Fast Path Online Suite
User Guide

Supporting

Version 3.11 of Fast Path Online Analyzer/EP
Version 3.11 of Fast Path Online Image Copy/EP
Version 3.11 of Fast Path Online Reorg/EP

January 2014
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

Address: BMC SOFTWARE INC
          2101 CITYWEST BLVD
          HOUSTON TX 77042-2827
          USA

Telephone: 1 713 918 8800 or 1 800 841 2031
Fax: 1 713 918 8000

Outside United States and Canada

Telephone: +01 713 918 8800
Fax: +01 713 918 8000

© Copyright 2000-2014 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.

IBM, IMS, MVS, Redbooks, z/Architecture, 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 UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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

You can obtain technical support by using the BMC Support Central website or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, see “Before contacting BMC.”

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 issues similar to yours and possible solutions
- order or download product documentation
- download products and maintenance
- report an issue or ask a question
- subscribe to receive proactive e-mail alerts when new product notices are released
- 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 issue
- commands and options that you used
- messages received (and the time and date that you received them)
  - product error messages
  - messages from the operating system, such as file system full
  - messages from related software
License key and password information

If you have questions about your license key or password, use one of the following methods to get assistance:

- Send an e-mail message to customer_support@bmc.com.
Contents

About this book

Contents 5

Related publications ................................................................. 17
Conventions ........................................................................... 17
Syntax statements ..................................................................... 18
Summary of changes .................................................................. 18

Chapter 1 Introduction

Overview .................................................................................. 21
Solutions in the Fast Path Online Suite ................................... 22
Fast Path Online Suite capabilities ......................................... 22
Product integration ................................................................. 22
Features and functions ............................................................ 24
Fast Path Online Reorg/EP ...................................................... 24
Fast Path Online Analyzer/EP ................................................ 25
Fast Path Online Image Copy/EP ............................................. 26
Common features ...................................................................... 26
Operational considerations .................................................... 27
More information about Fast Path DEDBs ............................. 27
Additional functionality for the offline environment ............... 27

Chapter 2 System resources and performance

Processing modes ................................................................. 31
Online processing ............................................................... 33
Execution JCL ........................................................................... 33
Region size .............................................................................. 33
STEPLIB requirements .......................................................... 33
Identifying the functions to be performed ............................... 34
Identifying the DEDB area to be processed ............................. 35
Access to online data sets .................................................... 36
IMS commands ........................................................................ 36
Simplifying the command set ................................................ 37
Randomizer module interface ............................................... 38
Dynamically modifying messages .......................................... 38
Permanently modifying messages .......................................... 39
Temporarily modifying messages .......................................... 40
Temporarily restoring permanent message customizations .... 41
Suppressing repetitious messages by suffix type ..................... 42
## Chapter 3  Reorganizing a DEDB online  

