeletons and how BCF processes them:

- The following skeleton is defined:
  ABC.&IMSID.XYZ.&JOB
  If the IMS system ID is BMC1 and the job name is BLP0001, BCF constructs the following data set name:
  ABC.BMC1.XYZ.BLP0001

- The following skeleton is defined:
  ABC&IMSID.&JOBN.&SEQ
  If the IMS system ID is BMC1, the job name is BLP0001, and this is the first time BCF has encountered the &SEQ substitution keyword in this job step, BCF constructs the following data set name:
  ABCBMC1.BLP0001.S0000001

- The following skeleton, containing the substitution keyword &BK0, is defined:
  ABC.&BK0.&IMSID.XYZ.&JOBN
  The data set name BCF constructs depends on whether BCF is allocating the data set for use by the IMS Batch Backout utility or the application program:
  — If the data set is for the IMS Batch Backout utility and the value of the &BK0 substitution keyword is BACKOUT, BCF constructs the following data set name:
    ABC.BACKOUT.BMC1.XYZ.BLP0001
—If the data set is for the application program and the value of the &BK0 keyword is BACKOUT, BCF constructs the following data set name (BACKOUT does not appear):
   ABC.BMC1.XYZ.BLP0001

■ The following skeleton contains only substitution keywords:
   &USER.&JOBN.&STEPN.&DAY.&TIME
   For a user ID of BOB, a job name of BLP001, a step name of DLT0A, and a typical execution date and time, BCF constructs the following data set name:
   BOB.BLP0001.DLT0A.D96254.T1046035

■ The following skeleton is defined:
   &COPY
   For an existing job step IEFRDER with a data set name of BCF.IEFRDER, BCF constructs the following data set name:
   BCF.IEFRDER

■ The following skeleton is defined:
   &COPY
   For an existing job step IEFRDER with a data set name of BCF.GDG.IEFRDER(+1), BCF constructs the following data set name:
   BCF.GDG.IEFRDER(+1)

■ The following skeleton is defined:
   &COPY2.&COPY1
   For an existing job step IEFRDER with a data set name of BCF.GDG.IEFRDER(+1), BCF constructs the following data set name:
   GDG.BCF

Dynamic recovery from allocation errors

When certain dynamic allocation errors occur that prevent allocation of a data set for a job, Dynamic Allocation Error Recovery allows you to correct these errors using operator intervention and avoid the associated abends.

Member BCFDALIN of the BCF sample library provides examples of allocation errors that you can correct using this recovery process. Figure 17 on page 137 lists
the default allocation errors that can occur during dynamic allocation and that can be corrected by using this function.

**Figure 17: Default allocation errors that can be corrected**

*Table showing default allocation errors*

<table>
<thead>
<tr>
<th>DC</th>
<th>C'020C0000'</th>
<th>DATA SET IN USE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>C'02100000'</td>
<td>ALLOCATED TO ANOTHER JOB</td>
</tr>
<tr>
<td>DC</td>
<td>C'02140000'</td>
<td>DEVICE NOT AVAILABLE</td>
</tr>
<tr>
<td>DC</td>
<td>C'02180000'</td>
<td>NOT MOUNTED</td>
</tr>
<tr>
<td>DC</td>
<td>C'02200000'</td>
<td>VOLUME NOT AVAILABLE</td>
</tr>
<tr>
<td>DC</td>
<td>C'02240000'</td>
<td>NOT ENOUGH UNITS</td>
</tr>
<tr>
<td>DC</td>
<td>C'02280000'</td>
<td>VOLUME IN USE BY SYSTEM</td>
</tr>
<tr>
<td>DC</td>
<td>X'FF'</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To enable BCF to process allocation errors that can be corrected, modify the BCFDALIN member, and assemble and link-edit the module into the BCF load library. Member #BCFALIN of the BCF sample library contains sample JCL for the assemble and link-edit task. If you modify the BCFDALIN member, you should move it to another library so it will not be overwritten when you apply maintenance to BCF. The IBM publication *OS/VS2 System Programming Library: Job Management for MVS/370* lists all dynamic allocation error codes that can be used for allocation error processing under the topic "SVC 99 Return Codes."

---

**Generation data groups**

This section describes how to work with GDGs. The topics covered here include the GDG environments that BCF supports, how to set up dynamic allocation skeletons for GDG processing, how to use the &COPY substitution keyword, and restrictions on the use of GDGs. The section also discusses several scenarios you may encounter when employing GDGs and provides a number of examples.

**GDG log data set support**

If you use GDGs for your log data sets, you need to understand how BCF handles dynamic allocation of new GDG log data sets during execution.

Most installations use GDG log data sets to provide for unique data set names. If you are using GDGs for this reason, BCF offers alternate techniques to provide unique log data set names. You may want to investigate these options and consider using them for your log data sets.

Most issues involving the use of GDGs for log data sets result from the fact that a relative generation number applies to the entire job, and not just to specific job steps. Because BCF may need to allocate new log data set generations, this can create problems if multiple job steps use the same data set GDG base. This section describes
the processing provided by BCF and outline the restrictions on the use of GDGs for BCF log data sets.

Pay particular attention to this GDG information if you use any of the following techniques to execute IMS batch jobs:

- Multiple IMS job steps use the same GDG base name to create IMS log data sets.
- You have jobs that create new log generations for a GDG base used by a previous step or that reference log data sets created in a prior step and which are coded as COND=EVEN or COND=ONLY.
- You plan to use the Log Device Exchange option, and the new data sets will employ GDGs.
- You do not plan to allow BCF access to the enhanced processing routines.

**GDG environments**

BCF supports the following basic GDG environments:

- The GDG base of the currently allocated IERFORD or IERFORD2 in the JCL (the current log data set) is the same as the GDG base in the dynamic allocation skeleton.
- The GDG base of the current log data set is different than the GDG base in the dynamic allocation skeleton, or the current log data set is not a GDG.

BCF needs to allocate new log data sets at the following times:

- When the application has abended and you must perform log recovery or batch backout
- During restart for a job step that was canceled
- During restart for a job step after a system failure

For the first case, BCF allocates new log data sets as needed and executes log recovery, if necessary, and batch backout. As the appropriate recovery actions take place, BCF creates at least one, and possibly two, new generations of the GDG data set.

During restart after a failure or job cancel, BCF may be required to perform log recovery and batch backout. If both operations are performed, BCF creates two new GDG data sets before the execution of your application.
The following sections provide details on the planning and preparation you must do to use GDGs in each environment.

### Allocation skeleton setup

When you use GDG log data sets, no two jobs executing concurrently can reference the same GDG by relative generation.

You must adhere to this restriction when you set up your dynamic allocation skeletons because multiple job steps may be using a dynamic allocation skeleton concurrently. You must also ensure that the data set name constructed with the skeleton creates a unique GDG reference. The IBM MVS/DFP Using Data Sets manual provides additional considerations concerning this restriction on GDG use.

When you define a dynamic allocation skeleton to be used by BCF, you do not normally need to anticipate which job or job step is being recovered. You specify a relative generation increment of (+1) in the skeleton, and BCF automatically calculates the actual generation to be used based on several variables. Some of these variables include the following:

- generation of the log data set (IEFRDER or IEFREDER2) provided by the job step JCL
- number of times the skeleton has been used to allocate new log data sets for the current step

**Note**

When defining a dynamic allocation skeleton for a GDG log data set, recognize that what is usually a relative generation number (for example, +1) is actually a generation increment. The actual relative generation number used to allocate the log data set is calculated by BCF using this increment in its calculations.

### Use of &COPY for the data set name

To help define dynamic allocation skeletons for log data sets that are defined as GDGs, BCF provides the substitution keywords &COPY and &COPYn.

&COPY is used as the data set name. &COPYn is used to construct the data set name. These special keywords instruct BCF to copy the data set name from the current log data set specified in the JCL as the foundation for the log data set name to be allocated. If you want the new log data sets for log recovery or batch backout to use the same GDG base as the corresponding log data set provided in the job step JCL, using &COPY in the dynamic allocation skeleton causes BCF to use the existing GDG base when allocating new log data sets.
The &COPY keywords allow you to define a minimum number of dynamic allocation skeletons for use with GDGs. This technique may be especially useful in certain instances and may even be necessary if multiple job steps within a job create IMS log data sets. It is your responsibility to ensure that a GDG base exists for the resulting data set name when using &COPY keywords.

You can request that BCF use a GDG increment by specifying &COPY (or &COPY n) (+i) where i represents the number to increment the generation by. If the current log data set in your JCL is a GDG and you specify &COPY with no increment, BCF defaults to &COPY (+1).

**Note**
The use of the &COPY keyword is not restricted to GDG log data sets but can result in duplicate data set names if used with non-GDG log data sets since the same name would be used for both recovery and backout log data sets. This condition may, in turn, result in a NOT CATLG 2 error.

If &COPYn and &COPY are mixed, the keywords are resolved as indicated, but the data set name constructed may exceed the maximum allowed for a data set.

If the data set name of the existing log DD statement in the JCL does not contain a node in the position indicated, no value is substituted for the copy node (&COPYn ) keyword.

The copy node keywords can be specified in any order and can be repeated, if necessary.

If no data set name is associated with the log DD statement being allocated, the following values are used as the first four nodes of the data set name:

&JOBNAME.COPY.&DAY.&TIME

**Use of GDGs with the same skeleton and current log base**

This scenario deals with allocation cases involving the following conditions:

- When the application has abended
- When a job step is canceled (without BCF cancel intercept support), or when the system fails

**Application abend (same skeleton and current log base)**

When the application program abends, BCF assumes that the job step also abends and that no additional steps execute.
BCF prevents any COND=ONLY steps for batch backout from executing, but care must be taken if the job contains COND=EVEN or nonbatch backout COND=ONLY steps that refer to GDG log data sets created in a previous step or that create new log GDGs.

**Note**

For BCF to prevent COND=ONLY batch backout steps from executing, BCF must execute in these conditional steps. Furthermore, if any of these conditional execution steps are allowed to execute, your results can be unpredictable.

The following general rules apply and must be considered when planning for use of BCF with GDG log data sets:

- If BCF performs log recovery, the recovered log data set will be one generation greater than the generation of the log data set in the JCL created by your application execution. This means that any reference to the original log data set by conditional steps within the same job may result in the wrong log data set being used as input for the conditional job step.

- Since at least one, and possibly two, new generations may be created by BCF following an application abend, any attempts to create new log generations may fail unless the requested generation number is at least 3 more than the highest generation number used for a log data set in the previous steps.

**Note**

Specifying a relative generation number that is 3 larger than the highest normal generation number can result in skipping generations. Skipping generations does not usually result in a problem and does not affect later references to the generation by a relative generation number. You may consider this alternative for subsequent steps that reference the same GDG base.

Consider the following example:

<table>
<thead>
<tr>
<th>User’s execution log data set in JCL (+1)</th>
<th>G0001V00</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCF Batch Backout log data set (+2)</td>
<td>G0002V00</td>
</tr>
<tr>
<td>Conditional step log data set (+4)</td>
<td>G0004V00</td>
</tr>
</tbody>
</table>

In this example, three log data sets have been created. The conditional step was COND=EVEN or was not for batch backout. If these log data sets were referenced in a later job, the results would be as follows:

<table>
<thead>
<tr>
<th>Relative generation (+0)</th>
<th>G0004V00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative generation (-1)</td>
<td>G0002V00</td>
</tr>
<tr>
<td>Relative generation (-2)</td>
<td>G0001V00</td>
</tr>
</tbody>
</table>

Even though data set G0003V00 was never created, you can still reference the log data sets by using relative generation numbers.
Restart after failure or cancel (same skeleton and current log base)

When BCF performs recovery for a job that was active when a system failure occurred or for a job that was canceled (without BCF cancel intercept support), new log data sets are allocated as described in the paragraphs that follow.

BCF frees the log data set supplied in the job step and reallocates it for use by log recovery or batch backout. If you are using JES2, the current generation data set base is adjusted to reflect any log generations used during restart. This allows BCF to reallocate a log data set for the application’s use that has the same relative generation number as the original log data set. For example, if the log data set provided in the job step JCL was (+1), the log data set provided by BCF will also be (+1) and can be referenced as such in subsequent steps within the job.

If your installation uses JES3, BCF reallocates the log for application execution using the next relative generation for the log data set. If both log recovery and batch backout have been performed during restart, the next available generation is 2 more than the log generation number contained in the job step JCL, or (+3) for most jobs. Thus, subsequent job steps may experience JCL errors or inconsistent results if they refer to logs created in a restarted step by relative generation number. You can prevent these errors by ensuring that only one IMS step uses a given GDG base when creating new log data sets.

Use of GDGs with different skeleton and current log bases

This scenario discusses the following allocation cases:
- When the application has abended
- When a job step is canceled (without BCF cancel intercept support), or when the system fails

Application abend (different skeleton and current log base)

Because the new log data set to be allocated is a different GDG base data set, BCF allocates the next available generation for the skeleton GDG base.

The actual generation used for log recovery or batch backout is calculated and the log data set allocated.

Because a different log data set GDG base is being used for log data sets created by BCF, conditional job steps must not reference the original log data set because they may not obtain the correct one if BCF has recovered the original log data set. BCF
prevents execution of COND=ONLY batch backout steps if BCF is executing in the job step.

**Restart after failure or cancel (different skeleton and current log base)**

When BCF performs recovery for a job that was active when a system failure occurred or for a job that was canceled, the allocation of new log data sets occurs as follows:

- The new log data sets needed for log recovery or batch backout are allocated using the GDG base contained in the skeleton.

- BCF adjusts the GDG base for the new log data sets as each new data set is allocated so that any additional GDG data sets created for log device exchange or for later recovery operations are correct.

**Note**

If your system uses JES3, dynamic allocation errors may occur when a later job step attempts to allocate new log data sets using the same GDG base as the new log data sets that had been allocated during restart following the cancel or system failure.

**Allocation of old logs**

BCF and DBRC reference existing log data sets for recovery or for batch backout by a unique data set name and do not use relative GDG numbers.

For this reason, BCF and DBRC are insensitive to generation adjustments or skipped generations that can occur during recovery actions taken by BCF or during conditional job step execution.

**Allocation using the Exchange Log Device Type option**

When BCF allocates new log data sets for the Exchange Log Device Type option, the calculation of the relative generation number for the log data sets is based on the existing IEFRDER and IEFRDER2 data sets specified in the JCL.

For more information on this option, see “Exchange Log Device Type” on page 91.
When both use the same GDG base

The new data set and the existing log data sets have the same base when you use the Exchange Log Device Type option to switch from DASD to tape and retain the same data set name.

When the existing log data sets and the skeleton GDG base are the same, the calculated relative GDG number for the new log data set is a function of the existing log relative generation number and the skeleton generation increment.

When the GDG base is not the same

When the skeleton GDG base is not the same as the existing log data set or the existing data set is not a GDG or is specified as DUMMY, BCF uses the current JES step number to calculate the GDG relative generation number.

This calculation allows for multiple instances of log device exchange within the same job but could result in skipped generations if non-IMS steps are present in the job.

Restrictions on the use of GDGs

The following restrictions or considerations apply to the use of GDGs with BCF:

- When new logs allocated by BCF use the same GDG base as the log data sets in the job step JCL, conditional execution steps coded as COND=EVEN or COND=ONLY must not create new log data sets with a relative generation number of less than 3 greater than the highest relative generation number used for new log data sets within the job. A JCL error can occur for the IEFRDER or IEFRDER2 data sets if the generations referenced in the conditional step have been used by BCF during log recovery or batch backout.

- For BCF to dynamically adjust the GDG base, you must be using JES2.

- If you are using JES3, only one step within a job should use the same GDG base for a new log data set.

- Conditional steps coded as COND=EVEN or nonbatch backout COND=ONLY steps must not reference log data sets created by a step that has terminated abnormally and has used relative generation numbers. If BCF has recovered the original log data set, the conditional step may allocate the wrong log data set as input.

- The GDG data set name constructed from the dynamic allocation skeleton must be unique to a single concurrent execution as required by the IBM MVS/DFP Using Data Sets manual.
Examples of GDG processing

The following examples illustrate the processing BCF that provides for log data set GDGs.

Log GDGs allocated in a normal job execution with abend

This example shows a job containing two steps that create log GDGs and use the same GDG base.

When an application abends, BCF allocates a new log GDG that the IMS Batch Backout utility can use. After batch backout completes, the job terminates and any additional steps are flushed.

<table>
<thead>
<tr>
<th>GDG Base when job begins</th>
<th>= A.B.G0004V00</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCF Dynamic Allocation Skeleton</td>
<td>= A.B(+1)</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Step 1 log DD A.B(+1) resulting log DSN</td>
<td>= A.B.G0005V00</td>
</tr>
<tr>
<td>Step 2 log DD A.B(+2) resulting log DSN</td>
<td>= A.B.G0006V00</td>
</tr>
<tr>
<td>ABEND 0C7</td>
<td></td>
</tr>
<tr>
<td>BCF allocates new log DS A.B(+3) (Backout)</td>
<td>= A.B.G0007V00</td>
</tr>
</tbody>
</table>

Log GDGs allocated during restart using JES3

This example shows how BCF allocates new log GDGs during restart for a job that was canceled or after a system failure.

BCF frees the original log data set so that the data set can be used as the output for the log recovery utility. New logs are allocated as needed for batch backout and for the execution of an application. Notice that Step 2 uses a different GDG base for its log data set.

<table>
<thead>
<tr>
<th>GDG Base when job begins</th>
<th>= A.B.G0004V00</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDG Base when job begins</td>
<td>= C.D.G0009V00</td>
</tr>
<tr>
<td>BCF Dynamic Allocation Skeleton</td>
<td>= &amp;COPY(+1)</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Step 1 log DD A.B(+1) resulting log DSN</td>
<td>= A.B.G0005V00</td>
</tr>
<tr>
<td>BCF Frees log DS A.B(+1)</td>
<td>= A.B.G0005V00</td>
</tr>
<tr>
<td>BCF allocates new log DS A.B(+1) (Recovery)</td>
<td>= A.B.G0005V00</td>
</tr>
<tr>
<td>BCF allocates new log DS A.B(+2) (Backout)</td>
<td>= A.B.G0006V00</td>
</tr>
<tr>
<td>BCF allocates new log DS A.B(+3) (Customer)</td>
<td>= A.B.G0007V00</td>
</tr>
</tbody>
</table>
Step 2  log DD  C.D(+1)  resulting log DSN  = C.D.G0010V00

Log GDGs allocated during restart using JES2

This example also shows how BCF allocates new log GDGs during restart for a job that was canceled or after a system failure.

BCF frees the original log data set so that the data set can be used as the output for the Log Recovery utility. New logs are allocated as needed for batch backout and application execution. This example differs from Example 2 because Step 2 uses the same GDG base for the log data set. Because BCF uses JES2, Step 2 can allocate a log data set correctly.

<table>
<thead>
<tr>
<th>GDG Base when job begins</th>
<th>= A.B.G0004V00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjusted GDG base after recovery</td>
<td>= A.B.G0005V00</td>
</tr>
<tr>
<td>Adjusted GDG base after backout</td>
<td>= A.B.G0006V00</td>
</tr>
<tr>
<td>BCF Dynamic Allocation Skeleton</td>
<td>= &amp;COPY(+1)</td>
</tr>
</tbody>
</table>

Step 1  log DD  A.B(+1)  resulting log DSN  = A.B.G0005V00
BCF Frees log DS A.B(+1) = A.B.G0005V00
BCF allocates new log DS A.B(+1) (Recovery) = A.B.G0005V00
BCF allocates new log DS A.B(+1) (Backout) = A.B.G0006V00
BCF allocates new log DS A.B(+1) (Customer) = A.B.G0007V00

Step 2  log DD  A.B(+2)  resulting log DSN  = A.B.G0008V00

Log GDGs allocated during restart using JES3 (when BCF cannot adjust the GDG base)

A JCL error occurs when Step 2 attempts to use generation +2, which had been used as the output of batch backout.

This example is the same as “Log GDGs allocated during restart using JES3” on page 145, except that BCF cannot adjust the GDG base for the log data set because of the use of JES3.

<table>
<thead>
<tr>
<th>GDG Base when job begins</th>
<th>= A.B.G0004V00</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCF Dynamic Allocation Skeleton</td>
<td>= &amp;COPY(+1)</td>
</tr>
<tr>
<td>JES3 is being used</td>
<td></td>
</tr>
</tbody>
</table>

Step 1  log DD  A.B(+1)  resulting log DSN  = A.B.G0005V00
Log GDGs allocated in a job execution with abend and COND=ONLY (execution fails)

This example shows that the job contains three steps that create log GDGs and use the same GDG base.

When the application abends, BCF allocates a new log GDG for use by the IMS Batch Backout utility. After batch backout completes, the COND=ONLY step attempts execution but fails with a JCL error while attempting to allocate a new log with generation +3.

Note

BCF does not adjust the GDG base. If BCF adjusted the GDG base, the resulting JCL error could be prevented but the IMSLOGR reference would be invalid and may result in a subsequent invalid job step execution.

Log GDGs allocated in a job execution with abend and COND=ONLY (execution succeeds)

This example shows that the job contains three steps that create log GDGs and use the same GDG base.

When an application abends, BCF allocates a new log GDG for use by the IMS Batch Backout utility.
After batch backout completes, the COND=EVEN step executes correctly in this example because the new log created has requested a generation number at least 3 greater than the highest generation number used in a previous step.

**Note**

Even though the job step executed, it may not produce the expected results. If BCF has recovered the original log data set, the use of generation +2 may result in the step executing without the correct input.

<table>
<thead>
<tr>
<th>GDG Base when job begins</th>
<th>= A.B.G0004V00</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCF Dynamic Allocation Skeleton</td>
<td>= A.B(+1)</td>
</tr>
</tbody>
</table>

| Step | log DD | resulting log DSN | = A.B.G0005V00 |
|------|--------|-------------------|
| 1    | A.B(+1)|                   |
| 2    | A.B(+2)|                   |
|      |        | ABEND  0C7         |
|      |        | BCF allocates new log DS | = A.B.G0007V00 |
|      |        | A.B(+3) (Backout) |
| 3    | COND=EVEN Batch Backout utility |
|      | IMSLOGR A.B(+2) | resulting log DSN | = A.B.G0006V00 |
|      | log DD A.B(+5) | resulting log DSN | = A.B.G0009V00 |

**Log GDGs allocated in a job execution with abend and COND=EVEN**

This example shows that the job contains three steps that create log GDGs and use the same GDG base.

When an application abends, BCF allocates a new log GDG for the IMS Batch Backout utility. After batch backout completes, the COND=EVEN step executes correctly in this example because the new log created requested a generation number at least 3 greater than the highest generation number used in a previous step.

The step, however, executes with the wrong input log data set. It references the original log data set, not the one recovered by BCF.

**Note**

This execution of batch backout should not create any database damage or integrity exposures because BCF has already performed batch backout in a previous step. The output log data set created by this step must be included in any database recoveries or change accumulation runs. If DBRC is active for the job, the conditional attempt of a batch backout will fail because backout is no longer required.
Planning considerations

In summary, the suggestions in this section can help you use GDG log data sets when you convert to BCF.

Consider carefully the advisability of each of the following steps:

- Remove any conditional execution steps or adjust the usage of GDGs.
- Increase the number of generations for the log data set if your installation may expect skipped generations as a result of BCF support.
- Use non-GDG techniques to create unique data set names for log data sets.
Using BCF execution control records

This section describes the records that the BMC Software BATCH CONTROL FACILITY (BCF) component of APPLICATION RESTART CONTROL (AR/CTL) uses to control job step execution. Then it describes how to handle nonstandard situations that may arise during execution of an application program with BCF.

Overview of types of records that BCF uses

This section describes the types of records that BCF uses to manage control information during batch job execution:
- Execution control records
- Log volume records

Execution control records

BCF creates an execution control record in the REGISET when a job step makes its first database update.

This record remains in the REGISET until the job step completes successfully or the actions necessary to restart the job step (batch backout, log recovery, and so on) have been performed successfully. BCF then deletes the execution control record.

If the job step has not completed successfully, or if all of the actions necessary to restart the job step have not been performed, the presence of the execution control record in the REGISET indicates to BCF that the job step is not in a restartable condition. In most cases, BCF can restore the environment for job step restart automatically.

If BCF is not able to restore the environment for job step restart, manual intervention is required. You must perform all tasks necessary for job step restart (as discussed in
“Manual intervention” on page 174. Then you must delete the execution control record as described in this section.

**Note**
You must delete execution control records before applying BCF maintenance; otherwise, you could experience unpredictable processing results.

---

**Log volume records**

BCF creates a log volume record, or a pair of records if dual logging is in use, in the REGISET at the same time as it creates the job step execution record.

It uses the log volume records to maintain information about the primary and secondary log data sets created during the job step execution. It also uses them to create the Log Data Set Information reports that can be displayed with the other BCF statistical reports.

When the job step has completed successfully or all actions necessary to restore the environment to a restartable condition have been taken, the log volume record is no longer needed for batch backout purposes. BCF deletes it from the REGISET automatically; however, if the Delete Volume Records option is set to No, BCF copies the log volume record to the history data set before deleting it from the REGISET.

If the job step does not complete successfully and the Automatic Batch Backout option is set to Yes, BCF uses the log volume records to obtain the information about the log data sets to use as input to the IMS Batch Backout utility. If the job step requires manual intervention and the log volume records are present in the REGISET, you can display the log records to obtain the information about the logs needed for the manual intervention process.

---

**Accessing execution control records**

You can display, delete, print, and modify execution control records. To access execution control records, perform the following steps:

1. Access the BATCH CONTROL FACILITY primary menu.
2. Type 3 (Display active job steps) in the choice entry field.
3. Press **Enter**. BCF displays a panel you can use to limit the list of records. (For generic information about this panel, see “Limiting the list of records” on page 50.)
4 Press Enter. BCF displays the Active Jobsteps Display panel.

```
BCFPAJ00               Active Jobsteps Display                      1   OF 3
Command ===> _____________________________  Scroll ===> PAGE
SYSTEM ID = *                                                Commands: REFresh
Note: Intervention required for highlighted jobs.
Type one or more action codes. Then press Enter.
D=Delete   P=Print   S=Display   M=Modify.

<table>
<thead>
<tr>
<th>Jobname</th>
<th>Stepname</th>
<th>PSB/PGM</th>
<th>Run Date/Time</th>
<th>Last Status</th>
<th>SYSID</th>
</tr>
</thead>
<tbody>
<tr>
<td>_ BCFV3123</td>
<td>DLT0A</td>
<td>DBIHDOXX</td>
<td>02/26/1997 17:49</td>
<td>APPL ABENDED</td>
<td>SYSP</td>
</tr>
<tr>
<td>_ BCFV3124</td>
<td>RUN1</td>
<td>BA7$ALL</td>
<td>02/26/1997 18:59</td>
<td>APPL ACTIVE</td>
<td>SYSP</td>
</tr>
<tr>
<td>_ BCFV3137</td>
<td>DLT0G</td>
<td>DBIHDOXX</td>
<td>02/27/1997 14:37</td>
<td>APPL ABENDED</td>
<td>SYSP</td>
</tr>
</tbody>
</table>
```

5 In the field to the left of the entry, you can type one of the following action codes:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>Delete the execution control record. BCF displays a panel so that you can confirm or cancel the delete action. (For generic information about this panel, see “Deleting a record” on page 52.). WARNING: Do not delete an execution control record until all actions have been performed to restore the environment for job step restart. If you delete an execution control record without manually restoring the environment, data integrity exposures can occur. DBRC can be used to limit these data integrity exposures.</td>
</tr>
<tr>
<td>P</td>
<td>Print the information contained in the execution control record. BCF uses the print options set on the Set Online Print Options panel. This panel is discussed in the APPLICATION RESTART CONTROL Administrator Guide.</td>
</tr>
<tr>
<td>S</td>
<td>Display the information contained in the execution control record. See “Displaying an execution control record” on page 154.</td>
</tr>
<tr>
<td>M</td>
<td>Modify the execution control record to support special recovery processing. See “Modifying an execution control record” on page 159.</td>
</tr>
</tbody>
</table>

6 Check the value in the Last Status field. This field contains the status of the job step when the execution controller last updated the REGISET. Here are some status conditions:

- NOT AVAILABLE
- APPL ACTIVE
- BACKOUT ACTIVE
- LOG RECOV ACTIVE
- APPL ENDED
- BACKOUT ENDED
- LOG RECOV ENDED
- APPL ABENDED
- BACKOUT FAILED
- LOG RECOV FAILED
Displaying an execution control record

When you select an execution control record for display on the Active Jobstep Display panel, BCF opens a temporary sequential data set under the ISPF Browse facility.

BCF displays information about the execution control record in the Active Job Display panel (Figure 18 on page 154 and Figure 19 on page 154).

**Figure 18: Active Job Display panel (Part 1 of 2)**

<table>
<thead>
<tr>
<th>ISRBROBA</th>
<th>SYS97065.T135812.RA000.PAG.PXUTEMP.H01</th>
<th>Line 0000000 Col 001 079</th>
</tr>
</thead>
<tbody>
<tr>
<td>*************** Top of Data ********************</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATCH CONTROL FACILITY - ACTIVE JOB DISPLAY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INSID</td>
<td>JOBNAME</td>
<td>STEPNAM</td>
</tr>
<tr>
<td>BMCI</td>
<td>BCFV3123</td>
<td>DLT0A</td>
</tr>
<tr>
<td>PROCESS Control STATUS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>APPLICATION</td>
<td>LOG RECOVERY</td>
<td>BATCH BACKOUT</td>
</tr>
<tr>
<td>ABENDED</td>
<td>ABENDED=OFF</td>
<td>ABENDED=OFF</td>
</tr>
<tr>
<td>CONTENT OF STATUS FLAGS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>APPLICATION</td>
<td>LOG RECOVERY</td>
<td>BATCH BACKOUT</td>
</tr>
<tr>
<td>ACTIVE=ON</td>
<td>ACTIVE=OFF</td>
<td>ACTIVE=OFF</td>
</tr>
<tr>
<td>ENDED=ON</td>
<td>ENDED=OFF</td>
<td>ENDED=OFF</td>
</tr>
<tr>
<td>ABENDED=ON</td>
<td>ABENDED=OFF</td>
<td>ABENDED=OFF</td>
</tr>
</tbody>
</table>

You can use any of the valid ISPF browse commands on this panel (Figure 19 on page 154).

**Figure 19: Active Job Display panel (Part 2 of 2)**

<table>
<thead>
<tr>
<th>ISRBROBA</th>
<th>SYS97065.T135812.RA000.PAG.PXUTEMP.H01</th>
<th>Line 0000002 Col 001 079</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXECUTION ENVIRONMENT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBRC  IRLM  DUAL LOGGING  DB UPDATED  IMS DYNAMIC BACKOUT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YES   NO     NO       YES              NOT ATTEMPTED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>APPLICATION COMPLETION STATUS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RETURN CODE</td>
<td>COMPLETION CODE</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>U0778</td>
<td></td>
</tr>
<tr>
<td>SPECIAL OPERATION REQUESTS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RETRY MANUAL INTERVENTION</td>
<td>DEFER APPLICATION RESTART</td>
<td></td>
</tr>
<tr>
<td>NO</td>
<td>NO</td>
<td></td>
</tr>
</tbody>
</table>

*************** Bottom of Data ********************
General information

The following fields are included in the general information section:

**IMSID**

The IMS system ID associated with the job step.

**JOBNAME**

The job name coded on the JOB statement in the batch application program JCL.

**STEPNAME**

The job step name coded on the EXEC statement in the job step JCL.

**PSB/PGM**

The PSB name coded in the `PARAM` parameter of the EXEC statement in the batch application program JCL or the program name from the `PGM` parameter of the EXEC statement. The Qualify with Program processing option (set in the global options module) controls whether the PSB name or the program name appears in this field.

**DATE**

The calendar date (day of week, month, day, and year) when the job step started.

**TIME**

The time (hour and minute) when the job step started.

**SYSID**

The system ID (SYSID) that the job step executed on. If the execution control record is for a level of BCF that does not provide the SYSID information, this field appears blank. You must delete execution control records before applying BCF maintenance; otherwise, you could experience unpredictable processing results.

**PROCESS CONTROL STATUS**

This section reports the status of the job step as of the last time BCF updated the REGISET; the following fields may be included:

**APPLICATION**

The status of application program processing. The following values are valid:
### Value Description

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE</td>
<td>The application program has started executing but has not yet completed.</td>
</tr>
<tr>
<td>ENDED</td>
<td>The application program completed successfully.</td>
</tr>
<tr>
<td>ABENDED</td>
<td>The application program terminated abnormally.</td>
</tr>
</tbody>
</table>

#### LOG RECOVERY

The status of log recovery processing. The following values are valid:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQUIRED, ACTIVE</td>
<td>BCF has invoked the IMS Log Recovery utility, but it has not yet completed.</td>
</tr>
<tr>
<td>REQUIRED, ENDED</td>
<td>BCF invoked the IMS Log Recovery utility, and it has completed successfully.</td>
</tr>
<tr>
<td>REQUIRED, FAILED</td>
<td>BCF invoked the IMS Log Recovery utility, and it has terminated abnormally.</td>
</tr>
</tbody>
</table>

#### BATCH BACKOUT

The status of batch backout processing. The following values are valid:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQUIRED, ACTIVE</td>
<td>BCF has invoked the IMS Batch Backout utility, but it has not yet completed.</td>
</tr>
<tr>
<td>REQUIRED, ENDED</td>
<td>BCF invoked the IMS Batch Backout utility, and it has completed successfully.</td>
</tr>
<tr>
<td>REQUIRED, FAILED</td>
<td>BCF invoked the IMS Batch Backout utility, and it has terminated abnormally.</td>
</tr>
</tbody>
</table>

#### CONTENT OF STATUS FLAGS

This section indicates the current settings of the following groups of status flags:

#### APPLICATION

The status of application program processing indicated by the following flags:

<table>
<thead>
<tr>
<th>Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE</td>
<td>Whether the application program has started executing but has not yet completed.</td>
</tr>
<tr>
<td>ENDED</td>
<td>Whether the application program completed.</td>
</tr>
<tr>
<td>ABENDED</td>
<td>Whether the application program terminated abnormally.</td>
</tr>
</tbody>
</table>

#### LOG RECOVERY

The status of log recovery processing. The following values are valid:
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE</td>
<td>Whether log recovery has started processing but has not yet completed.</td>
</tr>
<tr>
<td>ENDED</td>
<td>Whether log recovery completed successfully.</td>
</tr>
<tr>
<td>ABENDED</td>
<td>Whether log recovery terminated abnormally.</td>
</tr>
</tbody>
</table>

**BATCH BACKOUT**

The status of batch backout processing indicated by the following flags:

<table>
<thead>
<tr>
<th>Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE</td>
<td>Whether batch backout has started processing but has not yet completed.</td>
</tr>
<tr>
<td>ENDED</td>
<td>Whether batch backout completed.</td>
</tr>
<tr>
<td>ABENDED</td>
<td>Whether batch backout terminated abnormally.</td>
</tr>
</tbody>
</table>

**EXECUTION ENVIRONMENT**

This section summarizes the key facts about the execution environment and provides information about the fields:

**DBRC**

Whether DBRC was active for the job step.

**IRLM**

Whether the IRLM was active for the job step.

**DUAL LOGGING**

Whether dual logging was requested for the job step.

**DB UPDATED**

Whether the application program updated any database during job step execution.

**IMS DYNAMIC BACKOUT**

The status of any attempts at IMS dynamic backout. The following values are valid:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE</td>
<td>The process has started but has not completed.</td>
</tr>
<tr>
<td>ENDED</td>
<td>The process has completed successfully.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>FAILED</td>
<td>The process has terminated abnormally.</td>
</tr>
<tr>
<td>NOT ATTEMPTED</td>
<td>The process was not attempted.</td>
</tr>
</tbody>
</table>

**APPLICATION COMPLETION STATUS**

This section summarizes the completion status of the application that was executing:

**RETURN CODE**

The return code displayed when the job step completed for the application program.

**COMPLETION CODE**

The completion code displayed when the job step completed for the application program.

**IMS BATCH BACKOUT UTILITY COMPLETION STATUS**

If the process control status of the IMS Batch Backout utility you are using is ABENDED, BCF generates this section to indicate the return code and the completion code:

**RETURN CODE**

The return code displayed when the IMS Batch Backout utility finishes processing.

**COMPLETION CODE**

The completion code displayed when the IMS Batch Backout utility finishes processing.

**IMS LOG RECOVERY UTILITY COMPLETION STATUS**

If the process control status of the IMS Log Recovery utility you are using is ABENDED, BCF generates this section to indicate the return code and the completion code:

**RETURN CODE**

The return code displayed when the IMS Log Recovery utility finishes processing.
COMPLETION CODE

The completion code displayed when the IMS Log Recovery utility finishes processing.

SPECIAL OPERATION REQUESTS

This section indicates whether the following special processing functions were requested:

Retry Manual Intervention

Whether the Retry Manual Intervention function was requested.

Defer Application Restart

Whether the Defer Application Restart function was requested.

Modifying an execution control record

When you select an execution control record to modify on the Active Jobsteps Display panel, BCF displays a panel that displays a warning and allows you to confirm or cancel your modify request.

If you proceed with the modification, BCF displays the Active Job Status Modification panel (Figure 20 on page 159), allowing you to update the execution control record to specify special recovery processing.

Figure 20: Active Job Status Modification panel

BCFPAJOM  Active Job Status Modification  PAGE 1 of 1
Command  

<table>
<thead>
<tr>
<th>Command</th>
<th>Active Job Status Modification</th>
<th>PAGE 1 of 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retry Manual Intervention</td>
<td>N</td>
<td>(Y/N)</td>
</tr>
<tr>
<td>Defer Application Restart</td>
<td>N</td>
<td>(Y/N)</td>
</tr>
<tr>
<td>Set Job Abend</td>
<td>N</td>
<td>(Y/N)</td>
</tr>
</tbody>
</table>

Note: Only fields that are applicable can be modified.

To modify an execution control record

1 Set values in the following fields as appropriate:

Retry manual intervention

Use this field to allow one retry of the previously failed process. The record must be marked "Manual Intervention Required." Type one of the following options:
**Option | Description**
---|---
Y | Update the execution control record for one retry.
N | Do not update the execution control record.

**Defer application restart**

Use this field to request restart processing that recovers the environment to a point ready for application restart, then terminates. BCF updates the execution control record so that only recovery and backout actions are attempted when the job restarts. Type one of the following options:

**Option | Description**
---|---
Y | Defer application restart for the application program.
N | Do not defer restart processing for the application program.

**Set job abend**

Use this field to update the execution control record so its status indicates that the job abended during its last processing.

As BCF updates the status of the execution control record, it also attempts to catalog the log volumes if necessary. If a problem occurs with the catalog attempt, BCF displays an error panel; you may need to catalog the log volumes manually.

**Note**

Using the Automatic Restart Preparation utility or the Active Job utility is a better method for handling execution control records and log volumes after a system failure.

Type one of the following options:

**Option | Description**
---|---
Y | Set the execution control record status to abended.
N | Do not update the execution control record.
Cataloging of log volumes after system failure

If the processor or system fails during execution of the job, the log data set may not be cataloged when restart is attempted.

To ensure that BCF can perform the restart correctly, you must set jobs left in active status to a status of abended. This process also ensures that any log volumes created during the previous execution are cataloged if necessary.

A log may need to be cataloged in these situations:

- An application was active when the system failed. The IMS output log being created may need to be cataloged.
- The IMS Batch Backout utility, automatically called by BCF, was executing when the failure occurred. The output of batch backout needs to be cataloged but will not be used during later BCF processing. If the log is not cataloged, subsequent executions of batch backout can create duplicate log generations. Because DBRC knows about the real log data set name, recovery may fail when DBRC requests the same log data set twice.

You can use the Automatic Restart Preparation utility or the Active Job utility to catalog all log volumes after a system failure. You can use the Set job abend field on the Active Job Status Modification panel to catalog log volumes for individual jobs. Using these methods prevents BCF restart failures caused by prior log data sets that were not cataloged and avoids the implications of handling GDGs at that point.

Use of the Active Job utility

The Active Job utility is used to control the actions of BCF during restart processing by setting indicators in the execution control record.

This utility can be used before restart processing after a job step has been canceled (without cancel intercept support) or after a system failure. The Active Job utility can be useful when an application program encounters a problem that is not easily fixed but the databases and logs must be recovered immediately. In such a situation, this utility allows you to change the active job entry in the REGISET to indicate that any required log recovery or batch backouts functions should be performed without restarting the application. This utility also ensures that the required log data sets have been cataloged.

If you want the operator to be prompted before the execution control record is processed, specify the ASK parameter. To list the active job records found without
changing their status, specify the **LISTONLY** parameter. In addition, you can update an execution control record to change its status from active to abended by specifying the **ABENDED** parameter.

### JCL requirements for the Active Job utility

The figure below shows sample JCL for executing the Active Job utility. Member #BCFACTJ of the BCF sample library contains sample JCL for this task.

**Figure 21: Active Job utility—sample JCL**

```
//ACTJ EXEC PGM=BCFACTJ0,PARM='SSID=ssid,SYSTEM=SYSX,ASK=NO,
// LISTONLY=NO'
//STEPLIB DD DSN=BMC.BCF.LOAD,DISP=SHR
// DD DSN=BMC.AES.LOAD,DISP=SHR
//BCFPRINT DD SYSOUT=* 
//BCFSYSIN DD *
SET JOB(JOB*) DEFER(Y) ABENDED(Y) DBRC(N)
DELETE JOB(ABCD1234) STEP(*) PSB(PSBNAME) RUNDATE(01/29/1998)
//
```

### EXEC statement

The program to execute is BCFACTJ0.

Code the following parameters on the EXEC statement to control the Active Job utility functions. The utility requires values for all of these parameters and supplies some defaults.

**SSID**

*(required)* Code a one-to-four byte value that identifies the BMC Consolidated Subsystem (BCSS) that the job steps were using.

**SYSTEM**

*(optional)* Code a one-to-four byte value that corresponds to the MVS SYSID of the system where job steps were executing (sometimes called the SMFID). If no value is specified, the SYSID of the system where the Active Job utility is executing is used.

**ASK**

*(optional)* Code a value to indicate whether to prompt the operator for confirmation before each active entry is processed. The following values are valid; the default value is **NO**.
Value | Description
--- | ---
NO | Do not prompt the operator for confirmation before execution control records are processed.
YES | Prompt the operator for confirmation before execution control records are processed by issuing message BMC48078A.

**LISTONLY**

*(optional)* Code a value to indicate whether the Active Job utility should process execution control records or list only the execution control records found. The following values are valid; the default value is NO:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Process the execution control records as specified in the JCL.</td>
</tr>
<tr>
<td>YES</td>
<td>List the execution control records found without processing them.</td>
</tr>
</tbody>
</table>

**DD statements**

Specify the following DD statements:

**STEPLIB DD**

*(required)* Provide the name of the library that contains the AR/CTL and BCF component load modules. The AR/CTL and BCF component load modules should reside in the same library.

**BCFPRINT DD**

*(required)* Provide a data set to contain the report indicating the actions that the utility performed for BCF records. This DD statement usually defines a SYSOUT type data set, but you can specify a data set on DASD with the following data control block (DCB) parameters: RECFM=FBA, LRECL=132.

**BCFSYSIN DD**

*(required)* You can code the BCFSYSIN DD statement to describe the BCF control statement data set. The BCFSYSIN data set can be a sequential data set or a member of a PDS, or it can be specified instream. The record length must be defined as 80 bytes.

**BCFSYSIN commands**

You can specify the following BCFSYSIN commands to control the Active Job utility:
SET

(optional) Alter settings in the active job entry in the REGISET to control restart processing.

DELETE

(optional) Delete the active job entry and its associated statistics record, primary log record, and secondary log record from the REGISET.

Note
The EXEC statement ASK=YES parameter, which requests operator confirmation, is automatically set for any entry to be deleted.

BCFSYSIN keywords

You can specify the following BCFSYSIN keywords to control the Active Job utility:

JOB

(required) The job name of the active record.

STEP

(optional) The step name of the active record.

PSB

(optional) The PSB name or program name used to qualify the entry.

RUNDATE

(optional) The date when the active step began execution, in mm/dd/yyyy format.

RUNTIME

(optional) The time when the active step began execution, in hh/mm/ss format.

DEFER

(optional) Indicates whether the Active Job utility should set the Defer Application Restart flag (for use with SET command only). The following values are valid:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not set the Defer Application Restart flag.</td>
</tr>
</tbody>
</table>
### Value | Description
---|---
YES | Set the Defer Application Restart flag.

**ABENDED**

*(optional)* Allows updating an execution control record to change its current status to abended *(for use with SET command only)*. The following values are valid:

| Value | Description |
---|---|
NO | Do not set the execution control record to abended. |
YES | Set the execution control record to abended and catalog log data sets. |

**DBRC**

*(optional)* Indicates whether DBRC is notified that job step terminated abnormally *(for use with SET command only)*. The following values are valid:

| Value | Description |
---|---|
NO | Do not notify DBRC that the job step abended. |
YES | When an active job entry is being changed to abended, notify DBRC to change the subsystem record to abnormal termination. |

**BCFPRINT reports**

After processing for the Active Job utility is completed, the utility provides the Active Job Utility reports *(the figure below)* in the BCFPRINT data set.
Figure 22: Active Job utility report

The following fields are included in the Active Job Utility reports:

**SYSTEM**

The MVS system where the application program executed to produce the execution control records.

**ASK**

Indicates whether the operator wants to be prompted for confirmation before each active entry is processed. The following values are valid:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not prompt the operator for confirmation before execution control records are processed.</td>
</tr>
<tr>
<td>YES</td>
<td>Prompt the operator for confirmation, before execution control records are processed, by issuing message BMC48078A.</td>
</tr>
</tbody>
</table>

**LISTONLY**

Indicates whether the Active Job utility should update execution control records or list only the execution control records to be processed. The following values are valid:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Process the execution control records as specified in the control statement.</td>
</tr>
</tbody>
</table>
### Value Description

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>List the execution control records found without updating them.</td>
</tr>
</tbody>
</table>

**JOBNAME**

The job name of the execution control record.

**STEPNAME**

The step name of the execution control record.

**PSB/PGM**

The PSB name or program name used to qualify the entry.

**DATE**

The date when the job step began execution, in *mm/dd/yyyy* format.

**TIME**

The time when the job step began execution, in *hh:mm* format.

**SYSID**

The system ID of the system where the job step executed.

**OPER RESP**

The operator’s response to message BMC48078A if *ASK=YES* was specified.

**DEF REST**

The current status of the Defer Application Restart flag. An * after the status indicates that the Active Job utility changed the status.

**JOB ABND**

Indicates whether the current status of an execution control record was changed to abended. An * after the status indicates that the Active Job utility changed the status.