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reorganization function overview</td>
<td>44</td>
</tr>
<tr>
<td>Selecting UOWs with patented Intelligent Reorg</td>
<td>44</td>
</tr>
<tr>
<td>User-specified control of the reorganization</td>
<td>45</td>
</tr>
<tr>
<td>Automatic pointer validation during reorganization</td>
<td>45</td>
</tr>
<tr>
<td>Reorganization function restrictions</td>
<td>46</td>
</tr>
<tr>
<td>Control region requirement</td>
<td>46</td>
</tr>
<tr>
<td>Reorganization function inputs and outputs</td>
<td>46</td>
</tr>
<tr>
<td>Selecting the type of reorganization</td>
<td>47</td>
</tr>
<tr>
<td>Online reorganization</td>
<td>48</td>
</tr>
<tr>
<td>Control statement</td>
<td>48</td>
</tr>
<tr>
<td>Performance considerations</td>
<td>48</td>
</tr>
<tr>
<td>Shutting down the reorganization function</td>
<td>50</td>
</tr>
<tr>
<td>MADS considerations</td>
<td>50</td>
</tr>
<tr>
<td>REORGANIZE command keywords and subcommands</td>
<td>51</td>
</tr>
<tr>
<td>Selecting the database and areas</td>
<td>52</td>
</tr>
<tr>
<td>Allocating the area data set</td>
<td>53</td>
</tr>
<tr>
<td>Setting UOW selection criteria</td>
<td>53</td>
</tr>
<tr>
<td>Selectively specifying UOWs for reorganization</td>
<td>53</td>
</tr>
<tr>
<td>Selecting all UOWs for reorganization</td>
<td>55</td>
</tr>
<tr>
<td>Controlling segment compression</td>
<td>56</td>
</tr>
<tr>
<td>Extending IOVF and SDEP storage portions</td>
<td>57</td>
</tr>
<tr>
<td>Extending IOVF</td>
<td>57</td>
</tr>
<tr>
<td>Extending SDEP</td>
<td>59</td>
</tr>
<tr>
<td>Placing segments with load control</td>
<td>60</td>
</tr>
<tr>
<td>Defining segments for load control placement</td>
<td>61</td>
</tr>
<tr>
<td>Placement control using segment content</td>
<td>63</td>
</tr>
<tr>
<td>Placement control using segment counts</td>
<td>65</td>
</tr>
<tr>
<td>Controlling error tolerance</td>
<td>66</td>
</tr>
<tr>
<td>Analyzing the DEDB during reorganization</td>
<td>66</td>
</tr>
<tr>
<td>Making an image copy during reorganization</td>
<td>68</td>
</tr>
<tr>
<td>Sample REORGANIZE command scenarios</td>
<td>68</td>
</tr>
</tbody>
</table>

## Chapter 4  Analyzing a DEDB online  

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis function overview</td>
<td>70</td>
</tr>
<tr>
<td>Selecting the type of analysis</td>
<td>72</td>
</tr>
<tr>
<td>Online analysis</td>
<td>72</td>
</tr>
<tr>
<td>Control region requirement</td>
<td>73</td>
</tr>
<tr>
<td>Inputs and outputs</td>
<td>73</td>
</tr>
<tr>
<td>Control statement</td>
<td>73</td>
</tr>
<tr>
<td>MADS considerations</td>
<td>74</td>
</tr>
<tr>
<td>Concurrent analysis</td>
<td>74</td>
</tr>
<tr>
<td>Inputs and outputs</td>
<td>74</td>
</tr>
<tr>
<td>Control statement</td>
<td>75</td>
</tr>
<tr>
<td>DBRC considerations</td>
<td>76</td>
</tr>
<tr>
<td>MADS considerations</td>
<td>76</td>
</tr>
<tr>
<td>ANALYZE command keywords and subcommands</td>
<td>76</td>
</tr>
<tr>
<td>Selecting the Database and Areas</td>
<td>77</td>
</tr>
</tbody>
</table>
## Contents