**CATALOG PRILOG**

Indicates the action taken by the Active Job utility to catalog primary log data sets. The following values are valid:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/R</td>
<td>Cataloging the data set is not required.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LOCATED</td>
<td>The data set was already cataloged.</td>
</tr>
<tr>
<td>RC=xx</td>
<td>An attempt was made to catalog the data set, and the field contains the return code from the catalog function. A return code other than 0 indicates that an error has occurred.</td>
</tr>
</tbody>
</table>

**CATALOG SECLOG**

Indicates the action taken by the Active Job utility to catalog secondary log data sets. The following values are valid:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/R</td>
<td>Cataloging the data set is not required.</td>
</tr>
<tr>
<td>LOCATED</td>
<td>The data set was already cataloged.</td>
</tr>
<tr>
<td>RC=xx</td>
<td>An attempt was made to catalog the data set, and the field contains the return code from the catalog function. A return code other than 0 indicates that an error has occurred.</td>
</tr>
</tbody>
</table>

**DB UPT**

Indicates that the job has updated IMS databases.

**DISC LOG**

Indicates whether the job used the Discard Logging feature of BCF.

**DUAL LOG**

Indicates whether the job used dual logging.

**Utility return and abend codes**

The Active Job utility produces the following return and abend codes:

0

The utility ended normally. At least one execution control record was updated, and no abnormal conditions were detected.

4

The utility ended normally. No execution control record that matched the parameters in the control statements was found in the REGISET.
The utility ended normally. For at least one execution control record, an error occurred when the utility attempted to notify DBRC.

U1554

The utility ended abnormally. It issues a U1554 abend code to indicate that utility processing was not possible for one of the following reasons:

- The parameter information provided for the job was missing or invalid. The BCFPRINT data set contains a more detailed error message indicating the cause of the error.

- The utility encountered an error when processing one of the execution control records. The BCFPRINT data set contains a more detailed error message indicating the cause of the error. The abend does not occur until after the utility has attempted to process each selected execution control record.

Handling nonstandard situations

Standard operations for BCF are highly automated; however, certain nonstandard situations can require manual intervention. These situations include the termination of job steps without normal MVS termination processing and the damage or destruction of essential BCF components.

This section describes the actions to take in a variety of operational situations. It covers situations in which BCF has been installed correctly and is working as designed.

The messages documented in the BMC Documentation Center and in the Message Viewing utility (option 12 on the Application Enhancement Series primary menu) are another resource for problem determination and resolution.

Special restart processing situations

BCF maintains information about the status of the job step execution in an execution control record of the REGISET.

BCF creates this execution control record when it detects the first update to a database. As job step execution continues, BCF updates the execution control record to reflect critical changes in job step status. When the job step terminates with all necessary processing performed, BCF deletes the execution control record from the
REGISET. If the job step terminates abnormally, the execution control record is not deleted.

The following examples demonstrate the special restart processing capabilities provided by BCF.

**Application ended normally but execution control record was not deleted**

An application has ended normally and the execution control record has been updated to indicate successful completion.

A system failure occurred, or the job was canceled before the execution controller could delete the execution control record.

On restart, BCF takes the following processing steps for this condition:

- BCF deletes the execution control record.
- BCF issues message BMC48064I, indicating normal ending.
- BCF ends the job step with the condition code passed by the application in the previous execution.

**Application abended and job step ended without batch backout processing**

An application has abended.

The Log Recovery utility executed and ended successfully, but the system failed or the job was canceled before the need for batch backout could be determined.

On restart, BCF takes the following processing steps for this condition:

- BCF performs batch backout if required.
- BCF deletes the execution control record.
- BCF executes the application.

**Application abended; log recovery and batch backout were completed**

The application has abended.

Log recovery and batch backout have completed successfully. Log recovery and batch backout executed, but the system failed or the job was cancelled before the entry was deleted.
On restart, BCF takes the following processing steps for this condition:

- BCF deletes the execution control record.
- BCF executes the application.

**Application abended and retry manual intervention was requested**

The application has abended.

The Log Recovery utility executed but abended and requested manual intervention. The problem was corrected. The operations staff modified the execution control record and used the Active Job Status Modification panel to retry manual intervention.

On restart, BCF takes the following processing steps for this condition:

- BCF issues message BMC44232I (retry manual intervention active).
- BCF performs log recovery again.
- BCF performs batch backout if required.
- BCF deletes the execution control record.
- BCF executes the application.

---

**Note**

The Retry Manual Intervention flag must be on before BCF will retry a recovery or backout action marked abended. This flag is set automatically when you set the Set Job Abend field to Y on the Active Job Status Modification panel (see figure in "Modifying an execution control record" on page 159. If the job was canceled, you must set the Retry Manual Intervention field to Y on Active Job Status Modification panel.

---

**Application abended, batch backout abended, and retry manual intervention was requested**

The application has abended. The operations staff has modified the execution control record by setting the Retry Manual Intervention field or the Set Job Abend field to Y on the Active Job Status Modification panel, resulting in the Retry Manual Intervention flag being set.

On restart, BCF takes the following processing steps for this condition:

- BCF issues message BMC44232I (retry manual intervention active).
- BCF executes Batch Backout utility again.
- BCF deletes the execution control record.
**Note**
The Retry Manual Intervention flag must be on before BCF will retry a recovery or backout action marked abended. This flag is set automatically when you set the Set Job Abend field to \textbf{Y} on the Active Job Status Modification panel (see the figure in “Modifying an execution control record” on page 159). If the job was canceled, you must set the Retry Manual Intervention field to \textbf{Y} on the Active Job Status Modification panel.

**Application abended, batch backout abended, and retry manual intervention and defer application restart were requested**

The application has abended. The Log Recovery utility executed successfully. The Batch Backout utility abended.

The operations staff has used the Active Jobstep Display panel (see the figure in “Accessing execution control records” on page 152) to modify the execution control record. The Modify action sets the Retry Manual Intervention flag and the Defer Application Restart flag.

On restart, BCF takes the following processing steps for this condition:

- BCF issues message BMC44232I (retry manual intervention active).
- BCF executes the Batch Backout utility again.
- BCF deletes the execution control record.
- BCF issues message BMC44233I (defer application restart).
- BCF issues abend U1552 to prevent subsequent steps from executing.
- BCF prevents COND=ONLY backout steps from executing.

**Application still active or system failure ended execution**

The application is still active, or a system failure has ended the application execution without notifying BCF.

The execution control record shows the application active, but the example would be the same if any of the three processes (APPLICATION, LOG RECOVERY, or BATCH BACKOUT) were found to be active at restart.

On restart, BCF takes the following processing steps for this condition:

- BCF issues message BMC48061I (invalid restart environment; reason code 1).
- BCF issues abend U1550 to prevent subsequent steps from executing.
- BCF prevents COND=ONLY backout steps from executing.
Note
To prevent inadvertent restart processing for a job step that is still active, BCF requires the execution control record to indicate that no process is active before restart processing can continue. You must set the Set Job Abend field to Y on the Active Job Status Modification panel (see the figure in “Modifying an execution control record” on page 159), specify ABENDED=Y in the Active Job utility and execute it, or execute the Automatic Restart Preparation utility.

Application abended, log recovery and batch backout completed, execution control record was not deleted, and defer application restart was requested

The application has abended. The Log Recovery utility executed successfully. The Automatic Batch Backout utility executed successfully. The system failed, or the job canceled before the execution control record could be deleted.

The operations staff has requested that restart of the application program be deferred by using the Active Job Status Modification panel, the Active Job utility, or the Automatic Restart Preparation utility.

On restart, BCF takes the following processing steps for this condition:

- BCF deletes the execution control record.
- BCF issues message BMC44233I (defer processing complete).
- BCF issues abend U1552 to prevent subsequent steps from executing.
- BCF prevents COND=ONLY backout steps from executing.

Application abended but log recovery and batch backout have not executed

The application has abended. The application may have been active when a system failure occurred. The operations staff, the Active Job utility, or the Automatic Restart Preparation utility has changed the status of the execution control record to "APPLICATION ABENDED."

On restart, BCF takes the following processing steps for this condition:

- BCF performs log recovery if needed.
- BCF performs batch backout if needed.
- BCF deletes the execution control record.
- BCF executes the application.
Application abended and defer application restart was requested

The application has abended. The operations staff, the Active Job utility, or the Automatic Restart Preparation utility has changed the status of the execution control record to "APPLICATION ABENDED."

The operations staff has requested that restart of the application program be deferred by executing the Active Job utility or the Automatic Restart Preparation utility or by setting the Defer Application Restart field to Y on the Active Job Status Modification panel (see the figure in “Modifying an execution control record” on page 159).

On restart, BCF takes the following processing steps for this condition:

- BCF performs log recovery if required.
- BCF performs batch backout if required.
- BCF deletes the execution control record.
- BCF issues message BMC44233I (defer processing complete).
- BCF issues abend U1552 to prevent subsequent steps from executing.
- BCF prevents COND=ONLY backout steps from executing.

Manual intervention

If BCF cannot perform a necessary function or cannot determine which function is required, BCF indicates that manual intervention is required.

BCF requests manual intervention in the following situations:

- Log volume records missing
  BCF found an execution control record in the REGISET, but it could not find the associated log volume record. The execution control record indicates that the batch backout process is necessary before the application program can execute. BCF needs information from the log volume records to identify the logs to use as input to the batch backout process.

- Automatic Batch Backout option set to NO
  BCF found an execution control record in the REGISET, and the Automatic Batch Backout option was specified as NO for current execution.
Critical REGISET error
A critical REGISET error has occurred. BCF found an execution control record in the REGISET, and this record indicates that a critical REGISET error occurred in the previous execution. The REGISET may contain invalid recovery information.

Batch Backout utility failed
Execution of the IMS Batch Backout utility failed during restart processing or the previous execution.

Log Recovery utility failed
Execution of the IMS Log Recovery utility failed during restart processing or the previous execution.

If a situation requires manual intervention, BCF sets the Manual Intervention Required flag (in the execution control record) to Y, issues message BMC44242I, and stops execution of the affected job step. Before BCF will allow you to restart the job step under BCF control, you must reset the Manual Intervention Required flag.

Setting the manual intervention required flag
In a manual intervention situation, perform the following steps:

1. Analyze message BMC44242I.
   This message provides complete information about the process that was running when the error occurred and the reason the message was issued. For an explanation of the reason code, see the APPLICATION RESTART CONTROL messages in the BMC Documentation Center.
   You should also analyze the other messages written to the job log to get further information about the error condition.

2. Determine which actions you should take so that you can restart the job step.

3. Correct the condition that caused the error.

4. Reset the Manual Intervention flag. You must use one of the following methods; the first method is easier:
   - Set the Manual Intervention Override processing option to Y in the global options record, system options record, job step options record, or BCFSYSIN data set. This option allows BCF to reset the flag automatically; you can retry the operation that previously failed without updating the job step execution record in the REGISET. If you already have this option set to Y, you do not need to take any action. For more information, see “Manual Intervention Override” on page 106.
Manually set the Manual Intervention Required flag to Y in the execution control record:

1. Access the Active Jobsteps Display panel (see the figure in “Accessing execution control records” on page 152).

2. Enter action code M to the left of the record you want to update.

3. Set the Retry manual intervention field to Y on the Active Job Status Modification panel (see the figure in “Modifying an execution control record” on page 159).

5. Restart the job step. If the job step fails again, take more corrective action and perform Steps 1 through 4 again.

**Retry manual intervention**

BCF allows you to retry a failed recovery process.

BCF prevents you from restarting a failed job that it has identified as requiring manual intervention. However, after corrective action has been taken, you may want BCF to retry the failed process. In the case of a failed job, BCF allows you to retry a failed recovery process. When you must perform manual intervention for a failure of log recovery or batch backout, you can manually execute the utilities or you can correct the original problem and request that BCF retry the failed process.

During restart, you can perform these actions:

- If you have a job that requires manual intervention but you have taken the necessary corrective action and would like to retry the process, you must turn on the Retry Manual Intervention flag in the execution control record.

- If the process fails again, BCF allows you to repeat this procedure as many times as you want.

You can retry a failed process by specifying the BCFSYSIN keyword for the Manual Intervention Override option (MANintov) or by specifying Y in the Retry manual intervention field on the Active Job Status Modification panel (see the figure in “Modifying an execution control record” on page 159).

**Defer application restart**

The Defer Application Restart flag in the execution control record instructs BCF to perform the steps needed to restore the databases and log data sets to a point of consistency.
These steps can be performed during restart processing after a job step has been canceled or after a system failure. The Defer Application Restart flag can be useful when an application program encounters a problem that is not fixed easily but the databases and logs must be recovered immediately. In such a situation, the Defer Application Restart flag allows you to perform any necessary log recoveries and batch backouts without restarting an application program.

You can set the flag by specifying **Y** in the **Defer application restart** field on the Active Job Status Modification panel (see the figure in “Modifying an execution control record” on page 159). You can also set this flag by specifying the appropriate parameter when you execute the Active Job utility or the Automatic Restart Preparation utility. This action causes BCF to perform only recovery and backout actions.

For more information about the Active Job utility, see “Active Job utility” on page 177. For more information about the Automatic Restart Preparation utility, see “BCF Automatic Restart Preparation utility” on page 180.

**Set job abend**

BCF allows you to update an execution control record to change its status from active to abended.

This action prevents invalid restart processing in environments where the job entry subsystem does not prevent duplicate job name execution. BCF will not allow restart processing for a job step record that appears to be active. You must reset the status before restart processing can occur.

Resetting the status also ensures that any log data sets that should be cataloged and are necessary for restart have been cataloged before restart processing. You can reset the status by specifying **Y** in the **Set job abend** field on the Active Job Status Modification panel (see the figure in “Modifying an execution control record” on page 159). You can also reset the status by specifying the appropriate parameter when you execute the Active Job utility or the Automatic Restart Preparation utility.

For more information about the Active Job utility, see “Use of the Active Job utility” on page 161. If you want all execution control records to be processed following an IPL, you can use the Automatic Restart Preparation utility; see “Automatic Restart Preparation utility” on page 178.

**Active Job utility**

The Active Job utility automates the manual steps required to prepare BCF for application restart after a job or group of jobs has terminated abnormally.

This utility provides the same results as using the Set job abend and Defer application restart fields. It also enables you to list the execution control records
found without changing their status. For more information, see “Use of the Active Job utility” on page 161.

**Automatic Restart Preparation utility**

The Automatic Restart Preparation utility automates the manual steps that are required to prepare BCF for application restart following a system failure.

This utility automatically updates the execution control records, changing all statuses to abended, and catalogs any necessary log volumes needed for restart. For more information, see “BCF Automatic Restart Preparation utility” on page 180.

**REGISET problems**

The REGISET is essential to BCF operations. BCF uses it for a variety of purposes, such as storing the execution control record that contains the status of the job step, storing options, and storing dynamic allocation skeletons.

If a problem occurs with the REGISET, the actions taken by BCF depend on the function that detected the error and on certain execution options that are in effect for the job step execution.

For information about how to correct problems with the REGISET, see the *APPLICATION RESTART CONTROL Administrator Guide*.

When the execution controller detects an error during REGISET processing, it treats the error as an initialization error because it cannot determine the execution options to use.

When the BCF Physical Logger detects an error during REGISET processing, it handles the situation as follows:

- If the execution controller is not active, the error is accepted and BCF makes no further attempt to access the REGISET.

- If the execution controller is active but no database updates have been made, the error is accepted and BCF makes no further attempt to access the REGISET.

- If the execution controller is active and updates have been made but the Automatic Batch Backout option is specified as No, the error is accepted and BCF makes no further attempt to access the REGISET.

- If the execution controller is active and the Automatic Batch Backout option is specified as Yes, the error is accepted. If the record encountering the error is for job statistics, BCF makes no further attempt to access the REGISET.
If the record is one that is critical for automatic batch backout or job step restart, the actions taken depend on whether the BCF Physical Logger can ensure protection against incorrect restart processing. BCF attempts to update the execution control record to reflect the critical error on the REGISET. The actions taken depend on the results of the record update.

— If the record is updated successfully, restart has been protected so the job step is allowed to continue executing in the hope that it will terminate normally. If the application terminates abnormally, automatic batch backout is attempted with in-storage tables. If restart is attempted, manual intervention is requested, unless the Manual Intervention Override option is set to Y.

— If the record is not updated successfully, job step restart can be unpredictable. To protect the restart environment, the execution control record is locked to prevent access by the execution controller. If the execution controller finds a locked record during restart processing, it requests manual intervention for the job step. The job step is allowed to continue executing in the hope that it will terminate normally. If the application terminates abnormally, automatic batch backout is attempted with in-storage tables. This may be successful. If restart is attempted, manual intervention is requested, unless the Manual Intervention Override option is set to Y.

If the execution controller cannot lock the execution control record, it issues an abnormal termination for the job step so that the execution controller can attempt to invoke the IMS Batch Backout utility.

**Note**

Always use DBRC for job step execution. It provides additional protection against restarting a failed job step without first executing the IMS Batch Backout utility.

---

**System problems**

BCF handles most system failures automatically.

**Note**

It is beyond the scope of BCF to back out applications that were executing at the time of a system disaster or when a system snapshot was taken for disaster recovery. For more information, see the *APPLICATION RESTART CONTROL Administrator Guide*.

BCF marks a job step that is executing as **ACTIVE** in the execution control record. While the status is active, BCF does not allow the job step to be restarted. This action prevents problems if a job step is accidently resubmitted while it is still executing. If a system or processor failure occurs during execution of a job step controlled by BCF, BCF may not receive control to mark the job as **ABENDED** or the log data set may
not get cataloged. To ensure that BCF can take correct action at restart, you can change the status of jobs left in active status to **ABENDED**. This process also ensures that any log volumes created during the previous execution are cataloged, if necessary.

Typical situations in which a log may need to be cataloged follow:

- An application was active when the system failed. The IMS output log that was being created may need to be cataloged.

- The Automatic Batch Backout utility was executing when the system failed. The output of batch backout must be cataloged but will not be used during later BCF processing. If the log is not cataloged, additional executions of batch backout can create duplicate log generations. Because DBRC knows the name of the original log data set, recovery may fail when DBRC requests the same log data set twice.

You can catalog all volumes when setting the status of a job to **ABENDED** by using the Active Job utility, by using the Automatic Restart Preparation utility, or by setting the **Set to Abended** field to **Y** on the Active Job Status Modification panel (see the figure in “Modifying an execution control record” on page 159) for individual jobs. If you only want to back out database updates without restarting the application, set the Defer Application Restart flag by using the Active Job utility, by using the Automatic Restart Preparation utility, or by using the Active Job Status Modification panel. For more information about the Automatic Restart Preparation utility, see “BCF Automatic Restart Preparation utility” on page 180. For more information about the Active Job utility, see “Use of the Active Job utility” on page 161.

**BCF Automatic Restart Preparation utility**

The BCF Automatic Restart Preparation utility (program BCFAUTU0) automates the manual steps required to prepare BCF for application restart after a system failure.

BCFAUTU0 automates the process of setting the Set Job Abend option provided on the Active Job Status Modification panel.

You can execute BCFAUTU0 after a system IPL. The utility opens the REGISET and processes all execution control records for the MVS system that you identify in the job step parameters. The utility inspects each record found for the failed MVS system and, if the status of the record is **ACTIVE**, marks the status as **ABENDED**. Because the utility processes only records that display a status of **ACTIVE**, you can repeat the execution of the utility if it encounters any problems.

During processing of each execution control record, the utility catalogs log data sets that need cataloging for restart and notifies DBRC that the job has terminated abnormally. When the log volume record indicates that the log volume should be
cataloged, the utility verifies that the data sets are already cataloged and, if they are not, catalogs the log data sets.

This section describes the utility JCL requirements, reports, and return codes.

**JCL requirements in BCF Automatic Restart Preparation utility**

The figure below shows sample JCL for executing BCFAUTU0.

**Figure 23: BCFAUTU0 sample JCL**

```plaintext
//BCFAUTOR JOB (XXXX), 'user', CLASS=A, MSGCLASS=X, NOTIFY=USER, REGION=4M
//AUTOREST EXEC PGM=BCFAUTU0,REGION=0M,
//       PARM='SSID=BCFQ, SYSTEM=SYSP, ASK=NO,
//          RUN=YES, DBRC=YES'
//STEPLIB DD DISP=SHR, DSN=BMC.BCF.LOAD
//          DD DISP=SHR, DSN=IMSVS.RESLIB
//RECON1 DD DISP=SHR, DSN=IMSVS.RECON1
//RECON2 DD DISP=SHR, DSN=IMSVS.RECON2
//RECON3 DD DISP=SHR, DSN=IMSVS.RECON3
//SYSPRINT DD SYSOUT=* 
//BCFPRINT DD SYSOUT=* 
```

**EXEC statement parameters**

The program to execute is BCFAUTU0.

You can also code parameters to control utility execution. You can code parameters on the EXEC statement to control utility functions. Although the utility requires values for all parameters, it supplies defaults so you can omit them from the JCL.

**SSID**

*(required)* Provide the subsystem ID of the BCSS that you use to access BCF records in the REGISET.

**SYSTEM**

Required if the utility is executing on a different MVS system than the system associated with the execution control records. Code the **SYSTEM** parameter as follows:

```
SYSTEM=sysid
```

where **sysid** is the system ID, also known as the SMFID. It is a one- to four-byte value that corresponds to the MVS system ID of the system associated with the execution control records. The default value is the system ID of the system on which the utility is executing.
ASK

*(required)* If omitted, the default value is NO. Code the ASK parameter as follows:

\[
\text{ASK=value}
\]

where *value* is one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not query the operator during utility execution.</td>
</tr>
<tr>
<td>YES</td>
<td>For each execution control record found, BCFAUTU0 issues message BMC48066A to request operator confirmation before performing Set Job Abended processing. The MVS operator must provide permission for the utility to alter each active entry and, for BCF, to catalog any required log data sets.</td>
</tr>
</tbody>
</table>

RUN

*(required)* If omitted, the default value is YES. Code the RUN parameter as follows:

\[
\text{RUN=value}
\]

where *value* is one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>Leave the value of the Defer Application Restart flag in the execution control record set as found.</td>
</tr>
<tr>
<td>DEFER</td>
<td>Set the Defer Application Restart flag if the record is also marked as abended. When the application is submitted, the Defer Application Restart flag tells BCF to perform backout and log recoveries in preparation for a restart then terminate the job step before the application begins processing. If a record is not a candidate for Set Job Abend processing, the utility does not change the Defer Application Restart flag.</td>
</tr>
</tbody>
</table>

DBRC

*(required)* If omitted, the default value is YES. Code the DBRC parameter as follows:

\[
\text{DBRC=value}
\]

where *value* is one of the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>Notify DBRC that the job has abended if the record is a candidate for Set Job Abend processing.</td>
</tr>
<tr>
<td>No</td>
<td>Do not notify DBRC that the job has abended.</td>
</tr>
</tbody>
</table>
DD statements

Specify the following DD statements:

**STEPLIB DD**

*(required)* Supply the name of the library that contains the BCF load modules. Include the data set name of the IMS RESLIB library when processing execution control records with **DBRC=YES**.

**BCFPRINT DD**

*(required)* Provide a data set to contain the report indicating the actions that the utility performed for BCF records. It usually defines a SYSOUT-type data set, but you can specify a data set on DASD with the following data control block (DCB) parameters: **RECFM=FBA**, **LRECL=132**, and **BLKSIZE=1320**.

**RECONx DD**

*(required when processing execution control records with **DBRC=YES**)* Provide a set of three IMS RECON data sets.

Utility reports

BCFAUTU0 writes two reports to the BCFPRINT data set.

**First BCFPRINT report**

The first report in BCFPRINT lists each execution control record found and the results of any actions taken, including the operator response if **ASK** processing was requested.

The first line indicates the EXEC statement parameters that were in effect for the utility execution.

**Figure 24: BCFAUTU0—first BCFPRINT report**

The report contains the following fields:

**JOBNAME**

The job name associated with the execution control record.
STEPNAME

The step name associated with the execution control record.

PSB/PGM

The PSB name or program name used to qualify the record.

DATE

The date when the active step began execution.

TIME

The time when the active step began execution.

SYS ID

The system ID of the system that executed the job step.

FUNCTION ACTIVE

The status of the step in terms of BCF activity.

OPER RESP

The operator’s response to message BMC48066A if ASK=YES was specified.

MEMBER STATUS

The status of the execution control record.

SET TO ABENDED

An indication of the actions taken by the Automatic Restart Preparation utility for the active job step.

CATALOG PRILOG

An indication of the actions taken by the Automatic Restart Preparation utility to catalog the primary log data set. The following values are valid:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/R</td>
<td>Cataloging the data set is not required.</td>
</tr>
<tr>
<td>LOCATED</td>
<td>The data set was already cataloged.</td>
</tr>
<tr>
<td>RC=xx</td>
<td>An attempt was made to catalog the data set, and the field contains the return code from the catalog function. A return code other than 0 indicates that an error has occurred.</td>
</tr>
</tbody>
</table>
CATALOG SECLOG

An indication of the actions taken by the Automatic Restart Preparation utility to catalog the secondary log data set. The following values are valid:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/R</td>
<td>Cataloging the data set is not required.</td>
</tr>
<tr>
<td>LOCATED</td>
<td>The data set was already cataloged.</td>
</tr>
<tr>
<td>RC=xx</td>
<td>An attempt was made to catalog the data set, and the field contains the return code from the catalog function. A return code other than 0 indicates that an error has occurred.</td>
</tr>
</tbody>
</table>

DB UPT

An indication that the job has updated IMS databases.

DISC LOG

An indication that the job was using the Discard Logging feature of BCF.

DUAL LOG

An indication that the job was using dual logging.

Second BCFPRINT report

The second report in the BCFPRINT data set lists the results of the catalog locate request for the log data set being created at the time of failure and the results of any catalog request made for primary and secondary log data set names (for each job processed).

Figure 25: BCFAUTU0—second BCFPRINT report

<table>
<thead>
<tr>
<th>LOG DSN LISTING FOR ACTIVE JOBS</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBNAME</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>BCF4137U</td>
</tr>
</tbody>
</table>

Utility return and abend codes

The Active Job utility produces the following return and abend codes:

0

The utility ended normally. At least one execution control record was updated, and no abnormal conditions were detected.
The utility ended normally. No execution control record that matched the parameters in the control statements was found in the REGISET.

The utility ended normally. For at least one execution control record, an error occurred when the utility attempted to notify DBRC.

**U1554**

The utility ended abnormally. It issues a U1554 abend code to indicate that utility processing was not possible for one of the following reasons:

- The parameter information provided for the job was missing or invalid. The BCFPRINT data set contains a more detailed error message indicating the cause of the error.

- The utility encountered an error when processing one of the execution control records. The BCFPRINT data set contains a more detailed error message indicating the cause of the error. The abend does not occur until after the utility has attempted to process each selected execution control record.

---

**BCF Purge utility**

To delete BCF records from the REGISET or the BCF history data set, use the program BCFXPURG.

Member #BCFPURG of the sample library (Figure 26 on page 186) contains JCL to run BCFXPURG.

**Figure 26: BCF Purge utility JCL**

```plaintext
//BCFXPURG JOB (ACCT),'BCF V3 PURGE UTIL',CLASS=A,MSGCLASS=X, 
//   NOTIFY=&SYSUID,REGION=4M
//   PURGE EXEC PGM=BCFXPURG,
//     PARM='SSID=BCFQ'                             <== SUBSYSTEM ID
//     */
//STEPLIB DD DSN=BMC.BCF.LOAD,DISP=SHR           <== LOAD LIBRARY
//     */
//BCFHIST DD DISP=SHR,DSN=BMCBCF.QA.VSAM.HIST   <== HISTORY FILE
//     */
//SYSPRINT DD SYSOUT=*                          <== OUTPUT REPORTS
//     */
//SYSIN REPORTS JOB(RRD3105) STARDATE(-9) ENDDATE(-1)
```

The following statements are used in the JCL:
EXEC

(required) The program to execute is BCFXPURG. For the SSID parameter, code the subsystem ID of the BCSS that accesses the REGISET containing the records you want to delete.

STEPLIB DD

(required) Provide the data set name of the library that contains the AR/CTL and BCF component load modules.

BCFHIST DD

(required) Provide the data set name of the BCF history data set that contains the BCF records you want to purge.

SYSPRINT DD

(required) Specify SYSOUT=* or define it as a DASD data set.

SYSIN DD

(required) This control statement data set can contain several types of control statements. You must supply at least one control statement; however, the keywords on the control statements are optional. If you do not specify which items should be purged by supplying keywords and values for reports, options, skeletons, and log volumes, all records containing those types of data will be purged (depending on the control statement).

You can specify the following SYSIN commands to control the utility:

REPORTS

(optional) Purges the specified reports from the history data set. The following keywords are valid; all are optional. If you do not supply them, all statistics are purged.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB</td>
<td>The job name associated with the reports you want to purge.</td>
</tr>
<tr>
<td>STEP</td>
<td>The step name associated with the reports you want to purge.</td>
</tr>
<tr>
<td>PSB</td>
<td>The program specification block (PSB) name associated with the reports you want to purge.</td>
</tr>
<tr>
<td>STARDATE</td>
<td>The beginning date of a range of reports, in mm/dd/yyyy format or -nnnnnn (relative date) format.</td>
</tr>
<tr>
<td>ENDDATE</td>
<td>The ending date of a range of reports, in mm/dd/yyyy format or -nnnnnn (relative date) format.</td>
</tr>
</tbody>
</table>
OPTIONS

(optional) Purges the specified options records from the REGISET. The following keywords are valid; all are optional. If you do not supply them, all statistics are purged.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td>The system or job step level.</td>
</tr>
<tr>
<td>IMSID</td>
<td>The IMSID of the IMS system for which the options apply. Valid only at the system level.</td>
</tr>
<tr>
<td>JOB</td>
<td>The job name associated with the options you want to purge. Valid only at the job step level.</td>
</tr>
<tr>
<td>STEP</td>
<td>The step name associated with the options you want to purge. Valid only at the job step level.</td>
</tr>
<tr>
<td>PSB</td>
<td>The PSB name associated with the options you want to purge. Valid only at the job step level.</td>
</tr>
</tbody>
</table>

SKELS

(optional) Purges the specified skeleton records from the REGISET. The following keywords are valid; all are optional. If you do not supply them, all statistics are purged.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEVEL</td>
<td>The global, system, or job step level.</td>
</tr>
<tr>
<td>DDNAME</td>
<td>The ddname for the skeleton records you want to purge.</td>
</tr>
<tr>
<td>DEVCLASS</td>
<td>The device class for the skeleton records you want to purge. Valid only at the system level.</td>
</tr>
<tr>
<td>IMSID</td>
<td>The IMSID of the IMS system for which the skeleton records apply. Valid only at the system level.</td>
</tr>
<tr>
<td>JOB</td>
<td>The job name associated with the skeleton records you want to purge.</td>
</tr>
<tr>
<td>STEP</td>
<td>The step name associated with the skeleton records you want to purge.</td>
</tr>
<tr>
<td>PSB</td>
<td>The PSB name associated with the skeleton records you want to purge.</td>
</tr>
</tbody>
</table>

LOGVOLx

(optional) Purges the specified log volume records from the history data set. In the ddname, substitute 1 or 2 for x to specify primary log volume records or secondary log volume records. The following keywords are valid; all are optional. If you do not supply them, all primary or secondary log volume records are purged.
<table>
<thead>
<tr>
<th><strong>Keyword</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB</td>
<td>The job name associated with the log volume records you want to purge.</td>
</tr>
<tr>
<td>STEP</td>
<td>The step name associated with the log volume records you want to purge.</td>
</tr>
<tr>
<td>PSB</td>
<td>The PSB name associated with the log volume records you want to purge.</td>
</tr>
<tr>
<td>STARDATE</td>
<td>The beginning date of a range for specified log volume records, in <code>mm/dd/yyyy</code> format or <code>-nnnnn</code> (relative date) format.</td>
</tr>
<tr>
<td>ENDDATE</td>
<td>The ending date of a range for specified log volume records, in <code>mm/dd/yyyy</code> format or <code>-nnnnn</code> (relative date) format.</td>
</tr>
</tbody>
</table>
Using BCF reports

This section describes how to use the statistical and log volume reports that are provided by the BMC Software BATCH CONTROL FACILITY (BCF) component of APPLICATION RESTART CONTROL (AR/CTL).

Overview of BCF reports

This section introduces statistical and log volume reports and how BCF can write and store them.

Report types

During execution, BCF gathers various statistics as it monitors a job.

BCF can save these statistics as reports in the history data set and can write them to the BCFSTATS output data set.

BCF statistical reports help you with system management tasks such as performance measurement and capacity planning. BCF also produces information about the primary and secondary log volumes used during the job. BCF can save the log information in the history data set.

BCF can produce the following types of reports:

- Job Step reports identify the execution of the job step and provide summary information about DL/I call activity, logging performance, checkwrite calls, logging delays, dynamic log usage, and application checkpoints.

- Log Data Set Information reports identify the primary and secondary log data sets.

- The Log Data Set Record report provides details about the records in the log data sets.
- Database Activity reports provide information about the usage of database data sets.

- Database Buffer Pool reports provide information about the database buffers during application program execution.

- Optional DL/I Activity reports provide information about data set IWAITs and Background Write IWAITs.

- Optional DL/I Call reports provide information about the DL/I calls executed during the job step.

- The Execution Options Summary report documents the sources and values of the BCF options that were in effect during execution of the job step.

### Overview of saving statistics

BCF can save statistics in the history data set, write them to the BCFSTATS data set, or both.

**Note**

BCF produces a full set of reports for DLI and DBB programs if you set the DL/I Activity Statistics option to FULL. BCF produces a limited set of reports for ULU programs if the BCFSTATS DD statement is present in the application JCL.

The fields on the reports are the same whether you are viewing reports saved in the history data set or those written to the BCFSTATS data set.

### History data set

The history data set is a VSAM key-sequenced data set (KSDS) that BCF can use to store statistics.

The history data set can hold reports from many application program executions, and you control the number of reports saved for each application program. To save statistics in the history data set, you must allocate the history data set and must specify a number greater than zero for the Number of Statistics Sets option. You can display and print the reports through the BCF ISPF interface.

BCF creates a statistics record in the history data set after it closes the log data set and completes reporting of requested statistics for the job step. BCF can delete old statistics records automatically when the number of statistics records reaches a defined amount. You can delete statistics records manually through the ISPF interface or by executing a batch purge utility.
BCF can write log volume records to the history data set. During execution of the batch job step, BCF keeps log volume records, which contain control information about the logs used during execution, in the REGISET. When the job step terminates normally, BCF deletes the log volume records from the REGISET. If you set the Delete Volume Records option to N, BCF writes the log volume records to the history data set before deleting them from the REGISET.

The History Data Set Name option in the BCF global options record contains the name of the default history data set. You can use a history data set other than the default by setting the History Data Set Name option in a system level record or by including this DD statement in the job step JCL:

```
//BCFHIST DD DSN=history.data.set,DISP=SHR
```

A BCF history data set is usually allocated during BCF configuration. To allocate the history data set, use the sample IDCAMS job in member #BCFHIST of the BCF sample data set.

**BCFSTATS data set**

To write the reports to a data set external to BCF, code the BCFSTATS DD statement in the job step JCL to define the statistics data set.

The BCFSTATS data set is a typical SYSOUT-type data set; you can specify a data set on DASD with the following data control block (DCB) parameters: REC FM=FBA, LRE CL=132.

**Effects of options on the history data set**

BCF processing options have the following effects on the history data set:

- If the Record Activity Stats To option is set to N, BCF writes no history sets and no log volume records to the BCF history data set.

- If the Record Activity Stats To option is set to HIST and the Number of Statistics Sets is set to 0, BCF writes no history sets and no log volume records to the BCF history data set.

- If the Record Activity Stats To option is set to HIST and the Number of Statistics Sets is set to a number greater than 0, BCF writes history sets to the history data set and (if the Delete Volume Records option is set to N ) writes log volume records to the history data set.
Accessing reports with the BCF ISPF interface

You can view the statistics reports and log volume records saved in the history data set through the BCF ISPF interface. To access statistics and log volume records, perform the following steps:

1. Access the BATCH CONTROL FACILITY primary menu.

2. In the choice entry field, type 1 (Display reports) or 2 (Display log volumes by jobname).

   If you enter 2, BCF displays log data set information directly from the log volume records that were stored in the history data set. The information in the log volume records is identical to the information in the Log Data Set Information report in the statistics record. For more information, see “Log data set information” on page 209.

3. Press Enter.

   BCF displays the Report Options panel (Figure 27 on page 194).

4. In the Report sequence field, type one of the following values:

   - 1— List the report sets alphabetically by job name, step name, and program or program specification block (PSB) name.
   - 2— List the report sets by the date and time when the application program executed, with the most recent report set first.
   - 3— List the report sets by the date and time when the application program executed, with the oldest report set first.
5 In the **SYSOUT class** field, type or verify the system output class.

BCF sends the report sets you select for printing to this output class.

6 In the **Copies** field, type or verify the number of copies to print of each report set you select for printing.

7 In the **Destination** field, type the name of the JES node to which BCF should send the report sets you select for printing.

8 In the **Delete confirmation** field, type one of the following values:

- **1**— Display a delete confirmation panel to request confirmation for each report set selected for deletion
- **2**— Delete report sets without requesting confirmation.

9 In the **Sel** field, type **S** to the left of the history data set or data sets that you want to access.

The first row contains a dummy name that you can type over. Other rows may contain history data set names that are identified in system processing options records. The last row contains the history data set name that is identified in the global processing options record.

**Note**

To access reports in a history data set other than the one defined in the BCF global options record or the system processing options records, type the data set name over the dummy name displayed on the first row. Press **Enter** to add the history data set name to the list, then type **S** to the left of that name. This name remains on the list until you enter the end command on this panel.

10 Press **Enter**.

If you chose to display reports in job step order, BCF displays the Report List by Jobstep panel (Figure 28 on page 196). If you chose a different order, BCF
displays a report list panel similar to Figure 28 on page 196; the columns on the panel and the order of the reports on the list depends on the order you chose.

Figure 28: Report List by Jobstep panel

<table>
<thead>
<tr>
<th>Jobname</th>
<th>Stepname</th>
<th>PSB/PGM</th>
<th>Run Date/Time</th>
<th>Process Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCFC3101</td>
<td>BMISRT</td>
<td>DBIHDXX</td>
<td>MON 02/24/1997 17:38</td>
<td>DFSDDLT0</td>
</tr>
<tr>
<td>BCFC3102</td>
<td>DLT0A</td>
<td>DBIHDXX</td>
<td>MON 02/24/1997 17:32</td>
<td>DFSDDLT0</td>
</tr>
<tr>
<td>BCFC3102</td>
<td>DLT0A</td>
<td>DBIHDXX</td>
<td>WED 03/05/1997 15:27</td>
<td>DFSDDLT0</td>
</tr>
<tr>
<td>BCFC3103</td>
<td>DLT0A</td>
<td>DBIHDXX</td>
<td>MON 02/24/1997 17:41</td>
<td>DFSDDLT0</td>
</tr>
<tr>
<td>BCFC3104</td>
<td>DLT0A</td>
<td>DBIHDXX</td>
<td>MON 02/24/1997 17:45</td>
<td>DFSDDLT0</td>
</tr>
<tr>
<td>BCFM3100</td>
<td>DLT0A</td>
<td>DBIHDXX</td>
<td>MON 02/24/1997 17:12</td>
<td>DFSDDLT0</td>
</tr>
<tr>
<td>BCFM3102</td>
<td>RUN1</td>
<td>BA7$ALL</td>
<td>MON 02/24/1997 12:30</td>
<td>BATCH BACKOUT</td>
</tr>
<tr>
<td>BCFM3102</td>
<td>RUN1</td>
<td>BA7$ALL</td>
<td>MON 02/24/1997 17:23</td>
<td>BATCH BACKOUT</td>
</tr>
<tr>
<td>BCFM3102</td>
<td>RUN1</td>
<td>BA7$ALL</td>
<td>TUE 02/25/1997 08:05</td>
<td>BATCH BACKOUT</td>
</tr>
<tr>
<td>BCFM3104</td>
<td>RUN1</td>
<td>BA7$ALL</td>
<td>FRI 02/28/1997 10:59</td>
<td>BATCH BACKOUT</td>
</tr>
<tr>
<td>BCFM3104</td>
<td>RUN1</td>
<td>BA7$ALL</td>
<td>FRI 02/28/1997 13:36</td>
<td>BATCH BACKOUT</td>
</tr>
<tr>
<td>BCFM3104</td>
<td>RUN1</td>
<td>BA7$ALL</td>
<td>FRI 02/28/1997 18:16</td>
<td>BATCH BACKOUT</td>
</tr>
<tr>
<td>BCFM3104</td>
<td>RUN1</td>
<td>BA7$ALL</td>
<td>FRI 02/28/1997 18:41</td>
<td>BATCH BACKOUT</td>
</tr>
<tr>
<td>BCFM3110</td>
<td>DLT0A</td>
<td>DBIHDXX</td>
<td>FRI 02/21/1997 11:08</td>
<td>DFSDDLT0</td>
</tr>
<tr>
<td>BCFM3110</td>
<td>DLT0A</td>
<td>DBIHDXX</td>
<td>FRI 02/21/1997 14:54</td>
<td>DFSDDLT0</td>
</tr>
<tr>
<td>BCFM3110</td>
<td>DLT0A</td>
<td>DBIHDXX</td>
<td>FRI 02/21/1997 16:41</td>
<td>DFSDDLT0</td>
</tr>
</tbody>
</table>

11 In the Sel field to the left of the report set you want, type one of the following action codes:

- **D**—Delete the report set.
- **P**—Print the report set.
- **S**—Display the report set.

12 Press Enter.

- If you selected a report set to delete and if you requested delete confirmation on the Report Options panel, BCF displays a panel to confirm the deletion. (For generic information about this panel, see “Deleting a record” on page 52.) If you did not request delete confirmation, BCF deletes the report set.

- If you selected a report set to print, BCF prints the report set with the printing options you set on the Report Options panel.

- If you selected a report set for display, BCF creates a temporary data set, writes the report set to it, and invokes the ISPF Browse function to display it (Figure 29 on page 197). When you exit this panel, BCF deletes the temporary data set.

BCF formats the report set for display within the 80-column display area defined for most display devices. You can use the normal ISPF Browse commands (such as UP, DOWN, FIND, and RFIND) to scroll through this data set and find information of interest.
To scroll through the report set by logical page, turn on the display of carriage control characters and display the next logical page by entering the following commands:

**DISPLAY CC**

**FIND 'T' 1**

Figure 29: ISPF BROWSE panel displaying selected report

Jobstep reports

Jobstep reports contain job-related information.

BCF can write these reports to the history data set. If the BCFSTATS DD statement is included in the application program JCL, BCF writes the statistics to the BCFSTATS data set.

The Jobstep reports include the following reports:

- Jobstep Summary
- DL/I Call Summary
- Checkwrite Statistics
- Logging Delay Detail
- Dynamic Log Usage Detail
- Application Checkpoint Detail
- Time Between Checkpoints
- Time Between Checkpoints Distribution
- # DL/I Calls Between Checkpoints
- # DL/I Calls Between Checkpoints Distribution
Jobstep Summary

The Jobstep Summary report provides information about job step execution.

**Figure 30: Jobstep Summary report**

<table>
<thead>
<tr>
<th>JOBNAME=BCFV3150</th>
<th>PROCSTEPNAME=DLTOD</th>
<th>STEPNAME=DLTOD</th>
<th>SYSID=SYSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSBNAME=DBIHDXX</td>
<td>PROGRAM NAME=DFSDDLT0</td>
<td>DBRC=YES</td>
<td>IRLM=NO</td>
</tr>
<tr>
<td>START DATE=1997064</td>
<td>START TIME=14.33.41.45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOP DATE=1997064</td>
<td>STOP TIME=14.34.37.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMS LOGS CLOSED BY ESTAE DL/I ABEND CODE=NONE</td>
<td>TASK ABEND CODE=0252</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Jobstep Summary report includes the following information:

**JOBNAME**

The job name associated with the execution.

**PROCSTEPNAME**

If the job step has referenced a JCL procedure, this field is the step name contained in the JCL procedure. If the job step is not using a JCL procedure, this field is blank.

**STEPNAME**

The step name for the executing step.

**SYSID**

The system ID (SYSID) of the MVS system where the job step executed. If the active job step execution record is for a level of BCF that does not provide the SYSID information, this field contains blanks.

**PSBNAME**

The PSB name specified in the job step parameters used to invoke IMS.

**PROGRAM NAME**

The program named in the job step parameters used to invoke IMS.
DBRC

Whether DBRC is active for the job step.

IRLM

Whether IRLM is active for the job step.

START DATE

The Julian date of the start of the batch program. This date can vary from the actual date that the job step started execution. It matches the IMS accounting record on the IMS log data set produced for the execution.

START TIME

The start time of the batch program execution. This time can vary from the actual start time for the job step. It matches the IMS accounting record on the IMS log data set produced for the execution.

STOP DATE

The Julian date of the end of the program. This date can vary from the actual date that the job step stopped execution. It matches the IMS accounting record on the IMS log data set produced for the execution.

STOP TIME

The stop time of the end-of-program execution. This time can vary from the actual start time for the job step. It matches the IMS accounting record on the IMS log data set produced for the execution.

MESSAGES

If the job step required additional processing, this field indicates that this processing occurred. The following messages can appear:

- DYNAMIC BACKOUT WAS PERFORMED FOR THIS JOBSTEP
- DYNAMIC BACKOUT FAILED FOR THIS JOBSTEP
- IMS LOGS CLOSED BY ESTAE
- DL/I ABEND CODE=NONE
- TASK ABEND CODE=nnnn

**DL/I Call Summary**

The DL/I Call Summary report provides a summary of the DL/I calls issued by the application program during execution.
The report displays the number of times each type of DL/I call was made by the application program during current job execution.

**Note**
During batch backout, this report may contain values of zero because BCF does not issue DL/I calls during backout.

---

**Figure 31: DL/I Call Summary report**

```
+-------------------+
≠ DL/I CALL SUMMARY ≠
+-------------------+
GU..................0  GN..................0  GNP..................0
GHU..............752  GHN..................0  GHNP...............0
ISRT...........7,380  DLET.............329  REPL.............423
TOTAL=..........8,884
```

The total number of DL/I calls appears at the bottom of the report. Refer to the IBM *IMS Application Programming Reference Manual* for call types and descriptions of particular calls.

---

**Logging Performance Data**

The Logging Performance Data report displays performance information about the IMS log data set.

**Figure 32: Logging Performance Data report**

```
+--------------------------+
≠ LOGGING PERFORMANCE DATA ≠
+--------------------------+
TOTAL BYTES LOG DATA.............170,409
BLKS WRITTEN............120  MAX BLK.12,332  MIN BLK.....82
```

The Logging Performance Data report includes the following information:

**TOTAL BYTES LOG DATA**

The total byte count written to the log data set. If the Dual Logging option is set to Y, this field reports the amount of data written to the primary log data set.

**BLKS WRITTEN**

The number of blocks written to the log data set.

**MAX BLK**

The size of the largest block written to the log data set.
MIN BLK

The size of the shortest block written to the log data set. This block is normally the first block written because it contains the log header and accounting records. This block is forced out during log open and initialization.

Checkwrite Statistics

IMS uses a special logging call known as a checkwrite call to ensure that all log records necessary for database integrity are written to the log data sets.

This call tells the BCF Physical Logger to ensure that a log block is physically written to the log device before continuing execution. The checkwrite call also replenishes log buffers for the IMS Logical Logger.

The Checkwrite Statistics report provides details about the number of checkwrite calls and the number and reason for the calls.