<table>
<thead>
<tr>
<th>Chapter 4</th>
<th>Online data extract function overview</th>
<th>110</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Format flexibility</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td>Benefits</td>
<td>111</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 5</th>
<th>Extracting data from an online DEDB</th>
<th>109</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Online data extract function overview</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td>Format flexibility</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td>Benefits</td>
<td>111</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>How online extract works</td>
<td>111</td>
<td></td>
</tr>
<tr>
<td>Control region requirement</td>
<td>112</td>
<td></td>
</tr>
<tr>
<td>Online data extract function inputs and outputs</td>
<td>112</td>
<td></td>
</tr>
<tr>
<td>Process overview</td>
<td>112</td>
<td></td>
</tr>
<tr>
<td>EXTRACT command keywords and subcommands</td>
<td>113</td>
<td></td>
</tr>
<tr>
<td>Selecting the extract file format</td>
<td>114</td>
<td></td>
</tr>
<tr>
<td>EXTRACT_FORMAT=EXTRACT</td>
<td>115</td>
<td></td>
</tr>
<tr>
<td>EXTRACT_FORMAT=HDUNLOAD</td>
<td>117</td>
<td></td>
</tr>
<tr>
<td>EXTRACT_FORMAT=USER</td>
<td>118</td>
<td></td>
</tr>
<tr>
<td>Selecting the database and areas</td>
<td>119</td>
<td></td>
</tr>
<tr>
<td>Defining the extract output data set</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>Creating a separate data set for each area</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>Combining multiple areas into a single data set</td>
<td>122</td>
<td></td>
</tr>
<tr>
<td>Selecting database segments for input and output</td>
<td>124</td>
<td></td>
</tr>
<tr>
<td>Including and excluding segments for input</td>
<td>124</td>
<td></td>
</tr>
<tr>
<td>Narrowing the selection</td>
<td>125</td>
<td></td>
</tr>
<tr>
<td>Outputting segment records</td>
<td>129</td>
<td></td>
</tr>
<tr>
<td>Samples of INCLUDE, EXCLUDE, and OUTPUT subcommands</td>
<td>130</td>
<td></td>
</tr>
<tr>
<td>Controlling segment expansion</td>
<td>132</td>
<td></td>
</tr>
<tr>
<td>Controlling content of segment records</td>
<td>133</td>
<td></td>
</tr>
<tr>
<td>Controlling content of segment record prefixes and suffixes</td>
<td>134</td>
<td></td>
</tr>
<tr>
<td>Controlling header and trailer records</td>
<td>135</td>
<td></td>
</tr>
<tr>
<td>Specifying the generation of header and trailer records</td>
<td>135</td>
<td></td>
</tr>
<tr>
<td>Defining content of header and trailer records</td>
<td>136</td>
<td></td>
</tr>
<tr>
<td>Sorting the extract file</td>
<td>136</td>
<td></td>
</tr>
<tr>
<td>Using the SORT keyword</td>
<td>137</td>
<td></td>
</tr>
<tr>
<td>Using the SORT_OPTION keyword</td>
<td>137</td>
<td></td>
</tr>
<tr>
<td>Sample EXTRACT command scenarios</td>
<td>137</td>
<td></td>
</tr>
</tbody>
</table>

**Chapter 6 Making an online image copy**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to the image copy function</td>
<td>139</td>
</tr>
<tr>
<td>Control region requirement</td>
<td>140</td>
</tr>
<tr>
<td>Image copy function inputs and outputs</td>
<td>140</td>
</tr>
<tr>
<td>IMAGECOPY command keywords and subcommands</td>
<td>140</td>
</tr>
<tr>
<td>Selecting the database and areas</td>
<td>141</td>
</tr>
<tr>
<td>Allocating the image copy data set</td>
<td>143</td>
</tr>
<tr>
<td>Using JCL to allocate the image copy data set</td>
<td>143</td>
</tr>
<tr>
<td>Using dynamic allocation for the image copy data set</td>
<td>143</td>
</tr>
<tr>
<td>Stacking image copy data sets</td>
<td>144</td>
</tr>
<tr>
<td>Generating multiple output image copies</td>
<td>145</td>
</tr>
<tr>
<td>Recording the image copy within DBRC</td>
<td>146</td>
</tr>
<tr>
<td>Compressing the output image copy</td>
<td>146</td>
</tr>
<tr>
<td>Analyzing the area during image copy</td>
<td>147</td>
</tr>
<tr>
<td>Making an image copy during reorganization</td>
<td>149</td>
</tr>
<tr>
<td>Sample IMAGECOPY command scenarios</td>
<td>149</td>
</tr>
</tbody>
</table>
### Chapter 7  Extending a DEDB online

- Extend function overview ........................................ 152
- User-specified control of the extend .......................... 152
- Automatic pointer validation during an extend ............... 152

### Extend function requirements ............................... 152
- Control region requirement ....................................... 152
- DEDB maintenance requirements ............................... 153

### Extend function inputs and outputs ....................... 153
- Selecting the type of extend ..................................... 154
- Online extend .................................................... 154
- Control statement ............................................... 154
- Performance considerations ..................................... 155
- MADS considerations .......................................... 156

### EXTEND command keywords and subcommands ........ 156
- Selecting the database and areas ............................... 157
- Allocating the area data set ..................................... 158
- Specifying the execution mode .................................. 158
- Extending IOVF and SDEP storage portions .................. 158
  - Extending IOVF .............................................. 159
  - Extending SDEP .............................................. 161
- Analyzing the DEDB during an extend ........................ 163
- Making an image copy during an extend ...................... 164
- Sample EXTEND command scenarios ......................... 164