Figure 33: Checkwrite Statistics report

<table>
<thead>
<tr>
<th>CHECKWRITE STATISTICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL..........120   WRITE AHEAD............119   GET BUFFER............1</td>
</tr>
</tbody>
</table>

The Checkwrite Statistics report includes the following information:

TOTAL

The total number of checkwrite calls issued during the job step.

WRITE AHEAD

The number of checkwrite calls issued by IMS to provide database integrity. The IMS Logical Logger issues the checkwrite calls to ensure that the requested block is physically written before allowing IMS to continue execution. IMS DL/I batch processing issues checkwrite calls primarily when writing an altered database buffer and during application checkpoint processing.

GET BUFFER

The number of GET BUFFER calls issued during the job step. The GET BUFFER call is the normal way the IMS Logical Logger replenishes log buffers. When the IMS Logical Logger finds no available log buffers, it issues a checkwrite call to the IMS Physical Logger to have the oldest buffers checked for completion and returned to the available log buffer queue. The BCF Physical
Logger tries to reduce the number of times log buffers are unavailable by returning buffers to the available log buffer queue as soon as the write operation is completed.

Logging Delay Detail

The Logging Delay Detail report gives information about the delays the job step experiences because of the logging environment. All timings are based on elapsed time (in milliseconds).

Figure 34: Logging Delay Detail report

<table>
<thead>
<tr>
<th>OCCURRENCES</th>
<th>TIME IN MILLISECONDS</th>
</tr>
</thead>
<tbody>
<tr>
<td># BUFFER WAITS.................236</td>
<td>BUFFER WAIT TOTAL..............391</td>
</tr>
<tr>
<td># SYNCS BUFFERED DEVICE.............0</td>
<td>SYNC WAIT TOTAL.................0</td>
</tr>
<tr>
<td># QUERY BUFFERED DEVICE.............0</td>
<td>QUERY WAIT TOTAL.................0</td>
</tr>
<tr>
<td># READ LOG BUFFER WAITS.............0</td>
<td>READ LOG WAIT TOTAL.............0</td>
</tr>
</tbody>
</table>

The Logging Delay Detail report includes the following information:

# BUFFER WAITS

The number of times that the BCF Physical Logger was forced to wait for the write operation of a log buffer to complete. Normally the BCF Physical Logger waits for the write operation to complete only if the buffer is needed for buffer replenishment or if doing checkwrite processing for database integrity.

BUFFER WAIT TOTAL

The total time that the BCF Physical Logger waited for the logging device to complete an I/O operation.

# SYNCS BUFFERED DEVICE

The number of times that the BCF Physical Logger invoked the routine which provides physical synchronization for the log data set. Synchronization is necessary to provide for database integrity when tape logs are being written with buffered device support.

SYNC WAIT TOTAL

The total time used for accomplishing log data set synchronization. This time may include the processing time for certain operating system macros used during log data set synchronization.
# QUERY BUFFERED DEVICES

The number of buffered devices that the BCF Physical Logger queried.

**QUERY WAIT TOTAL**

The total time that the BCF Physical Logger waited for the logging device to perform a query.

# READ LOG BUFFER WAITS

If IMS requires reading the log data set for dynamic backout, the number of times that the BCF Physical Logger waited for a read operation to complete.

**READ LOG WAIT TOTAL**

The total time that the BCF Physical Logger waited for read I/O operations to complete.

## Dynamic Log Usage Detail

The Dynamic Log Usage Detail report provides the details of dynamic log usage and any delays experienced with using a dynamic log to provide IMS dynamic batch backout.

**Figure 35: Dynamic Log Usage Detail report**

<table>
<thead>
<tr>
<th></th>
<th># DYNAMIC LOG USAGE DETAIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIZE IN BLOCKS</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>2,948,400</td>
</tr>
<tr>
<td>LOG TYPE</td>
<td>DATASPACE</td>
</tr>
<tr>
<td>LOG BLOCKSIZE</td>
<td>32,760</td>
</tr>
<tr>
<td># LOGICAL WRITS</td>
<td>182</td>
</tr>
<tr>
<td># LOGICAL READS</td>
<td>193</td>
</tr>
<tr>
<td># TIMES SUSPENDED</td>
<td>2</td>
</tr>
<tr>
<td>READ/WRT DELAY(MIC)</td>
<td>12,591</td>
</tr>
</tbody>
</table>

The Dynamic Log Usage Detail report includes the following information:

**SIZE IN BLOCKS**

The size of the dynamic log, in blocks.

**SIZE IN BYTES**

The size of the dynamic log, in bytes.

**LOG TYPE**

The type of dynamic log data set used. The following values are valid:
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMORY</td>
<td>The dynamic log is located in extended private memory.</td>
</tr>
<tr>
<td>DATASPACE</td>
<td>The dynamic log is located in a data space provided by MVS/ESA.</td>
</tr>
</tbody>
</table>

**LOG BLOCKSIZE**

The size of each block of the dynamic log data set.

**# LOGICAL WRTS**

The number of times a log block was logically written to the dynamic log data set.

**# LOGICAL READS**

The number of times a log block was logically read from the dynamic log data set for IMS dynamic backout.

**# TIMES SUSPENDED**

The number of times logical write operations to the dynamic log data set were suspended because the dynamic log data set was full. (Logical writing is suspended when the log is full.) If the application issues a checkpoint, the dynamic log logical write function resumes until the log again becomes full.

**READ/WRT DELAY (MIC)**

The elapsed time of logical read and write operations to the dynamic log, in microseconds.

### Application Checkpoint Detail

BCF produces the Application Checkpoint Detail report to provide information about the application checkpoints performed during application program execution.

If no application checkpoints were performed, BCF omits this report.

**Figure 36: Application Checkpoint Detail report**

<table>
<thead>
<tr>
<th># CHECKPOINTS</th>
<th># LOG BLOCKS</th>
<th># BLKS/CKPT</th>
<th>LOG BYTES/CHECKPOINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>120</td>
<td>30.00</td>
<td>42,602</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LAST CKPT ID</th>
<th>LOG BLOCK #</th>
<th>PRIMARY LOG VOLUME</th>
<th>SECONDARY LOG VOLUME</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCF0002</td>
<td>64</td>
<td>DEV230</td>
<td>DEV050</td>
</tr>
</tbody>
</table>
The Application Checkpoint Detail report includes the following information:

# CHECKPOINTS

The number of checkpoint intervals for the application. The number of intervals is equal to the number of checkpoints performed plus end-of-job processing.

# LOG BLOCKS

The total number of blocks written to the IMS log data set.

# BLKS/CKPT

The average number of blocks written to the IMS log data set between checkpoints. This value is calculated by dividing the total number of blocks by the total number of checkpoints.

LOG BYTES/CHECKPOINT

The average amount of log data, expressed in bytes, written to the IMS log data set between checkpoints. This value is calculated by dividing the total number of bytes in the log by the total number of checkpoints.

LAST CKPT ID

The checkpoint ID of the last checkpoint performed.

LOG BLOCK #

The number of the log block that contains the last checkpoint. This number is relative to the beginning of the log data set, where the first log block is number 1.

PRIMARY LOG VOLUME

The volume serial number (VOLSER) of the primary log volume that contains the last checkpoint.

SECONDARY LOG VOLUME

The volume serial number (VOLSER) of the secondary log volume that contains the last checkpoint.

**Time Between Checkpoints**

BCF produces the Time Between Checkpoints (SEC) report to provide information about the time between checkpoints performed during application program execution.
If no application checkpoints were performed, BCF omits this report.

Because end-of-job is considered the final application checkpoint, the interval covered between the last checkpoint performed and the end of the job may include some IMS processing. For example, if IMS dynamic backout is invoked for the job step, the time spent performing dynamic backout is included in the interval.

**Figure 37: Time Between Checkpoints (SEC) report**

<table>
<thead>
<tr>
<th>AVG</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.945</td>
<td>1.894</td>
<td>39.470</td>
</tr>
</tbody>
</table>

The Time Between Checkpoints report includes the following information:

**AVG**

The average time between checkpoints, in seconds.

**MIN**

The minimum time between checkpoints, in seconds.

**MAX**

The maximum time between checkpoints, in seconds.

**Time Between Checkpoints Distribution**

BCF produces the Time Between Checkpoints (SEC) Distribution report to provide information about the time between checkpoints performed during application program execution.

If no application checkpoints were performed, BCF omits this report.

**Figure 38: Time Between Checkpoints (SEC) Distribution report**

<table>
<thead>
<tr>
<th>&lt; 1 SEC</th>
<th>1 - 15 SEC</th>
<th>15 - 60 SEC</th>
<th>60 - 300 SEC</th>
<th>&gt; 300 SEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The Time Between Checkpoints Distribution report includes the following information:
<1 SEC

The number of checkpoint intervals less than 1 second.

1 - 15 SEC

The number of checkpoint intervals between 1 and 15 seconds.

15 - 60 SEC

The number of checkpoint intervals between 15 and 60 seconds.

60 - 300 SEC

The number of checkpoint intervals between 60 and 300 seconds.

> 300 SEC

The number of checkpoint intervals over 300 seconds.

Number of DL/I Calls Between Checkpoints

BCF produces the Number of DL/I Calls Between Checkpoints report to provide information about the number of DL/I calls that the application program issued between the checkpoints performed during application program execution.

If no application checkpoints were performed, BCF omits this report.

Figure 39: Number of DL/I Calls Between Checkpoints report

+------------------------------------------+
≠                                  ≠
NUMBER OF DL/I CALLS BETWEEN CHECKPOINTS ≠
+------------------------------------------+

AVG | MIN | MAX
--------------   --------------   --------------
2,221.00            0              8,694

The Number of DL/I Calls Between Checkpoints report includes the following information:

AVG

The average number of DL/I calls issued by the application program between checkpoints.

MIN

The minimum number of DL/I calls issued by the application program between checkpoints.
The maximum number of DL/I calls issued by the application program between checkpoints.

**Number of DL/I Calls Between Checkpoints Distribution**

BCF produces the Number of DL/I Calls Between Checkpoints Distribution report to provide information about the number of DL/I calls that the application program issued between the checkpoints performed during application program execution.

If no application checkpoints were performed, BCF omits this report.

*Figure 40: Number of DL/I Calls Between Checkpoints Distribution report*

<table>
<thead>
<tr>
<th>NUMBER OF DL/I CALLS BETWEEN CHECKPOINTS DISTRIBUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 50 CALLS</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

The Number of DL/I Calls Between Checkpoints Distribution report includes the following information:

**< 50 CALLS**

The number of checkpoint intervals where the number of DL/I calls issued between checkpoints is less than 50.

**50-100 CALLS**

The number of checkpoint intervals where the number of DL/I calls issued between checkpoints is between 50 and 100.

**100-500 CALLS**

The number of checkpoint intervals where the number of DL/I calls issued between checkpoints is between 100 and 500.

**500-1000 CALLS**

The number of checkpoint intervals where the number of DL/I calls issued between checkpoints is between 500 and 1000.

**> 1000 CALLS**

The number of checkpoint intervals where the number of DL/I calls issued between checkpoints is more than 1000.
Log data set information

The Log Data Set Information section of the report set provides log data set usage and volume information.

To display this information directly from the log volume records, enter 2 instead of 1 on the BATCH CONTROL FACILITY primary menu.

Primary Log

The Primary Log report provides log data set usage and volume information.

If DBRC is active, the start and stop times and volume information should be the same as recorded in the RECON data sets for job step execution.

Figure 41: Primary Log report

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>BLOCKSIZE</th>
<th>BUFNO</th>
<th>DEVICE TYPE</th>
<th>INTEGRITY MODE</th>
<th>OPTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEFRDER</td>
<td>32,760</td>
<td>5</td>
<td>DASD</td>
<td>IMS STANDARD</td>
<td>N/A</td>
</tr>
</tbody>
</table>

DATA SET NAME
dl1.bcf.gdx.iefrder.g3702v00

<table>
<thead>
<tr>
<th>VOLUME # VOLSER START TIME</th>
<th>STOP TIME</th>
<th>LAST BLOCK</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 DEV230 1997064/14.33.42.3</td>
<td>1997064/14.34.37.8</td>
<td>120</td>
<td>OK</td>
</tr>
</tbody>
</table>

The Primary Log report includes the following information:

**DDNAME**

The ddname associated with this log data set.

**BLOCKSIZE**

The log data set block size.

**BUFNO**

The number of buffers used for log data set creation.

**DEVICE TYPE**

The generic device type used for the log data set.
INTEGRITY MODE

The actions that the BCF Physical Logger took to provide physical log data set integrity. The following values are valid:

- BCF BUFFERED
- IMS STANDARD
- TAPE WRITE IMMED

DATA SET NAME

The log data set name.

VOLUME #

The volume sequence number for the log data set.

VOLSER

The volume serial number for this volume.

START TIME

The Julian date and start time of the log data set volume. This time matches the time in the RECON data set if DBRC is active.

STOP TIME

The Julian date and stop time of the log data set volume. This time matches the time in the RECON data set if DBRC is active. If zeros appear under this column, the log data set has not been closed.

LAST BLOCK

The last block written to this log volume.

STATUS

The current status of the volume. The following values are valid:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>No errors have occurred for the volume.</td>
</tr>
<tr>
<td>ERROR</td>
<td>One or more errors have been detected.</td>
</tr>
</tbody>
</table>
Secondary Log

The Secondary Log report (not shown) provides the log data set usage and volume information for the secondary log data set.

Dual logging must be used for this report to be created. If DBRC is active, the times and volume information should be the same as the times and volume information recorded in the RECON data sets for the job step execution.

The fields on the Secondary Log report are the same as the fields on the Primary Log report.

Log Data Set Record Detail report

The Log Data Set Record Detail report provides detailed information about the number and types of log records recorded in the IMS log data set.

Figure 42: Log Data Set Record Detail report

<table>
<thead>
<tr>
<th>RECORD TYPE</th>
<th>RECORD COUNT</th>
<th>RECORD BYTE COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>06</td>
<td>2</td>
<td>64</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>108</td>
</tr>
<tr>
<td>41</td>
<td>3</td>
<td>234</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>52</td>
</tr>
<tr>
<td>48</td>
<td>120</td>
<td>5,520</td>
</tr>
<tr>
<td>5050</td>
<td>1,136</td>
<td>161,182</td>
</tr>
</tbody>
</table>

The Log Data Set Record Detail report includes the following information:

RECORD TYPE

The record identifier in the IMS log record, reported in hexadecimal format.

RECORD COUNT

The physical number of each type of log record encountered.

RECORD BYTE COUNT

The total number of bytes for the specific record type.
Database Activity reports

Database Activity Reports provide information about each database data set accessed or updated by the application program. The following reports are included:
- Jobstep Database Data Set Detail
- Jobstep Database Detail

Jobstep Database Data Set Detail

The Jobstep Database Data Set Detail report provides the physical database data set characteristics for each database data set.

**Figure 43: Jobstep Database Data Set Detail report**

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>BYTES LOGGED</th>
<th># LOG RECORDS</th>
<th>CI/BLK SIZE</th>
<th>INDEX CI SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBIHD01</td>
<td>153,414</td>
<td>1,085</td>
<td>4,096</td>
<td>N/A</td>
</tr>
<tr>
<td>DBIIX01</td>
<td>10,537</td>
<td>69</td>
<td>4,096</td>
<td>4,096</td>
</tr>
</tbody>
</table>

The Jobstep Database Data Set Detail report includes the following information:

**DDNAME**

The ddname associated with the database data set.

**BYTES LOGGED**

The total number of bytes logged for this database data set.

**# LOG RECORDS**

The total number of log records produced for this database data set.

**CI/BLK SIZE**

The control interval (CI) or block size defined for this database data set.

**INDEX CI SIZE**

The control interval size of the index component if this database data set has an index component. Otherwise, the value is specified as N/A (not applicable).
Jobstep Database Detail

The Jobstep Database Detail report provides details about the database.

Figure 44: Jobstep Database Detail report

<table>
<thead>
<tr>
<th>DBDNAME</th>
<th>DDNAME</th>
<th>DSORG</th>
<th>REQUESTED ACCESS</th>
<th>AUTHORIZATION STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBIHD01</td>
<td>DBIHD01</td>
<td>HDAM VSAM</td>
<td>UPDATE</td>
<td>SINGLE UPDATE</td>
</tr>
<tr>
<td>DBIIIX02</td>
<td></td>
<td>INDEX VSAM</td>
<td></td>
<td>SHARED INDEX</td>
</tr>
<tr>
<td>DBIIIX03</td>
<td></td>
<td>INDEX VSAM</td>
<td></td>
<td>SHARED INDEX</td>
</tr>
<tr>
<td>DBIIIX01</td>
<td>DBIIIX01</td>
<td>INDEX VSAM</td>
<td>UPDATE</td>
<td>SINGLE UPDATE</td>
</tr>
</tbody>
</table>

The Jobstep Database Detail report includes the following information:

DBDNAME

The DBD name of the database.

DDNAME

The ddname of the database data set.

DSORG

The data set organization of the database data set.

REQUESTED ACCESS

The access type required by the job step for this database data set. This value is determined by the processing option (PROCOPT) defined in the database program communication block (PCB).

AUTHORIZATION STATUS

The DBRC authorization status of the database data set. The following DBRC authorization status messages can be issued. The "NOT REGISTERED" message can be issued because the database data set is not registered or because DBRC is not active.

- READ-ONLY
- READ SHARE
- MULTIPLE UPDATE
- BATCH UPDATE
- READ EXCLUSIVE
- SINGLE UPDATE
- EXCLUSIVE
- SHARED INDEX
- NOT AUTHORIZED
Database Buffer Pool reports

The Database Buffer Pool report provides details on the usage of the database buffer pools. This report is produced only when BCF detects buffer pool activity.

**OSAM report**

The OSAM Section of the Database Buffer Pool report is produced only for OSAM databases.

The format is similar to IMS DB monitor reports or buffer pool reports that are produced when the DFSSTATS DD statement is included in the job step JCL.

The Database Buffer Pool report includes the following information:

**POOLID**

The $poolid$ specified in the buffer pool definition statement.
BUFFER SIZE

The size of each buffer in this subpool.

FIXED COMPONENTS: PREFIX/BUFFERS

The components of the buffer pool that were page-fixed according to the buffer pool definition statement. A value of N indicates the item is not page-fixed and a value of Y indicates the item is page-fixed.

BUFFERS IN SUBPOOL

The total number of buffers in this subpool.

LOCATE CALLS

The number of times that the IMS OSAM request handler issued an ALLOCATE request to obtain buffer ownership for a buffer in this buffer subpool.

REQUESTS TO CREATE NEW BLOCK

The number of times that the IMS OSAM request handler created a new block for a database assigned to this subpool. The database block has been added to the end of the database.

# BUFFER ALTERATIONS RECEIVED

The number of times that a buffer in the buffer pool was marked altered by database management routines.

# PURGE USER REQUESTS

The number of times that a purge user action was requested.

LOCATE CALLS SATISFIED FROM POOL

The number of times that the buffer requested by a LOCATE call was found in the subpool.

BUFFERS SEARCHED BY LOCATE CALLS

The number of buffers that were searched in an attempt to satisfy all LOCATE type requests. Buffers are normally searched with a hashing technique. If more buffers are searched than LOCATE calls issued, some synonym is being produced by the hashing technique. You have no control over this technique.

READS I/O REQUESTS

The number of times that a read I/O was issued because the RBA referenced by a locate request was not in the subpool.
WRITES TO STEAL BUFFER

The number of times that it was necessary to steal an altered buffer. Before the buffer could be stolen, it had to be written to the database.

# BLOCKS WRITTEN BY PURGE

The number of database blocks written to the database during an OSAM purge operation. Normally, blocks written by purge are sorted and written to the database one track at a time.

# WAITS FOR LOCATE, ID BUSY

The number of times that a task had to wait because LOCATE determined that the subpool or buffer latch was held by another task.

# WAITS FOR LOCATE, WRITE BUSY

The number of times that a task had to wait because LOCATE determined that the buffer was being written by another task.

# WAITS FOR LOCATE, READ BUSY

The number of times that a task had to wait because LOCATE determined that the buffer was being read by another task.

# STEAL/PURGE REQUESTS WAITING BUF REL

The number of times that a task had to wait because a buffer chosen for a steal or purge operation was owned by another task. This condition should not occur unless an unreasonably low number of buffers are in the subpool.

# STEAL REQUESTS WAITING FOR BUFFERS

The number of times that a task had to wait because a buffer chosen for a steal operation was owned by another task. This condition should not occur unless an unreasonably low number of buffers are in the subpool.

# I/O ERRORS IN SUBPOOL

The number of times that an I/O error was logged for this subpool. The block in error has been moved to a write error buffer, and the error has been noted in DBRC (if the database is registered).

# BUFFERS LOCKED DUE TO ERRORS

The number of times that a write error occurred and IMS could not obtain a write error buffer. This condition is abnormal because error buffers are not locked in the subpool unless a write error buffer could not be obtained.
VSAM report

The format of the VSAM Section of the Database Buffer Pool report depends on the IMS version, release, and maintenance level of the executing system.

The report contains detailed information for a given subpool only if BCF detects activity for that subpool.

A summary of all subpools appears in the final report.

Figure 46: Database Buffer Pool report—VSAM section

<table>
<thead>
<tr>
<th>VSAM BUFFER POOL DETAIL</th>
<th>BUFFER SIZE=...........512</th>
</tr>
</thead>
<tbody>
<tr>
<td>POOLID=NONE</td>
<td>FIXED COMPONENTS: INDEX/BLOCK/DATA=N/N/N</td>
</tr>
<tr>
<td>SHARED RESOURCE POOL TYPE.........................................DATA</td>
<td></td>
</tr>
<tr>
<td>NO ACTIVITY DETECTED</td>
<td>TOTAL BUFFERS THIS SUBPOOL=.............5</td>
</tr>
<tr>
<td>VSAM BUFFER POOL DETAIL</td>
<td>BUFFER SIZE=...........1,024</td>
</tr>
<tr>
<td>POOLID=NONE</td>
<td>FIXED COMPONENTS: INDEX/BLOCK/DATA=N/N/N</td>
</tr>
<tr>
<td>SHARED RESOURCE POOL TYPE.........................................DATA</td>
<td></td>
</tr>
<tr>
<td>NO ACTIVITY DETECTED</td>
<td>TOTAL BUFFERS THIS SUBPOOL=.............5</td>
</tr>
<tr>
<td>VSAM BUFFER POOL DETAIL</td>
<td>BUFFER SIZE=...........2,048</td>
</tr>
<tr>
<td>POOLID=NONE</td>
<td>FIXED COMPONENTS: INDEX/BLOCK/DATA=N/N/N</td>
</tr>
<tr>
<td>SHARED RESOURCE POOL TYPE.........................................DATA</td>
<td></td>
</tr>
<tr>
<td>NO ACTIVITY DETECTED</td>
<td>TOTAL BUFFERS THIS SUBPOOL=.............5</td>
</tr>
<tr>
<td>VSAM BUFFER POOL DETAIL</td>
<td>BUFFER SIZE=...........4,096</td>
</tr>
<tr>
<td>POOLID=NONE</td>
<td>FIXED COMPONENTS: INDEX/BLOCK/DATA=N/N/N</td>
</tr>
<tr>
<td>SHARED RESOURCE POOL TYPE.........................................DATA</td>
<td></td>
</tr>
<tr>
<td># BUFFERS IN SUBPOOL</td>
<td># BUFFERS IN HIPERSPACE</td>
</tr>
<tr>
<td># TOTAL BUFFERS IN SUBPOOL=..................5</td>
<td># TOTAL BUFFERS IN SUBPOOL=..................0</td>
</tr>
<tr>
<td># RETRIEVE BY RBA CALLS...</td>
<td># RETRIEVE BY KEY CALLS ...% 27,722</td>
</tr>
<tr>
<td># LOGICAL RECORDS INSERTED INTO ESDS</td>
<td># LOGICAL RECORDS INSERTED INTO KSDS ...% 0</td>
</tr>
<tr>
<td># LOGICAL RECORDS ALTERED THIS SUBPOOL</td>
<td># TIMES BACKGROUND WRITE INVOKED ...% 793</td>
</tr>
<tr>
<td># TIMES SYNCHRONIZATION CALLS RECEIVED</td>
<td># VSAM GET CALLS ISSUED ...% 17,973</td>
</tr>
<tr>
<td># VSAM SCHBF CURRENTS</td>
<td># VSAM SCHBF CURRENTS ...% 1,400</td>
</tr>
<tr>
<td># TIMES CONT INT ALREADY IN POOL</td>
<td># CONT INT READ FROM EXTERNAL STORAGE ...% 10,286</td>
</tr>
<tr>
<td># VSAM WRITES INITIATED BY IMS</td>
<td># VSAM WRITES TO MAKE SPACE IN POOL ...% 9,328</td>
</tr>
<tr>
<td># VSAM WRITES TO MAKE SPACE IN POOL</td>
<td># VSAM WRITES INITIATED BY IMS ...% 0.235</td>
</tr>
<tr>
<td># PERM WRT ERRORS BUFFS NOW IN SUBPOOL</td>
<td># PERM WRT ERRORS BUFFS NOW IN SUBPOOL ...% 0.220</td>
</tr>
<tr>
<td>LARGEST # PERM ERR BUFFS EVER IN SUBPOOL</td>
<td># VSAM READS FROM HIPERSPACE BUFFERS ...% 0.0</td>
</tr>
<tr>
<td># VSAM WRITES TO HIPERSPACE BUFFERS</td>
<td># VSAM WRITES TO HIPERSPACE BUFFERS ...% 0.0</td>
</tr>
<tr>
<td># FAILED READS FROM HIPERSPACE BUFFERS</td>
<td># FAILED READS FROM HIPERSPACE BUFFERS ...% 0.0</td>
</tr>
<tr>
<td># FAILED WRITES TO HIPERSPACE BUFFERS</td>
<td># FAILED WRITES TO HIPERSPACE BUFFERS ...% 0.0</td>
</tr>
</tbody>
</table>
| BUFFER HIT RATIO= 52.4% #

The Database Buffer Pool report includes the following information:

POOL ID

The pool ID supplied by the buffer pool definition statement.
BUFFER SIZE

The size, in bytes, of the buffers associated with this subpool.