### Appendix A  Diagnostic procedures and tools ............ 167
- Overview ....................................................... 167
- Note software product levels ................................... 168
- Locate generated SVC dump .................................... 168
- Define the general problem ..................................... 170
- Provide duplication and resolution information ............. 171

### Appendix B  Dynamic message modification ............... 183
- Overview ....................................................... 183
- Benefits ....................................................... 184
- Customization capabilities ..................................... 184
- Activation requirement ........................................ 185
- Maintenance considerations .................................... 185
- Message modification subcommands ......................... 185
- Dynamically modifying messages .............................. 186
  - Message severity levels ..................................... 187
  - Messages eligible for dynamic modification ................ 187
  - Changing severity level for a message ...................... 189
  - Suppressing repetitious messages ......................... 189
Listing message customizations ........................................ 190
Restoring customized messages to product defaults ............... 190
Activating message customizations .................................. 191
   Activating customizations by using JCL .......................... 191
   Activating customizations by using dynamic allocation of the catalog on OPTIONS command ........................................ 192
   Activating dynamic allocation of customizations by using DFSMDA member. 192

Appendix C Sample online command scenarios 195

How to interpret the scenarios ........................................ 196
   JCL and control statement ......................................... 196
   Descriptive text .................................................... 196
   Sample library JCL .................................................. 197
   Segment hierarchy for sample DEDB ................................ 197
   DBD for sample DEDB ............................................... 197
Online REORGANIZE command scenarios .............................. 198
   Simultaneously reorganize, analyze, and create image copy 199
   Disable automatic pointer analysis, place dependent segments, and stack image copies ........................................ 200
   Specify UOW selection criteria ..................................... 201
   Extend SDEP ................................................................ 202
Online ANALYZE command scenarios .................................. 203
   Analyze all areas and detect space thresholds .................... 204
   Request repository processing, analyze all areas, and store statistics in repository 205
   Analyze all areas and create online image copies .................. 206
   Analyze a single area and request selected reports ............. 207
Online EXTRACT command scenarios .................................. 207
   Extract and output only selected segments ........................ 209
   Extract to multiple output files and control data selection 210
   Extract all areas and generate a subset of area data .......... 212
   Modify segment layout during extract .............................. 213
   Create a customized extract file format ........................... 214
Online IMAGECOPY command scenarios .............................. 217
   Simultaneously create online image copy and analyze pointers 218
   Disable automatic pointer analysis and stack image copies 219
Online EXTEND command scenarios .................................. 220
   Simultaneously extend SDEP, analyze, and create image copy 221
   Extend SDEP, disable automatic pointer analysis, and stack image copies 221
   Extend IOVF .......................................................... 222