FIXED COMPONENTS: INDEX/BLOCK/DATA

The components of the buffer pool that were page-fixed according to the buffer pool definition statement. A value of N indicates that the item is not page-fixed, and a value of Y indicates that the item is page-fixed.

SHARED RESOURCE POOL TYPE

The usage or type activity this pool is to handle.

# BUFFERS IN SUBPOOL

The total number of buffers in this subpool that are defined in virtual storage only.

# BUFFERS IN HIPER SPACE

The total number of buffers in this subpool that are defined in hiperspace buffers only.

TOTAL BUFFERS IN SUBPOOL

The total number of buffers in this subpool that are defined in virtual storage and hiperspace buffers.

# RETRIEVE BY RBA CALLS

The number of requests by the IMS VSAM request handler to locate a subpool buffer by database RBA.

# RETRIEVE BY KEY CALLS

The number of requests by the IMS VSAM request handler to locate a subpool buffer by database key.

# LOGICAL RECORDS INSERTED INTO ESDS

The number of times that a VSAM INSERT call was issued to insert a new data record into a VSAM ESDS.

# LOGICAL RECORDS INSERTED INTO KSDS

The number of times that a VSAM INSERT call was issued to insert a new data record into a VSAM KSDS.
# LOGICAL RECORDS ALTERED THIS SUBPOOL

The number of times that the IMS VSAM request handler requested that VSAM mark the buffer altered because of DL/I update activity.

# TIMES BACKGROUND WRITE INVOKED

The number of times that the VSAM Background Write function was invoked to write altered buffers. VSAM background write is requested to check for altered buffers that can be written back to the database to make buffers available. If background write is successful, buffer stealing will be avoided when buffers are needed for later DL/I activity.

# SYNCHRONIZATION CALLS RECEIVED

The number of times that IMS processed a checkpoint or syncpoint. For IMS DL/I batch jobs, a syncpoint occurs during an application checkpoint call and (automatically) during job step termination.

# VSAM GET CALLS ISSUED

The number of times that the IMS VSAM request handler issued a VSAM GET call to locate an RBA or key and establish ownership of the requested items buffer.

# VSAM SCHBFR CALLS ISSUED

The number of times that the IMS VSAM request handler issued a VSAM SCHBFR call to locate a buffer or range of buffers by RBA. This process is normally used during space management to locate candidate CIs that are already in the buffer pool.

# TIMES CONT INT ALREADY IN POOL

The number of times that the RBA or key requested by the VSAM GET call was already in the buffer subpool and no I/O was needed.

# CONT INTS READ FROM EXTERNAL STORAGE

The number of times that the RBA or key requested by the VSAM GET call was not in the buffer subpool and I/O was needed to obtain the CI.

# VSAM WRITES INITIATED BY IMS

The number of altered buffers in the subpool written back to the database. These buffers are normally written by background write or by synchronization processing.
VSAM WRITES TO MAKE SPACE IN POOL

The number of CIs written back to the database because of stealing of the subpool buffer for another request. Before the buffer can be used to satisfy the new request, it must be written back to the database because it is marked altered. Background Write was ineffective or not used.

PERM WRT ERRORS BUFFS NOW IN SUBPOOL

The number of error buffers in the pool.

LARGEST # PERM ERR BUFFS EVER IN SUBPOOL

The largest number of write errors ever in the subpool.

VSAM READS FROM HIPERSPACE BUFFERS

The number of subpool buffers read into the pool from hiperspace buffers.

VSAM WRITES TO HIPERSPACE BUFFERS

The number of times that a buffer was written or moved from a virtual storage buffer to a hiperspace buffer. If the buffer was altered, it was first written back to the database.

FAILED READS FROM HIPERSPACE BUFFERS

The number of times that a CI was expected to be in a hiperspace buffer but could not be located because the hiperspace buffer page had been stolen by MVS/ESA. A hiperspace buffer page can be stolen when system demand for ESTORE pages becomes excessive.

FAILED WRITES TO HIPERSPACE BUFFERS

The number of times that VSAM attempted to write a buffer out to a hiperspace buffer but the write failed because of a shortage of available ESTORE. Pages in ESTORE may not be available if the systems demand for ESTORE pages becomes excessive.

BUFFER HIT RATIO

The ratio of the number of CIs in the pool to the number of CIs in the pool and the number of read operations:

\[
\frac{\text{CIs in pool}}{\text{CIs in pool} + \text{reads}}
\]

This ratio is calculated for each subpool for which BCF detects activity.
DL/I Activity reports

DL/I Activity reports are optional. The collection and format of the report set is similar to the IMS DB Monitor report.

The timing and numbers reported are similar, but not identical, to the DB Monitor report collected for the same job step. See the IBM IMS Utilities Reference Manual for more information on the DB Monitor report.

DL/I Activity reports provide information about the number and time lengths of IWAIT and background IWAITS for the database. It is helpful to compare the results of these reports when you are tuning your database to achieve improved performance.

DL/I Activity reports include the following reports:

- Data Set IWAIT Summary
- Data Set IWAIT Distribution
- Data Set IWAIT Detail
- Background Write IWAIT Summary
- Background Write IWAIT Distribution

Data Set IWAIT Summary

The Data Set IWAIT Summary report is the first of three related reports that provide information about database data set IWaits.

The reference numbers (REF#) associate the rows on the report with rows on the other reports. Each row with the same reference number describes the same set of events. All times are reported in milliseconds.

Figure 47: Data Set IWAIT Summary report

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>REF#</th>
<th>TOTAL IWAITS</th>
<th>TOTAL TIME</th>
<th>MAX IWAIT</th>
<th>AVG IWAIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBIHD01</td>
<td>1</td>
<td>9,700</td>
<td>31,128.3</td>
<td>130.970</td>
<td>3.209</td>
</tr>
<tr>
<td>DBIIX01</td>
<td>2</td>
<td>52</td>
<td>172.4</td>
<td>7.210</td>
<td>3.315</td>
</tr>
</tbody>
</table>

The Data Set IWAIT Summary report includes the following information:

DDNAME

The ddname associated with the database data set.
The number used to relate the three IWAIT reports.

The total number of IWAITs detected during job step execution.

The total elapsed time for all detected IWAITs.

The maximum time encountered for an IWAIT during job step execution.

The average time for an IWAIT during job step execution.

Data Set IWAIT Distribution

The Data Set IWAIT Distribution report represents the distribution of IWAITs.

Figure 48: Data Set IWAIT Distribution report

<table>
<thead>
<tr>
<th>REF#</th>
<th>&lt; 10 MS</th>
<th>10 - 20 MS</th>
<th>20 - 40 MS</th>
<th>40 - 100 MS</th>
<th>&gt; 100 MS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9,533</td>
<td>137</td>
<td>26</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>52</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The Data Set IWAIT Distribution report includes the following information:

The number used to relate the three IWAIT reports.

The number of IWAITs less than 10 milliseconds.

The number of IWAITs of between 10 and 20 milliseconds.

The number of IWAITs of between 20 and 40 milliseconds.
40 - 100 MS
   The number of IWAITs of between 40 and 100 milliseconds.

> 100 MS
   The number of IWAITs greater than 100 milliseconds.

Data Set IWAIT Detail

The Data Set IWAIT Detail report shows additional details about the IWAIT events detected.

Figure 49: Data Set IWAIT Detail report

<table>
<thead>
<tr>
<th>REF#</th>
<th>MODULE</th>
<th>FIRST IWAIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VSAM</td>
<td>2.878</td>
</tr>
<tr>
<td>2</td>
<td>VSAM</td>
<td>3.261</td>
</tr>
</tbody>
</table>

The Data Set IWAIT Detail report includes the following information:

REF#

The number used to relate the three IWAIT reports.

MODULE

The buffer handler type that issued the IWAIT.

FIRST IWAIT

The time for the first IWAIT detected, in milliseconds.

Background Write IWAIT Summary

BCF produces two related reports on any IWAIT events detected for Background Write.

The reference numbers (REF#) associate the rows on the Background Write IWAIT Summary report to the rows on the Background Write IWAIT Distribution report (see “Background Write IWAIT Distribution” on page 224). Each row with the same reference number describes the same set of events. The reference numbers in these reports are separate and distinct from the reference numbers in the Data Set IWAIT reports.
Background Write reports are generated only when VSAM Background Write caused IWAIT events. All times are reported in milliseconds.

**Figure 50: Background Write IWAIT Summary report**

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>REF#</th>
<th>TOTAL IWAITS</th>
<th>TOTAL TIME</th>
<th>MAX IWAIT</th>
<th>AVG IWAIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBIHD01</td>
<td>1</td>
<td>102</td>
<td>579.0</td>
<td>27.911</td>
<td>5.676</td>
</tr>
<tr>
<td>DBIIX01</td>
<td>2</td>
<td>3</td>
<td>162.2</td>
<td>135.967</td>
<td>54.075</td>
</tr>
</tbody>
</table>

The Background Write IWAIT Summary report includes the following information:

**DDNAME**

The ddname associated with the database data set.

**REF#**

The number used to relate the two Background Write IWAIT reports.

**TOTAL IWAITS**

The total number of IWAITs detected during job step execution.

**TOTAL TIME**

The total elapsed time for all detected IWAITs.

**MAX IWAIT**

The maximum time encountered for an IWAIT during job step execution.

**AVG IWAIT**

The average time for an IWAIT during job step execution.

**Background Write IWAIT Distribution**

The Background Write IWAIT Distribution report provides information on the distribution of the detected Background Write IWAITS.
The Background Write IWAIT Distribution report includes the following information:

**REF#**

The number used to relate the two Background Write IWAIT reports.

**< 10 MS**

The number of Background Write IWAITs less than 10 milliseconds.

**10 - 20 MS**

The number of Background Write IWAITs between 10 and 20 milliseconds.

**20 - 40 MS**

The number of Background Write IWAITs between 20 and 40 milliseconds.

**40 - 100 MS**

The number of Background Write IWAITs between 40 and 100 milliseconds.

**> 100 MS**

The number of IWAITs greater than 100 milliseconds

---

### DL/I Call reports

DL/I Call reports, which are optional, include a series of matching reports that provide information about DL/I calls executed during the job step.

These reports are similar in appearance, but not identical, to the DB Monitor reports. A series of DL/I Call reports is produced for each PCB encountered.

DL/I Call reports include the following reports:

- DL/I Call IWAIT Summary
- DL/I Call Timing Summary
The DL/I Call IWAIT Summary report contains additional information about DL/I call IWAIT events.

Column totals appear at the end of the report.

**Figure 52: DL/I Call IWAIT Summary report**

| REF NO. | CALL FUNC | LOW LEV | SEGMENT NO. | PCBNAME=DBIHDO1 | DL/I CALLS IWAITS/ NOT IWAIT TOTAL | TOTAL
|---------|-----------|---------|-------------|------------------|-------------------------------------|--------
| 1       | ISRT      | 01      | DB#AASEG    |                  | 6,889 1.20 2.102 14,479.388         |        |
| 4       | ISRT      | 01      | DB#AASEG    |                  | 303 2.38 2.064 625.353              |        |
| 9       | GU        | 01      | DB#AASEG    |                  | 1 2.00 .814 .814                    |        |
| 10      | DLET      | 01      | DB#AASEG    |                  | 1 4.00 6.719 6.719                  |        |
| 14      | GU        | 01      | DB#AASEG    |                  | ...276 .68 .379 104.573            |        |
| 15      | DLET      | 01      | DB#AASEG    |                  | ...276 0.00 .004 1.207              |        |
| 2       | GU        | 03      | DB#CASEG    |                  | ...423 .81 .450 190.237            |        |
| 3       | REPL      | 03      | DB#CASEG    |                  | ...423 .03 4.911 2,077.159         |        |
| 11      | ISRT      | 03      | DB#CASEG    |                  | ...2 6.00 6.076 12.152             |        |
| 18      | ISRT      | 03      | DB#CASEG    |                  | ...92 .50 .274 25.229              |        |
| 5       | GU        | 02      | DB#BASEG    |                  | ...5 1.80 .720 3.599               |        |
| 6       | DLET      | 02      | DB#BASEG    |                  | ...5 3.60 4.319 21.595             |        |
| 12      | ISRT      | 02      | DB#BASEG    |                  | ...1 .00 .629 .629                 |        |
| 19      | ISRT      | 02      | DB#BASEG    |                  | ...46 1.09 .610 28.053             |        |
| 7       | GU        | 02      | DB#BBSEG    |                  | ...1 1.00 .497 .497                |        |
| 8       | DLET      | 02      | DB#BBSEG    |                  | ...4 4.00 8.049 8.049              |        |
| 13      | ISRT      | 02      | DB#BBSEG    |                  | ...1 6.00 2.270 2.270             |        |
| 20      | ISRT      | 02      | DB#BBSEG    |                  | ...46 0.00 .115 5.312              |        |
| 16      | GU        | 00      | GE          |                  | ...46 .98 .464 21.356              |        |
| 17      | DLET      | 00      | DJ          |                  | ...46 0.00 .005 .215               |        |
| TOTAL   |           |         |             |                  | 8,884 1.10 1.983 17,614.406       |        |

The DL/I Call IWAIT Summary report includes the following information:

**REF NO.**

The number used to relate events between associated reports. In addition, the reference number is used to report the chronological order in which each new call type was detected. A call type is identified by its call function, the lowest level retrieved, the segment name retrieved, and the status code. For this reason, the reference numbers may not occur in sequential order in the report.

**CALL FUNC**

The current type of DL/I call.

**LOW LEV**

The lowest level reached in the database during call execution. It is the same as the value reported in the database PCB.
SEGMENT NAME

The segment name returned in the database PCB when a call completes.

SC

The status code associated with a DL/I call. Refer to the IBM manual *Application Programming: DL/I Calls* for a complete listing of the status codes.

On this report, the SC (status code) field contains a blank to indicate successful retrieval of a segment. Status codes that IMS could have issued include GA, GE, GK, and blank.

DL/I CALLS TOTAL

The total number of DL/I calls for this call type detected.

IWAITS/CALL

The average number of IWAITs per call detected for this call type.

NOT IWAIT TIME AVG

The call elapsed time minus any IWAIT time, in milliseconds. This time includes execution time as well as any suspensions caused by non-IMS events.

NOT IWAIT TIME TOTAL

The total elapsed time, in milliseconds, for all the calls of this type when the call was not in an IWAIT state. This time includes execution time as well as any suspensions caused by non-IMS events.

TOTAL

The totals for the column. If the column represents a count, the total value at the end of the report is the sum of the values in the column. If the column represents an average, the total value at the end of report is the average of the values in the column.

**DL/I Call Timing Summary**

The DL/I Call Timing Summary report (the figure below) provides information for each PCB encountered.
Column totals appear at the end of the report.

**Figure 53: DL/I Call Timing Summary report**

<table>
<thead>
<tr>
<th>REF NO.</th>
<th>NUMBER IWAITS</th>
<th>CALL IWAIT TIME AVG</th>
<th>CALL IWAIT TIME TOTAL</th>
<th>CALL ELAPSED TIME AVG</th>
<th>CALL ELAPSED TIME TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8.288</td>
<td>3.204</td>
<td>26,556.069</td>
<td>5.957</td>
<td>41,035.457</td>
</tr>
<tr>
<td>4</td>
<td>720</td>
<td>3.193</td>
<td>2,299.156</td>
<td>9.652</td>
<td>2,924.509</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>3.354</td>
<td>6.707</td>
<td>7.521</td>
<td>7.521</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>3.559</td>
<td>14.235</td>
<td>20.954</td>
<td>20.954</td>
</tr>
<tr>
<td>14</td>
<td>188</td>
<td>3.060</td>
<td>575.199</td>
<td>2.463</td>
<td>679.772</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>.000</td>
<td>.000</td>
<td>.004</td>
<td>1.207</td>
</tr>
<tr>
<td>2</td>
<td>344</td>
<td>3.089</td>
<td>1,062.783</td>
<td>2.962</td>
<td>1,253.020</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>3.237</td>
<td>42.080</td>
<td>5.010</td>
<td>2,119.239</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>3.266</td>
<td>39.186</td>
<td>25.669</td>
<td>51.338</td>
</tr>
<tr>
<td>18</td>
<td>46</td>
<td>3.351</td>
<td>154.164</td>
<td>1.950</td>
<td>179.393</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>17.518</td>
<td>157.664</td>
<td>32.253</td>
<td>161.263</td>
</tr>
<tr>
<td>6</td>
<td>18</td>
<td>4.076</td>
<td>73.373</td>
<td>18.994</td>
<td>94.968</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>2.239</td>
<td>2.239</td>
<td>2.868</td>
<td>2.868</td>
</tr>
<tr>
<td>19</td>
<td>50</td>
<td>2.938</td>
<td>146.882</td>
<td>3.803</td>
<td>174.935</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>3.048</td>
<td>3.048</td>
<td>3.545</td>
<td>3.545</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>4.011</td>
<td>16.044</td>
<td>24.093</td>
<td>24.093</td>
</tr>
<tr>
<td>13</td>
<td>6</td>
<td>2.848</td>
<td>17.089</td>
<td>19.359</td>
<td>19.359</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>.000</td>
<td>.000</td>
<td>.115</td>
<td>5.312</td>
</tr>
<tr>
<td>16</td>
<td>45</td>
<td>2.935</td>
<td>132.097</td>
<td>3.336</td>
<td>153.453</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>.000</td>
<td>.000</td>
<td>.005</td>
<td>.215</td>
</tr>
<tr>
<td>TOTAL</td>
<td>9,751</td>
<td>3.210</td>
<td>31,298.015</td>
<td>5.506</td>
<td>48,912.421</td>
</tr>
</tbody>
</table>

The DL/I Call Timing report includes the following information:

**REF NO.**

The number used to relate events between associated reports. In addition, the reference number is used to report the chronological order in which each new call type was detected. A call type is identified by its call function, the lowest level retrieved, the segment name retrieved, and the status code. For this reason, the reference numbers may not occur in sequential order in the report.

**NUMBER IWAITS**

The total number of IWAITs detected for this call type.

**CALL IWAIT TIME AVG**

The average IWAIT time, in milliseconds, for all IWAITs detected for this call type.

**CALL IWAIT TIME TOTAL**

The total IWAIT time, in milliseconds, for all IWAITs detected for this call type.
CALL ELAPSED TIME AVG

The average elapsed time, in milliseconds, for a call of this type.

CALL ELAPSED TIME TOTAL

The total elapsed time, in milliseconds, for all calls of this type.

TOTAL

The totals for the column. If the column represents a count, the total value at the end of the report is the sum of the values in the column. If the column represents an average, the total value at the end of report is the average of the values in the column.

Execution Options Summary report

The Execution Options Summary report lists the BCF processing option values that were used during this job step execution.

This report is saved in the statistics record in the history data set and in the BCFSTATS data set.

A similar report, the Pre-Execution Options Summary report (not shown), shows the same information as the Execution Options Summary report, but the pre-execution report is produced before the option values are adjusted to accommodate situations
encountered during initialization. The pre-execution report is saved only in the BCFPRINT data set.

**Figure 54: Execution Options Summary report**

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>OPTION</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM</td>
<td>BATCH CONTROL FACILITY ACTIVE</td>
<td>YES</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>USE EXECUTION Control</td>
<td>YES</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>USE PHYSICAL LOGGER</td>
<td>YES</td>
</tr>
<tr>
<td>BCFSYSIN</td>
<td>DASD UNIT NAME</td>
<td>SYSDA</td>
</tr>
<tr>
<td>BCFSYSIN</td>
<td>STANDARD TAPE UNIT NAME</td>
<td>VCRT90</td>
</tr>
<tr>
<td>BCFSYSIN</td>
<td>BUFFERED TAPE UNIT NAME</td>
<td>VCRT80</td>
</tr>
<tr>
<td>BCFSYSIN</td>
<td>JOBLOG MESSAGES</td>
<td>LIMITED</td>
</tr>
<tr>
<td>BCFSYSIN</td>
<td>PHYSICAL LOGGER ERROR OPTION</td>
<td>ABEND</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>DELETE VOLUME MEMBERS</td>
<td>NO</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>QUALIFY WITH PROGRAM</td>
<td>NO</td>
</tr>
<tr>
<td>BCFSYSIN</td>
<td>AUTOMATIC BATCH BACKOUT</td>
<td>YES</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>BUFFERED DEVICE SUPPORT</td>
<td>BUFFER</td>
</tr>
<tr>
<td>BCFSYSIN</td>
<td>DISCARD LOGGING ACTIVE</td>
<td>NO</td>
</tr>
<tr>
<td>BCFSYSIN</td>
<td>EXCHANGE LOG DEVICE TYPE</td>
<td>NO</td>
</tr>
<tr>
<td>BCFSYSIN</td>
<td>DYNAMIC LOG OPTION</td>
<td>NO</td>
</tr>
<tr>
<td>BCFSYSIN</td>
<td>DYNAMIC LOG TYPE</td>
<td>MEMORY</td>
</tr>
<tr>
<td>BCFSYSIN</td>
<td>TAPE BKO</td>
<td>NO</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>MAX SLOTS BLOCK COUNT</td>
<td>0</td>
</tr>
<tr>
<td>BCFSYSIN</td>
<td>DL/I ACTIVITY STATISTICS</td>
<td>FULL</td>
</tr>
<tr>
<td>BCFSYSIN</td>
<td>RECORD DL/I ACTIVITY STATS TO</td>
<td>HIST</td>
</tr>
<tr>
<td>BCFSYSIN</td>
<td>NUMBER OF STATISTIC SETS</td>
<td>2</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>SYSOUT CLASS</td>
<td>*</td>
</tr>
<tr>
<td>BCFSYSIN</td>
<td>LOGGER PERF % THRESHOLD</td>
<td>10</td>
</tr>
<tr>
<td>BCFSYSIN</td>
<td>ALERT USERID</td>
<td>TFC2</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>DEPARTMENT VARIABLE</td>
<td>DEPT</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>BACKOUT VARIABLE</td>
<td>BK0</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>USE LOG COMPACTION</td>
<td>YES</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>MANUAL INTERVENTION OVERRIDE</td>
<td>NO</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>IGNORE CHECKPOINT</td>
<td>NO</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>BCF ABEND CODES</td>
<td>OLD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>OPTION</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOBAL</td>
<td>HISTORY DSN</td>
<td>BMCBCF.QA.BCFS.VSAM.HIST</td>
</tr>
</tbody>
</table>

The Executions Options Summary report includes the following information:

**SOURCE**

The source of the option value. The following values are valid in this field:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOBAL</td>
<td>The option value was set at the global level in the REGISET.</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>The option value was set at the system level in a system record of the REGISET.</td>
</tr>
<tr>
<td>JOBSTEP</td>
<td>The option value was set at the job step level in a job step record of the REGISET.</td>
</tr>
<tr>
<td>BCFSYSIN</td>
<td>The option value was set at the execution level with a keyword in the BCFSYSIN control statement data set.</td>
</tr>
</tbody>
</table>
### Value Description

**AUTO**
The option value was set automatically by BCF. For example, if AR/CTL is active during BCF initialization, BCF automatically sets the Ignore Checkpoints option to **N** and sets the field to **AUTO** in the Execution Options Summary report (saved in the statistics record of the history data set and in the BCFSTATS data set). This setting is different from the setting in the Pre-Execution Options Summary report (saved in the BCFPRINT data set).

**JCL**
The option value was set in the JCL to override preset option values.

### OPTION

The name of the option. The History DSN option is listed separately at the bottom of the report since it can be a large value. For more information, see “Using BCF options” on page 61.

### VALUE

The option value that was used during job step execution.
Using the log utilities

This section provides information about the log utilities that are available with the BMC Software BATCH CONTROL FACILITY (BCF) component of APPLICATION RESTART CONTROL (AR/CTL).

BCF Log utility

The BCF Log utility helps you analyze BCF operations and provides an emergency solution to some recovery requirements.

The utility allows you to perform the following tasks:

- Run the IMS Batch Backout utility and back out the complete execution of a job step when the application has used checkpoint calls but did not begin execution with a checkpoint call
- Complete backout for a batch message processing (BMP) execution since the IMS Batch Backout utility does not support complete backout for a BMP that has participated in syncpoint processing

**WARNING**

Using the BCF Log utility to satisfy the foregoing recovery requirements may create data integrity exposures. Only experienced DBAs should use the utility in emergency situations and then only as a measure of last resort.

The BCF Log utility provides the following functions:

- Statistics (STATS)
- Logical compare (COMPARE)
- Batch backout log create (BCREATE)
- Online backout log create (OCREATE)
- Find online BMP executions (FIND)
BCF provides sample JCL for executing the BCF Log utility. Member #BCFLOGU of the BCF sample library contains sample JCL for IMS versions before 5.1. Member #BCFLOG5 of the BCF sample library contains sample JCL for IMS Version 5.1. Member #BCFLOG6 of the BCF sample library contains sample JCL for IMS Version 6.1.

## Control statements

The control statements used by the BCF Log utility are simple and easy to use.

The control statements consist of two major elements: commands and keywords. The following general rules apply when you specify the control statements:

- Comment cards are allowed and should begin with asterisk (*) in column 1.
- Keywords and values must be contained in columns 1 to 72. Use columns 1 to 72 only.
- Continuation is not allowed.
- A control statement can contain multiple requests. Commands can be separated by one or more blanks or commas. Keywords and values must be separated by a comma.
- A ddname value cannot be longer than eight characters and must be present in the JCL.
- If a control statement contains all blanks, it will be bypassed and the job step will end with a return code of 4 or greater.
- The DBRC parameter (PARM=DBRC) causes OCREATE to set DBRC active in the batch log data set that is created.

When specifying the keywords to invoke these functions, you can furnish the entire keyword or just the abbreviated portion (shown in capital letters).

When requesting a utility function, you must adhere to the following syntax rules:

- The command is separated from its values by an equal (=) sign.
- All values must be contained within a single set of parentheses.
- If the request requires multiple values for the request, each value must be separated by a comma.
- If the command allows keywords, the keyword is separated from its value with an equal sign (=).
A keyword and its associated value are separated from other keyword/value combinations and other values with commas.

**Commands**

The following commands are used with the BCF Log utility:

**STAts**

Create a statistical report from the input log data set. The BCFSTATS DD statement is required.

**COMpare**

Compare two log data sets logically and report on any differences.

**BCReate**

Create an IMS batch log data set to be used as input to the IMS Batch Backout utility. The input is an IMS batch log tape containing checkpoint records. The log data set created does not contain any checkpoint records.

**OCReate**

Create an IMS batch log data set to be used as input to the IMS Batch Backout utility. The input is an IMS online log tape (system log data set) containing BMP executions. The log data set created contains all changes created by the BMP execution. Any checkpoints issued by the BMP are ignored.

**FINd**

Scan the input log and create a report of BMP executions.

**Keywords**

The following keywords are used with the BCF Log utility:

**PSB=psbname**

The program specification block (PSB) name to be used with the FIND and OCReate functions.

**UOW=uow**

The unit of work to be used with the OCReate functions.

**BBO=psbname**

The PSB name to be inserted into any selected database change records written to the output log data set. The PSB named in this keyword is used.
when the batch backout execution is actually performed. The PSB used with batch backout must contain all databases for which database change records have been selected during utility execution.

**UPD=PSB/ALL**

The **UPD** keyword indicates to the **OCREATE** function which database change records are to be selected and written to the log data set. A value of **PSB** indicates that, to be written to the log data set, the database change records must match the specified PSB/UOW.

A value of **ALL** indicates that the **OCREATE** function must write all database change records necessary to back out all *updated* databases. This is a two-phase process, which is reflected on the associated reports. The PSB/UOW specified is used to identify the starting point for the extract operation.

The collection of all database change records may create a cascading effect and may require backout of databases not updated by the PSB/UOW named as the starting point.

**FORCE**

The **FORCE** keyword controls creation of the output log data set when **UPD=ALL** is specified. This option allows you to execute the second phase unconditionally.

If specified, the output data set will be created even if errors have been detected that may result in an incomplete backout of the affected databases.

If this keyword is not specified, the output data set will not be created if errors are detected that may result in an incomplete backout of the affected databases.

**TRAN=trancode**

The **TRAN** keyword provides the message processing program (MPP) transaction code for **OCREATE** and **FIND**. This keyword also qualifies the begin point for extraction and allows printing of MPP schedules using the **FIND** function. The **TRAN** keyword can only be used when the begin point is an MPP and not a BMP. If the **TRAN** keyword is not used and the desired scheduling is an MPP, the **OCREATE** request will fail. When **TRAN** is used with the **FIND** function, the PSB keyword must also be used.

### Overview of units of work (UOWs)

The IMS online system uses a unit of work (UOW) concept to identify each application scheduling uniquely.
The identifier is 16 bytes (hexadecimal) and has the following format:

**Bytes 0-3**

The IMSID of the system.

**Bytes 4-7**

Blanks.

**Bytes 8-11**

The origin application sequence number (OASN) for the event. This number is set to zero when the online IMS system is cold-started and is incremented by one each time an application (the transaction or the BMP) is scheduled.

**Bytes 12-15**

The syncpoint number. The syncpoint number is set to zero during scheduling and increments each time the application is involved in the syncpoint process. The application is involved in the syncpoint process during application checkpoint processing, processing for the DL/I **SYNC** call, application termination if database updates or communication messages occurred, and when a **GU** to the I/O PCB is issued.

The **OCREATE** function requires the unit of work OASN to be supplied as part of the BMP execution selection criteria. This is necessary to prevent selection of database change records for a parallel scheduled program specification block (PSB). The OASN appears on the Listing of BMP Start/Stop Records Report generated by the **FIND** function. See the figure in “Use of the Find Online BMP Executions function” on page 246 for an example.

---

**Note**

The **FIND** function is used to obtain the executions for the desired PSB and return the OASN and syncpoint number for the UOW associated with each PSB execution. If the BMP issued checkpoints, the syncpoint number in the UOW when the BMP accounting record is written will be greater than 1.

---

**Control statement formats**

Use control statements to request functions.

Each request for a function requires different input for execution. Here are examples of each request type and their required and optional values:
Statistics

The format consists of the keyword STATS (or STA), followed by the ddname of the input log data set. You can specify multiple ddname values. For example:

```
STATS=(DD1,DD2,DD3,...)
```

Compare batch log data sets

The format consists of the keyword COMPARE (or COM), followed by two input log data sets. You can specify only two ddname values. The first value represents the input ddname, and the second value represents the ddname of a second input log data set. For example:

```
COMPARE=(DD1,DD2)
```

Batch backout log create

The format consists of the keyword BCREATE (or BCR), followed by an input and an output log data set. You can specify only two ddname values. The first value represents the input ddname and the second value represents the ddname of the output log data set. For example:

```
BCREATE=(INDD1,OUTDD1)
```

Online backout log create

The format consists of the OCOREATE (or OCR) keyword, followed by input and output log data sets and the PSB and UOW keywords and their values. You can specify only two ddname values. The first value represents the input ddname, and the second value represents the ddname of the output log data set. PSB and UOW provide the PSB name and UOW OASN (see “Overview of units of work (UOWs)” on page 236) used to qualify the BMP execution. PSB and UOW can appear in any order. The value for the UOW keyword must be eight characters. For example:

```
OCREATE=(INDD1,OUTDD1,PSB=psb,UOW=xxxxxxxx)
```

Find online PSB executions

The format consists of the FIND (or FIN) keyword, followed by an input log data set and the PSB to find BMP executions. You can specify only one ddname. The optional PSB keyword invokes reporting only on the PSB name provided with this keyword. For example:

```
FIND=(DD1,PSB=psbname)
```

Example control statement formats

Figure 55 on page 238 is an example of control statement formats.

**Figure 55: Control statement format—example 1**

```
STATS=(DD1)
STA=(DD1,DD2,DD3)
```
Use of the statistics function

The statistics (STATS) function provides a quick analysis of a log data set to help you understand the logging and execution characteristics of a job step or online system that created the log.

The utility furnishes reports for batch or online log data sets.

Figure 57: IMS Log Utility report (page 1)

<table>
<thead>
<tr>
<th>BCF</th>
<th>IMS LOG UTILITY</th>
<th>PAGE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LOG DATA SET INFORMATION</td>
<td></td>
</tr>
<tr>
<td>DDNAME</td>
<td>DEVICE TYPE</td>
<td>BLOCKSIZE</td>
</tr>
<tr>
<td>-------</td>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td>DD1</td>
<td>TAPE</td>
<td>32,760</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DATA SET NAME</th>
<th>JOBNAME</th>
<th>SHARE STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLI.BCFA.BCV205.IEFRDER</td>
<td>BCFV2205</td>
<td>IRLM=NO DBRC=YES</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOG START TIME</th>
<th>LOG STOP TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>97031/11.55.18.1</td>
<td>97031/11.59.56.5</td>
</tr>
</tbody>
</table>

SIZE OF BLOCKS ENCOUNTERED

MIN=58 AVG=11,132 MAX=32,696

NUMBER OF PAD RECORDS FOUND=20
NUMBER OF LOG WRITE AHEADS=18
TOTAL PHYSICAL BLOCKS WRITTEN=20

VOLUME # VOLSER
---------- ----------
1 200732
For batch data sets, the report includes the following information:

- IMS execution type that created the log data set
- Type and distribution of the log records on the data set
- Log physical characteristics
If it cannot be determined that the log data set is a batch log (the log volume does not contain X'06' log record, indicating a batch log data set), the utility assumes the data set is an online log data set. The following information is reported:

- IMS execution type that created the log data set
- Type and distribution of the log records on the data set
- An analysis of log records for each database data set
  (The analysis includes the number of database change records and the total bytes of log records for each database data set.)

**Overview of the Logical Compare function**

The Logical Compare (COMPARE) function allows you to compare IMS batch log tapes to determine whether the output logs are logically identical.

Using the Logical Compare function is necessary when comparing identical runs of a batch program with and without use of BCF. The Logical Compare function uses a logical approach because log records contain time-sensitive information. The Logical Compare function also checks log record sequence numbers and key log record
flags. The utility provides a short report that shows block and record counts for both logs.

**Figure 60: Log Utility Logical Compare report**

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>BLKSIZE</th>
<th>DSNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>16,384</td>
<td>DLI.BATCH.LOG</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>16,384</td>
<td>SYS94215.T152748.RA000.BCFLGBCS.LOG1</td>
</tr>
</tbody>
</table>

**Overview of the Batch Backout Log Create function**

The Batch Backout Log Create (BCREATE) function creates a batch log file that allows batch backout to the beginning of the application execution.

It creates a log data set that is supplied to the IMS Batch Backout utility. This data set does not contain any application checkpoint X'41' log records because they are converted to X'67' pad records. The utility validates that the input log has been produced by a batch IMS execution.

The report section provides details of the input log and execution of the function in the following areas:

- Job name, start time, and start date
- DBRC and IRLM indicators
- Analysis of updated database data sets
- Number of checkpoint log records converted
Number of blocks in the log data set
Log data set stop time and stop date

As shown in Figure 61 on page 243, the following types of warning messages also may appear in the output reports:

Usage disclaimer
Warning of potential data sharing environment if the IRLM was active during job step execution

**Figure 61: Batch Backout Log Create Utility report**

<table>
<thead>
<tr>
<th>JOBNAME</th>
<th>LOG START DATE/TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCFBLD1</td>
<td>97031 16.20.07.4</td>
</tr>
</tbody>
</table>

**Overview of the Online Backout Log Create function**

The Online Backout Log Create (OCREATE) function allows batch backout of an IMS batch message region (BMP) execution.

The OCREAT E function is used to create an IMS batch log data set from an IMS online system log data set. The batch log data set is used as input to the IMS Batch Backout utility to perform backout of the BMP execution. The batch log data set created represents the total BMP execution even if the BMP issued checkpoints. The BMP should have terminated normally.

You need to supply control statement information as follows:

- ddnames for input and output log data sets
- PSB name of the BMP execution
- UOW OASN identifier associated with the scheduling of the BMP execution
- Transaction code if desired scheduling is not a BMP
Note
The beginning point for extract must be a BMP execution unless UPD=ALL is specified.

Warning messages that may appear in the report include the following:

- Usage disclaimer
- Database data set update by another UOW detected

Figure 62: Online Backout Log Create report (page 1)
PROJECTED RESULTS OF PHASE II EXECUTION. BASED UPON PHASE I, THE FOLLOWING DATABASE CHANGES FROM THE LISTED PSB(S) WILL BE EXTRACTED FOR BACKOUT DURING PHASE II EXECUTION.

DATABASE AND PROJECTED COUNT OF RECORDS TO BE EXTRACTED DURING PHASE II

- BCFTA21D.................59
- TOTAL RECORDS ............59

PSBNAME(S) RESPONSIBLE FOR DATABASE CHANGES TO BE EXTRACTED IN PHASE II

- BCFL2010
- BCFL2011
- BCFP3010
- BCFP3011

PHASE II BEGINNING POINT FOR EXTRACT; PSB=BCFL2011 UOW=0040A5A7

BEGINNING PHASE II EXECUTION

BMC48001I BCF LOG UTILITY; BEGIN LOG TAPE EXTRACT PHASE 2
PSB:IBBL2011  UOW:0040A5A700000000
BMP SCHEDULING START LOG RECORD SEQ NUMBER=00DDFFC9
DATABASE UPDATE FOUND FOR UOW THAT STARTED BEFORE SELECTED PSB/UOW

BMC48002I BCF LOG UTILITY; END LOG TAPE EXTRACT PHASE 2, RC=000000C
NEW PSB/UOW NEEDED FOR RESTART; PSB=BCFL2010  UOW=0040A5A6
Use of the Find Online BMP Executions function

The purpose of the Find Online BMP Executions (FIND) function is to identify all BMP executions of a given PSB name and to provide the unit of work OASN and syncpoint number.

The OASN is then provided in the OCREATE request to identify the BMP execution uniquely.

The function will scan the input log data set looking for batch message region start and stop records. When these records are found, the function provides certain information.

For a BMP start log record, the following information is reported:

- START constant
- PSB name
- UOW (OASN and syncpoint number)
- Start date
- Start time

For a BMP stop log record, the following information is reported:

- STOP constant
- PSB name
- UOW (The first eight bytes are the OASN, and the last eight bytes are the syncpoint number.)
- Stop date
- Stop time
- An indication that the application issued DL/I update calls
This function can also be used to find start and stop records of MPP executions. It can only be performed, however, for an execution when a PSB name is requested.

### Figure 65: Listing of BMP Start/Stop Records report

<table>
<thead>
<tr>
<th>EVENT</th>
<th>PSB NAME</th>
<th>UOW</th>
<th>DATE</th>
<th>TIME</th>
<th>UPDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>STOP</td>
<td>BCF007</td>
<td>0041B091000000000</td>
<td>97031</td>
<td>9.30.59.9</td>
<td>NO</td>
</tr>
<tr>
<td>START</td>
<td>BCF006</td>
<td>0041B0F8000000000</td>
<td>97031</td>
<td>9.31.07.5</td>
<td></td>
</tr>
<tr>
<td>STOP</td>
<td>BCF006</td>
<td>0041B0F8000000000</td>
<td>97031</td>
<td>9.31.11.0</td>
<td>YES</td>
</tr>
<tr>
<td>STOP</td>
<td>BCF006</td>
<td>0041B120000000000</td>
<td>97031</td>
<td>9.31.11.7</td>
<td></td>
</tr>
<tr>
<td>START</td>
<td>BCF006</td>
<td>0041B121000000000</td>
<td>97031</td>
<td>9.31.17.7</td>
<td></td>
</tr>
<tr>
<td>START</td>
<td>BCF006</td>
<td>0041B121000000000</td>
<td>97031</td>
<td>9.31.18.1</td>
<td>YES</td>
</tr>
<tr>
<td>START</td>
<td>BCF006</td>
<td>0041B121000000000</td>
<td>97031</td>
<td>9.31.18.6</td>
<td></td>
</tr>
<tr>
<td>START</td>
<td>BCF006</td>
<td>0041B121000000000</td>
<td>97031</td>
<td>9.31.18.7</td>
<td></td>
</tr>
</tbody>
</table>

**Utility return codes**

The return codes issued by the BCF Log utility are based on the following conventions:

- A return code of 8 or more results in termination of the utility.
- A return code of less than 8 allows the utility to continue processing the next function request.
The return codes produced by the BCF Log utility have the following meanings, depending on the conditions under which they were produced:

4

A SYSIN data set control statement that did not contain any keywords was processed, or the COMPARE function found a minor compare error.

7

The OCREATE function detected a condition that requires evaluation but completed processing otherwise, or the BCREATE function detected a condition that requires evaluation but completed processing otherwise.

8

The COMPARE function found a severe compare error.

12

The OCREATE function found a severe error when extracting the records for the requested program specification block (PSB) execution.

Database Changed Block Reconciliation utility

The Database Changed Block Reconciliation utility is an extension of the BCF Log utility and is used to identify and report on database blocks that have been changed by more than one UOW.

This function produces a report that enables a DBA to make a more informed judgement about the effect of executing the Batch Backout utility with the log data set created by the OCREATE function. By using this report with the reports produced during OCREATE execution, the DBA can determine whether execution of Batch Backout is likely to result in a damaged database.

Utility control information and JCL

The utility program JCL and control information has the following characteristics:

- The utility accepts parameter input on line count in the format LINCT=XXX.
- The output report is written to the BCFSTATS DD statement with an FBA format and an LRECL of 81.
The input log data set is the BCFLOGIN DD statement. It uses a VB format with an LRECL of 32756 and a block size of 32760. For more information, refer to “Utility input” on page 249.

In addition, the utility requires that Sort Work DD statements used for the installation sort program must function correctly.

You must prepare JCL that executes program BCFDBCON from the BMC.BCF.LOAD library supplied on the distribution tape. Member #BCFCBR of the BCF sample library contains sample JCL for this task. Sample JCL is shown in the figures in “Program report” on page 250.

**Basic program function**

The Database Changed Block Reconciliation utility operates like a sort exit for the BCF Log utility.

After the utility reads an input file and creates a consolidated sort element, it passes the element record for sorting. Then the sorted elements are retrieved and used to produce a report on database blocks that have been updated by more than one UOW.

**Utility input**

The Database Changed Block Reconciliation utility is intended to process input from an OCREATE log or BCFDBCON DD statement file created by the BCF Log Utility.

The Database Changed Block Reconciliation utility will also accept system log data set (SLDS) input. When a complete SLDS is used as input, only the database update log records are used. All other log records are bypassed.

The Database Changed Block Reconciliation utility also supports Fast Path log records and reports on the same concurrent database updates for Fast Path databases and full-function databases.

**Utility output**

The output of the Log utility is a report that provides detailed information on each database that has change log records.
The blocks of each database that were updated by multiple units of work are detailed. The report indicates which RBA or RBN, PSB name, and UOW are associated with each block.

**Program report**

The report produced by the Database Changed Block Reconciliation utility is similar to most reports the BCF Log utility produces.

The header for this report is identical to the header used for other reports generated by this utility. The width of the report is limited to 80 characters plus control characters, to enable easy viewing on a standard 3270 screen.

Figure 66 on page 250 shows sample JCL for the Database Changed Block Reconciliation utility using IMS logs as input.

**Figure 66: Database Changed Block Reconciliation Utility JCL**

```plaintext
//******************************************************************
//********* DATABASE CHANGED BLOCK RECONCILIATION UTILITY **********
//******************************************************************
//STEP1    EXEC  PGM=BCFDBCON,REGION=4096K,PARM='LINCT=030'
//STEPLIB  DD  DISP=SHR,DSN=BMC.AES.LOAD
//         DD  DISP=SHR,DSN=BMC.BCF.LOAD
//SYSPRINT DD  SYSOUT=*  
//BCFSTATS DD  SYSOUT=*  
//BCFLOGIN DD  DISP=SHR,DSN=SLDS1.LOG
//         DD  DISP=SHR,DSN=SLDS2.LOG
//*
//SYSOUT  DD  SYSOUT=*  
//SORTLIB  DD  DISP=SHR,DSN=SYS1.SORTLIB
//SORTWK01 DD  SPACE=(CYL,(50,1)),UNIT=VIO
//SORTWK02 DD  SPACE=(CYL,(50,1)),UNIT=VIO
//SORTWK03 DD  SPACE=(CYL,(50,1)),UNIT=VIO
//SORTWK04 DD  SPACE=(CYL,(50,1)),UNIT=VIO
//SORTWK05 DD  SPACE=(CYL,(50,1)),UNIT=VIO
//SORTWK06 DD  SPACE=(CYL,(50,1)),UNIT=VIO
///*
```
Figure 68 on page 251 shows an example of the report that the Database Changed Block Reconciliation utility produces.