Index 225
Figures

Fast Path Online Suite system flow in online mode ........................................ 23
Sample JCL for multiple Fast Path Online Suite online functions ................. 23
Online mode execution JCL ........................................................................ 33
Online mode execution JCL with RESLIB preceding the product library .......... 34
Online reorganization using DBD and IAREA defaults ................................ 35
Specifying the GLOBAL command ............................................................... 37
Setting a report default .............................................................................. 37
Reorganizing areas with pointer validation and image copy ......................... 38
Specifying message overrides on the PFPOPTS DD statement ..................... 40
Restoring selected permanent message overrides on the PFPOPTS DD statement . 41
Restoring all permanent message overrides on the PFPOPTS DD statement . 41
Using the MESSAGE_SUPPRESSION keyword .......................................... 42
Reorganization function inputs and outputs ............................................... 47
Sample control statement for online reorganization ..................................... 48
Sample control statement for reorganizing all areas .................................... 52
Sample control statement for reorganizing specific areas ............................. 53
Sample control statements for selective UOW reorganization ....................... 53
Sample control statements for selective UOW reorganization based on minimum CIs used ................................................................. 54
Sample control statements for selecting fragmented UOWs ......................... 55
Sample control statements for setting minimum IOVF savings by count ........ 55
Sample control statements for setting minimum IOVF savings by percent ....... 55
Sample control statements for reorganizing All UOWs ............................... 55
Using the COMPRESS keyword with the REORGANIZE command ............. 56
Selectively specifying segments for compression with REORGANIZE .......... 56
Sample control statements for extending IOVF during reorganization .......... 58
Sample control statements for extending SDEP during reorganization ........... 60
Database hierarchical structure in LOADCTL examples ............................. 61
LOADCTL with SEGMENT only ................................................................. 62
Sample LOADCTL for segment dependents ................................................. 62
Sample LOADCTL for segment and its dependents ....................................... 63
Sample LOADCTL basic selection criteria .................................................... 63
Sample LOADCTL using qualified selection criteria ..................................... 64
Using multiple LOADCTL subcommands to control segment placement .... 64
Sample LOADCTL using multiple WHERE statements for the same segment type 65
Sample LOADCTL using segment occurrences ............................................. 65
Sample LOADCTL using segment occurrences and qualified selection criteria ... 65
Using the ERROR_THRESHOLD keyword to control error tolerance ............ 66
Control statements for requesting analysis during reorganization ................. 67
Requesting an image copy during reorganization ....................................... 68
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online analysis function</td>
<td>73</td>
</tr>
<tr>
<td>Example of JCL for an online analysis</td>
<td>73</td>
</tr>
<tr>
<td>Concurrent analysis function</td>
<td>75</td>
</tr>
<tr>
<td>Example JCL for a concurrent analysis</td>
<td>75</td>
</tr>
<tr>
<td>Sample control statement for analyzing all areas</td>
<td>78</td>
</tr>
<tr>
<td>Sample control statement for analyzing specific areas</td>
<td>78</td>
</tr>
<tr>
<td>JCL to allocate a pointer error corrections data set by using the DDNAME keyword</td>
<td>86</td>
</tr>
<tr>
<td>Dynamic allocation of pointer error corrections data set by using DSNNAME keyword</td>
<td>87</td>
</tr>
<tr>
<td>JCL to process the generated output</td>
<td>88</td>
</tr>
<tr>
<td>Sample JCL specifying special reporting to exclude one report</td>
<td>90</td>
</tr>
<tr>
<td>Sample JCL specifying special reporting to include one report</td>
<td>90</td>
</tr>
<tr>
<td>Sample JCL limiting the range of UOWs to be analyzed</td>
<td>91</td>
</tr>
<tr>
<td>Sample JCL for special use of the REPORT subcommand</td>
<td>92</td>
</tr>
<tr>
<td>Specifying the LARGEST_DATABASE_RECORDS keyword</td>
<td>93</td>
</tr>
<tr>
<td>Requesting the record length analysis report simultaneously with specification of number of largest database records</td>
<td>93</td>
</tr>
<tr>
<td>Using the THRESHOLD subcommand with the GLOBAL command</td>
<td>94</td>
</tr>
<tr>
<td>Sample JCL for generating input to the Area Change Modeling Utility</td>
<td>95</td>
</tr>
<tr>
<td>Using the MESSAGE_SUPPRESSION keyword</td>
<td>96</td>
</tr>
<tr>
<td>Using the ORPHANED_SDEP_MSG keyword</td>
<td>96</td>
</tr>
<tr>
<td>Using the HISTORY_DSNNAME keyword</td>
<td>97</td>
</tr>
<tr>
<td>Limiting range of UOWs in record written to history file</td>
<td>97</td>
</tr>
<tr>
<td>Requesting an image copy during analysis</td>
<td>98</td>
</tr>
<tr>
<td>Specifying the analysis function with the IMAGECOPY command</td>
<td>99</td>
</tr>
<tr>
<td>Control statements for requesting analysis during reorganization</td>
<td>100</td>
</tr>
<tr>
<td>Data extract function inputs and outputs</td>
<td>112</td>
</tr>
<tr>
<td>Online extract data control process</td>
<td>113</td>
</tr>
<tr>
<td>Map of the EXTRACT format extract output file</td>
<td>116</td>
</tr>
<tr>
<td>Map of the HDUNLOAD format extract output file</td>