**Figure 68: Database Block Update report**

```
<table>
<thead>
<tr>
<th>DATABASE BLOCK UPDATE REPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATABASE........DBASE1</td>
</tr>
<tr>
<td>RBA/RBN........006A3000</td>
</tr>
<tr>
<td>PSBN0001</td>
</tr>
<tr>
<td>PSBN0004</td>
</tr>
<tr>
<td>PSBN0005</td>
</tr>
<tr>
<td>PSBN0008</td>
</tr>
<tr>
<td>PSBN0008</td>
</tr>
<tr>
<td>PSBN0008</td>
</tr>
<tr>
<td>RBA/RBN........006B4000</td>
</tr>
<tr>
<td>PSBN0002</td>
</tr>
<tr>
<td>PSBN0003</td>
</tr>
<tr>
<td>PSBN0005</td>
</tr>
<tr>
<td>PSBN0005</td>
</tr>
<tr>
<td>PSBN0005</td>
</tr>
</tbody>
</table>
```

Database Changed Block Reconciliation utility
Contacting BMC Software product support

A BMC Software product support representative for the BATCH CONTROL FACILITY (BCF) component of APPLICATION RESTART CONTROL (AR/CTL) is on call 24 hours a day, seven days a week, 365 days a year. You can contact a product support representative as described on the back of the title page of this manual.

How to expedite resolution of your problem

You can help the product support representative find a faster resolution to your problem by having the following information available when you call:

- Version, release, and maintenance level of the BMC Software product
- Version, release, and maintenance level of IMS
- Listing of the JCL you were using to run the job
- List of any product messages that were issued by the job, which can be found in the JES job log data set

The following DD statements may also be useful if you include them in the job step JCL:

- BCFTRACE
- BCFBOTRC
- SYSUDUMP

BCFPRINT data set

The BCFPRINT data set is used to report syntax errors for keywords specified on the BCFSYSIN DD statement.
It is also used by BMC Software Product Support to diagnose errors in dynamically allocating data sets. The text units for any SVC99 dynamic allocation request issued by BCF is detailed in the BCFPRINT data set.

**BCFTRACE DD**

If a problem occurs during BCF processing, the product support representative may ask you to include the BCFTRACE DD statement to define a trace data set.

The BCF Physical Logger writes trace information to this data set. The trace data set should have a logical record length (LRECL) of 4096, a record format (RECFM) of FB, and a block size (BLKSIZE) that is an even multiple of 4096.

If the block size is not supplied in the JCL, it will default to the **AUTO BLOCK** setting for the device being used.

**BCFBOTRC DD**

This DD statement serves the same purpose as BCFTRACE, except that it is used when the execution controller invokes batch backout.

**SYSUDUMP DD**

If you include the standard SYSUDUMP DD statement in the JCL and a problem occurs during execution of the batch job, the execution controller writes diagnostic information to the SYSUDUMP data set.

It is normally defined as a **SYSOUT=*** data set.

The SYSUDUMP DD statement should be present for most abend conditions. If it is not present, contact BMC Software before rerunning the job. A product support representative will determine whether a SYSUDUMP is necessary.

Abbreviated dumps do not provide the type of information needed for problem resolution. Please ensure that the dump is complete. If you have the Abend-AID product installed, you can use one of the following DD statements to produce a complete dump:

```plaintext
//ABNLIGNR DD DUMMY
//ABNLDUMP DD DUMMY
```
The first statement turns Abend-AID off for this job step and allows a SYSUDUMP to be produced. The second statement produces an Abend-AID dump, followed by a SYSUDUMP.
Index

&-type substitution keywords 133

#BCFACTJ sample 162
#BCFALIN sample 136
#BCFCBR sample 248
#BCFHIST sample 192
#BCFLOG5 sample 233
#BCFLOG6 sample 233
#BCFLOGU sample 233
#BCFPURG member 186

A

abend codes
  U625 120
  U628 27
  U629 27

abend filter processing 25

Abend-AID dump 254

ABENDED keyword in BCFACTJ0 164

ABNcode keyword 108

ABNLDUMP DD statement 254

ABNLIGNR DD statement 254

activation of BCF 73

Active BMC Consolidated Subsystems panel 49

Active Job Display panel 154

Active Job Status Modification panel 159

Active Job utility
  use after program abend 161
  using to catalog log volumes 161

Active Jobsteps Display panel 152

Add Record panel
  general 51

AES
  application supervisor 19
  overview 15

AES global options
  record overview 41

AES Status Check utility 39

AESUVBCS program 39

ALERT keyword 102

Alert User ID option 102

Application Checkpoint Detail report 204

application checkpoint processing 30

Application Enhancement Solutions primary menu 46

application reattach and automatic log exchange 113

application supervisor 19

AR/CTL
  reattach of applications 113
  use with U616 device class 115

ASK parameter of BCFACTJ0 162

AUTObbo keyword 84

automated operations 36

Automatic Batch Backout option 20, 84, 161

automatic log exchange 113

Automatic Restart Preparation utility 159, 161, 178, 180

B

Background Write IWAIT Distribution report 225

Background Write IWAIT Summary report 223

BACKOUT device class 115

batch backout 27, 233

  BCF participation in DFSBBO00 35
  log data sets 115

Batch Backout % Space Increase field for dynamic allocation 122, 129

Batch Backout Log Create report 242

BATCH CONTROL FACILITY - Active Job Display panel 154

BATCH CONTROL FACILITY Active option 73

BATCH CONTROL FACILITY primary menu 48

BBO keyword of BCFLGUTL utility 235

BBOSYSIN ddname 58

BCF
automated operations support 36
Automatic Restart Preparation utility 180
components 16
DBRC support 35
history data set 186
IMS levels supported 34
operations 151
overview 15
purge utility 186
restart processing 161
user tasks 37
BCF Log utility 233
batch create JCL 233
BCFLGUTL program 233
commands 235
compare batch log data sets 237, 241
count control statements 234, 237
create batch backout log data sets 237, 242
find BMP executions 246
find PSB executions 237, 246
FIND report 246
OASN (Origin Application Sequence Number) 236
online backout log create 237, 234
online create JCL 233, 234
PSB keyword 246
return codes 247
statistics 237, 239
Statistics report 239
syncpoint number 236
unit of work 236
UOW keyword 246
BCFACTJ0 program 162
BCFAUTU0 utility 180
BCFBOTRC DD statement 254
BCFDALIN error recovery table 136
BCFDBCON program 248
BCFHIST DD statement 56, 192
BCFPRT DD statement 110, 253
BCFRDER ddname 58
BCFRDER2 ddname 58
BCFSTATS DD statement 56, 193
BCFSYSIN data set
BCFSYSIN DD statement 56, 109
BCFRACE DD statement 254
BCFXPURG utility 186
BCREATE command of BCFLGUTL utility 235
BCSID 48
BCSS
  ID 48
  level 48
BECIDIR module
  overview 61
benefits of BCF 31
&BK0 substitution keyword 133
BKO (Backout Variable) option 104
Block Size field for dynamic allocation 127
BMC Consolidated Subsystem (BCSS)
  ID 48
  level 48
  overview 17
BMC Primary Subsystem (BMCP) 17
BMC Software Product Support 253
BMP RESTART 34
Buff DEV keyword 86
Buffered Device Support option 86
Buffered Tape Unit Name option 78
BUFFTAPE keyword 78
BUFFTAPE device class 115
bypassing conditional backout steps 27

C

cancel intercept processing 24
cataloging log volumes 161
checkpoints, option to ignore 107
Checkwrite Statistics report 201
COMP keyword 105
Compaction field for dynamic allocation 129
COMPARE command of BCFLGUTL utility 235
components of BCF 16
conditional backout steps 27
Conditional Disposition field for dynamic allocation 127
considerations of BCF 34
conventions, documentation 12
&COPY substitution keyword 133, 139
Copy Current DCB from IEFRDER field for dynamic allocation 129
Copy Current Space Allocation field for dynamic allocation 121, 129
Copy Current UNIT from IEFRDER field for dynamic allocation 129
copy of REGISET records 46

D

DASD device class 115
DASD keyword 76
DASD Unit Name option 76
Data Class field for dynamic allocation 129
Data Set IWAIT Detail report 223
Data Set IWAIT Distribution report 222
Data Set IWAIT Summary report 221
data set name skeletons
   examples 135
   overview 132
Data Set Status field for dynamic allocation 127
data sets
   temporary 35
   uncataloged 35
data sets, dynamic allocation of 121
Database Activity reports 212
Database Block Update report 250
Database Buffer Pool report 214
Database Changed Block Reconciliation utility 248
&DAY substitution keyword 133
DBRC
   notification 90
DBRC device class 115
DBRC keyword in BCFACTJ0 164
DBRC, BCF support for 35
DCB copy option 129
DD statements for BCF 56
DDNAME qualifier in record key 43
ddnames
   dynamic allocation of 121
   used by BCF 58
deactivation of BCF 73
DEFAULT device class 115
Defer application restart field 159
Defer Application Restart flag 164, 176
DEFER keyword in BCFACTJ0 164
DELETE command in BCFACTJ0 163
Delete Volume Records option 82
DELVOL keyword 82
&DEPT substitution keyword 133
DEPT (Department Variable) option 105
device classes for dynamic allocation 115
DFSBB00 program 35
disaster recovery 179
DISCARD device class 115
Discard Logging Active option 88, 115
DISLOG keyword 88
DL/I Activity reports 221
DL/I Activity Statistics option 98
DL/I Call IWAIT Summary report 226
DL/I Call reports 225
DL/I Call Summary report 199
DL/I Call Timing Summary report 227
DL/I database organizations supported by BCF 34
DLSTAT keyword 98
documentation information 11
DSN field for dynamic allocation 127
DSN For GDG Model DSCB field for dynamic allocation 129
DTYPE keyword 95
dual SLDS 97
dummy logging environment support 28
dynamic allocation
accessing allocation records 123
automatic space increase option 122
BACKOUT device class 115
BUFTAPE device class 115
conditions 114
Copy Current Space Allocation 121
DASD device class 115
data set name skeletons 132
DBRC device class 115
DEFAULT device class 115
device classes for 115
DISCARD device class 115
error recovery 136
errors 120
Exchange Log Device Type option 91
GDGs 137
global records 115, 120, 123
job step records 115, 120, 125
log data set names 121, 132
log ddnames 121
new log allocation 118
of logs 29
option panels 126
options 127, 129
overview 113
preparing for 113
processing flow 120
records 41, 115
RECOVERY device class 115
skeleton examples 135
STATS device class 115
substitution keywords 133
system records 115, 120, 124
TAPE device class 115
U616 device class 115
UPDATE device class 115
uses of 113
Dynamic Allocation Options panel 127
DYNAMIC keyword 93
dynamic log 96
Dynamic Log option 93
Dynamic Log Type option 95
Dynamic Log Usage Detail report 203

E

Electronic documentation 11
enhanced processing

abend filter 25
cancel intercept 24
overview 23
environment registration record
overview 41
using 53
errors
Physical Logger Error option 81
BCF control statement syntax 110
recovery of dynamic allocation 136
EXCHange keyword 91
Exchange Log Device Type option 91
AUTO 113
device class 115
use during dynamic allocation 120
exclusion
of job steps from BCF participation 59
execution control records
ABENDED status 164
accessing 152
displaying 154
modifying 159
options to modify 159
overview 19, 41, 151
execution controller
activating and using 74
batch backout 84
log recovery 84
overview 19
execution level options 62
choosing this level 64
Execution Options Summary report 229
Expiration Date field for dynamic allocation 129

F

Fast OSAM Buffer Purge feature 30
FIND command of BCFLGUTL utility 235
Force BCF Execution option 59
FORCE keyword of BCFLGUTL utility 235

G

GDGs
&COPY substitution keyword 133, 139
abend of application 140
considerations 149
data set name skeletons 133
different skeleton and log bases 142
dynamic allocation of 137
environments 138
examples 145
Exchange Log Device Type option 143
old log allocation 143
restart of after failure or cancel 142
restrictions on use 144
skeleton setup 139
global dynamic allocation records
  accessing 123
  overview 115
  processing 120
global options record
  accessing 69
  choosing this level 64
  overview 61

H

Help
  online 11
history data set
  accessing BCFHIST data sets 194
  accessing records 194
  allocation 192
  BCF 186
  data set name default 80
  DD statement 56
  displaying records 194
  job step statistics records 192
  log volume records 192
  number of statistics records 100
  overview 192
  recording statistics to 100
History Data Set Name option 80

I

IDRC recording mode 105, 129
IGNorckp keyword 107
Ignore Checkpoints option 107
implementation of BCF 53, 56
IMS Batch Backout utility 233
IMS component of BCSS 17
IMS environment registration record
  overview 41
  using 53
IMS levels supported by BCF 34
IMS program exclusion record
  overview 41
  using 59
IMS program registration record
  overview 41
  using 55
&IMSID substitution keyword 133
IMSID
  qualifier in record key 43
ISPF interface
  limiting a list of records 50
  listing records 50
  overview 45
  setting up a session 45

J

JCL
  changes for BCF 56
  example changes for BCF 56
job step dynamic allocation records 115
  accessing 125
  processing 120
job step options records
  accessing 70
  choosing this level 64
  overview 41, 62
JOBLOG keyword 79
JOBLOG Messages option 79
&JOBN substitution keyword 133
Jobstep Database Data Set Detail report 212
Jobstep Database Detail report 213
Jobstep Database Detail report 213
Jobstep Database Detail report 213
Jobstep reports 197
Jobstep Summary report 198

K

keywords
ABNcode 108
ALERT 102
AUTOObbo 84
BKO 104
BUFFDEV 86
BUFFTAPE 78
COMP 105
DASD 76
DELVOL 82
DEPT 105
DISLOG 88
DLSTAT 98
DTYPE 95
DYNAmic 93
EXCHange 91
IGNorckp 107
JOBLOG 79
LERR 81
MANintov 106
MSLDS 97
NOTIFY 90
PERF 101
RECORD 100
STATS 100
summary table 110
SYSOUT 96
TAPE 77
TBKO 96

Log Recovery % Space Increase field for dynamic allocation 122
log volume records
  automatic deletion 82
  overview 41, 152
  written to history data set 192
Logger Perf % Threshold option 101
Logging Delay Detail report 202
Logging Performance Data report 200
Logical Compare report 241
Logical Record Length field for dynamic allocation 127

Management Class field for dynamic allocation 129
MANintov keyword 106
manual intervention
  Manual Intervention flag 175
  Retry manual intervention field 159, 176
  Set job abend field 177
  situations that require 174
Manual Intervention Override option 106
Max SLDS Block Count option 97
merge of option values 62
messages
  Alert category 102
  automated operations 36
  for problem determination 169
  JOBLOG option 79
  Use Abend Code Series option 108
MSLDS keyword 97
MVSID qualifier in record key 43

nonstandard situations
  manual intervention required 169
  overview 169
  REGISET 178
  system problems 179
Normal Disposition field for dynamic allocation 127
NOTIFY keyword 90
Notify Option for DBRC option 90
Number of Buffers field for dynamic allocation 127
Number of DL/I Calls Between Checkpoints
  Distribution report 208
Number of DL/I Calls Between Checkpoints report 207
Number of Statistic Sets option 100

OCREATE command of BCFLGUTL utility
  input to BCFDBCON 250
  use of 235
Online Backout Log Create report 243
online Help 11
operations
  automated 36
  with BCF 151
options
Alert User ID 102
Automatic Batch Backout 84
BATCH CONTROL FACILITY Active 73
BKO (Backout Variable) 104
Buffered Device Support 86
Buffered Tape Unit Name 78
DASD Unit Name 76
Delete Volume Records 82
DEPT (Department Variable) 105
Discard Logging Active 88, 115
DL/I Activity Statistics 88, 115
Dynamic Log Option 93
Dynamic Log Type 95
Exchange Log Device Type 91, 113, 115, 143
execution level 62
explanation of descriptions 65
Force BCF Execution 59
global level 61
global options record 69
History Data Set Name 80
Ignore Checkpoints 107
job step level 62
job step options records 70
JOBLLOG Messages 79
levels of 61
Logger Perf % Threshold 101
Manual Intervention Override 106
Max SLDS Block Count 97
merging values from different levels 62
Notify Option for DBRC 90
Number of Statistic Sets 100
Physical Logger Error Option 81
Record Activity Stats To 100
Replace IMS 55
setting values for 70
setting with ISPF interface 68
Standard Tape Unit Name 77
summary report 229
summary table sorted by keyword 110
summary table sorted by panel field 66
SYSOUT Class 96
system level 62
system options records 69
Tape BKO 96
Use Abend Code Series 108
Use Execution Controller 74
Use Log Compaction 105
Use Physical Logger 75
OSAM buffers 30
overriding manual intervention 106
P

panel IDs
AESPPRIM 46
ARCPSPF01 49
BCFOPY01 73
BCFOPY02 81
BCFOPY03 95
BCFOPY04 104
BCFPAJ00 152
BCFPAJ0M 159
BCFPPDS0 123
BCFPFP00 194
BCFPFP01 194
BCFPK01 127

panels
Active BMC Consolidated Subsystems 49
Active Job Display 154
Active Job Status Modification 159
Active Jobsteps Display 152
Add Record 51
Application Enhancement Solutions primary menu 46
BATCH CONTROL FACILITY - Active Job Display 154
BATCH CONTROL FACILITY primary menu 48
DB2 element display 45
display options 45
Dynamic Allocation Options 127
IMS element display 45
Limit List of Records 50
List Records 50
Processing Options Page 1 73
Processing Options Page 2 81
Processing Options Page 3 95
Processing Options Page 4 104
Report List by JOBSTEP 194
Report Options 194
Set Dynamic Allocation Skeleton Information 123
using options panels 70
VSAM element display 45
PERF keyword 101
performance threshold 101
Physical Logger

activating and using 75
buffered device support 86
discard logging 88, 90
dual logging 97
dynamic log 93
Dynamic Log Option 96
dynamic log type 95
overview 21
Physical Logger Error option 81
Pre-Execution Options Summary report 229
Primary Log report 209
primary log volume record 41
Primary Quantity field for dynamic allocation 127
printing reports 194
problems
handling 169
product support 253
Processing Options Page 1 panel 73
Processing Options Page 2 panel 81
Processing Options Page 3 panel 95
Processing Options Page 4 panel 104
&PROCSN substitution keyword 133
PROCSTEP qualifier in record key 43
&PROG substitution keyword 133
PROGNAME qualifier in record key 43
program exclusion record
and AES$ssid DD statement 56
overview 41
using 59
program registration record
overview 41
using 55
&PSB substitution keyword 133
PSB UPDATE intent 115
PSB/PGM qualifier in record key 43
PSBNAME qualifier in record key 43
publications, related 11
Purge (BCF records) utility 186

R

reattach of applications 113
Record Activity Stats To option 100
RECORD keyword 100
records
adding 51
AES 41
BCF 41
copying 46
deleting 52
limiting list 50
listing 50
RECOVERY device class 115
recovery, log data sets 115
REGISET
   BCF record purge 186
dynamic allocation records 115, 123
execution control records 19, 151
log information 113
log volume records 152
problems 178
record key qualifiers 43
records for AES 41
records for BCF 18, 41
use in BCF 18, 40
   wildcard characters in record key 44
REGISET Record Copy utility 46
related publications 11
Release Space at Data Set Close field for dynamic allocation 129
Replace IMS option 55
Report List by JOBSTEP panel 194
Report Options panel 194
reports
Active Job utility 166
Application Checkpoint Detail 204
Background Write IWAIT Distribution 225
Background Write IWAIT Summary 223
Batch Backout Log Create 242
BCF Log Utility FIND 246
BCF Log Utility Logical Compare 241
BCF Log Utility Statistics 239
Checkwrite Statistics 201
Data Set IWAIT Detail 223
Data Set IWAIT Distribution 222
Data Set IWAIT Summary 221
Database Activity 212
Database Block Update 250
Database Buffer Pool 214
Database Buffer Pool report 214
DL/I Activity 221
DL/I Call 225
DL/I Call IWAIT Summary 226
DL/I Call Summary 199
DL/I Call Timing Summary 227
Dynamic Log Usage Detail 203
Execution Options Summary 229
Jobstep 197
Jobstep Database Data Set Detail 212
Jobstep Database Detail 213
Jobstep Summary 198
Log Data Set Information 209
Log Data Set Record Detail 211
Logging Delay Detail 202
Logging Performance Data 200
Number of DL/I Calls Between Checkpoints 207
Number of DL/I Calls Between Checkpoints Distribution 208
Online Backout Log Create 243
options summary 229
overview 191
Pre-Execution Options Summary 229
Primary Log 209
saving in the history data set 192
Secondary Log 211
setting report display and printing options 194
Time Between Checkpoints 205
Time Between Checkpoints Distribution 206
types 191
VSAM Database Buffer Pool report 217
restart of applications
cataloging log data sets 161
Defer application restart field 159
dynamic allocation during 113
special processing 169, 174
restart preparation after abend 180
Retention Period field for dynamic allocation 129
Retry manual intervention field 159
RUNDATE keyword in BCFACTJ0 164
RUNTIME keyword in BCFACTJ0 164

S
Secondary Log report 211
Secondary Quantity field for dynamic allocation 127
&SEQ substitution keyword 133
SET command in BCFACTJ0 163
Set Dynamic Allocation Skeleton Information panel 123
Set job abend field 159, 177
skeletons for data set names 132
SLDS, dual 97
SMS
  Data Class 129
  Management Class 129
  SMS Data Set field for dynamic allocation 127
  Storage Class 129
Space Allocation field for dynamic allocation 127
special ddnames used by BCF 58
special restart processing
  Defer Application Restart 176
elements 169
  manual intervention 174
Standard Tape Unit Name option 77
statistics records
  accessing 194
displaying 194
  limiting number of 100
  overview 192
  writing DL/I Activity statistics to 98
  written to history data set 100
STATS command of BCFLGUTL utility 235
STATS device class 115
STATS keyword 100
Status Check utility 39
  status of execution control record
    ABENDED 164
    Set job abend field 159
STEPLIB and uncataloged or temporary data sets 35
STEPLIB DD statement 56
&STEPN substitution keyword 133
Storage Class field for dynamic allocation 129
substitution keywords
    &-type 133
    &COPY 139
      for data set name skeletons 135
    syntax 133
subsystems, use in BCF 17, 39
support for BCF 253
SVC99 retry processing 29
syntax statement conventions 12
SYSOUT Class option 96
SYSOUT keyword 96
system dynamic allocation records
  accessing 124
  overview 115
  processing 120
system options records
  accessing 69
  choosing this level 64
  overview 41, 62
system problems 179
SYSUDUMP DD statement 254

T
Tape BKO option 96
tape data sets, compaction of 129
TAPE device class 115
TAPE keyword 77
tasks for using BCF 37
TBKO keyword 96
temporary data set in STEPLIB 35
temporary file unit name 45
&TIME substitution keyword 133
Time Between Checkpoints Distribution report 206
Time Between Checkpoints report 205
TRAN keyword of BCFLGUTL utility 235

U
U616 device class 115
U626 abend 120
U628 abend 27
U629 abend 27
ULU regions
BCF processing in 35
discard logging not supported 115
limited set of reports 192
uncataloged data set in STEPLIB 35
UNIT copy option 129
Unit Count field for dynamic allocation 127
Unit field for dynamic allocation 127
UOW (unit of work) 236
UPD keyword of BCFLGUTL utility 235
UPDATE device class 115
Use Abend Code Series option 108, 120
Use Execution Controller option 74
Use Log Compaction option 105
Use Physical Logger option 75
&USER substitution keyword 133
user ID 102, 133
utilities
Active Job 161
AES Status Check 39
Automatic Restart Preparation 161, 178, 180
BCF Log 233
BCFAUTU0 180
BCFXPURG 186
Database Changed Block Reconciliation 248
Purge (BCF records) 186
REGISER Record Copy 46

V
Volume Serial field for dynamic allocation 127
VSAM Database Buffer Pool report 217

W
wildcard characters 44