<td>117</td>
</tr>
<tr>
<td>Map of the USER format extract output file</td>
<td>118</td>
</tr>
<tr>
<td>Sample control statement for extracting data from all areas</td>
<td>119</td>
</tr>
<tr>
<td>Sample control statement for extracting data from specific areas</td>
<td>120</td>
</tr>
<tr>
<td>JCL to allocate extract data sets</td>
<td>121</td>
</tr>
<tr>
<td>Dynamic allocation of extract data sets by using OUTPUT_DSN_MASK keyword</td>
<td>122</td>
</tr>
<tr>
<td>JCL to allocate an extract data set by using the DDNAME keyword</td>
<td>122</td>
</tr>
<tr>
<td>Dynamic allocation of extract data sets by using DSNNAME keyword</td>
<td>123</td>
</tr>
<tr>
<td>Using a generation data set group</td>
<td>124</td>
</tr>
<tr>
<td>Using the EXTRACT command with INCLUDE and EXCLUDE subcommands</td>
<td>125</td>
</tr>
<tr>
<td>EXTRACT command using INCLUDE, EXCLUDE, and WHERE</td>
<td>126</td>
</tr>
<tr>
<td>EXTRACT command using INCLUDE statements that identify root keys and invoke the randomizer</td>
<td>127</td>
</tr>
<tr>
<td>Qualifying field references using WHERE keyword</td>
<td>128</td>
</tr>
<tr>
<td>EXTRACT command with SAMPLE_INTERVAL and SAMPLE_LIMIT keywords</td>
<td>128</td>
</tr>
<tr>
<td>Using the OUTPUT subcommand to control generation of segment records</td>
<td>129</td>
</tr>
<tr>
<td>Using the WHERE keyword to conditionally specify segment output</td>
<td>130</td>
</tr>
</tbody>
</table>
SEGA root segment and its dependents .......................................................... 130
Selectively specifying segments for expansion with extract .......................... 132
EXTRACT command with OUTPUT subcommand and FIELDS keyword ........... 133
Using the WHERE keyword to conditionally specify segment content ............ 134
Using SEGMENT_RECORD_PREFIX keyword to control content of segment record
prefix ............................................................................................................. 135
Using SEGMENT_RECORD_SUFFIX keyword to control content of segment record
suffix ............................................................................................................. 135
Using BREAK and FIELDS keywords to create database header and trailer records . 136
Image copy function inputs and outputs ......................................................... 140
Sample control statement for generating image copy of all areas .................. 142
Sample control statement for generating image copy of specific areas .......... 142
JCL to allocate an image copy data set .......................................................... 143
Dynamic allocation of an image copy data set .............................................. 144
Using a GDG ................................................................................................. 144
Using the STACK_NAME keyword ............................................................... 145
Unacceptable use of the same stack name within a single command .............. 145
Specifying multiple image copy data sets for each area ............................... 145
Recording image copies with DBRC ............................................................ 146
Compressing the image copy data set ......................................................... 147
Specifying the analysis function with the IMAGECOPY command ............... 148
Specifying the image copy function with the ANALYZE command .............. 148
Specifying the image copy function with the REORGANIZE command ........ 149
Extend function inputs and outputs ............................................................. 153
Sample control statement for an online extend of IOVF ............................... 154
Sample control statement for extending IOVF for all areas ......................... 157
Sample control statement for extending SDEPs for specific areas ............... 158
Sample control statements for extending IOVF by 50 units of work ............. 160
Sample control statements for extending IOVF by 50 cylinders .................... 160
Sample BMC110001I messages and PPPRPTS information .......................... 160
Sample control statements for extending SDEP by 100 control intervals ...... 162
Sample control statements for extending SDEP by 50 cylinders .................... 162
Control statements for requesting analysis during an extend ....................... 164
Requesting an image copy during extend .................................................... 164
Sample symptom dump ................................................................................ 171
Sample Save Area Trace ............................................................................. 172
Problem summary form ................................................................................ 175
Problem documentation form ....................................................................... 176
Example exception report for common data exceptions ............................... 179
Example of a complete exception report ..................................................... 179
Changing severity level for specific message .............................................. 189
Specifying a suppression threshold for specific message .............................. 189
Listing all message customizations stored in repository catalog ................. 190
Restoring all message customizations to product defaults .......................... 190
Restoring selected message customizations to product defaults .................. 191
Activating message customizations by using JCL ....................................... 191
Activating message customizations by using REPOSITORY_DSNAME with
OPTIONS command ...................................................................................... 192
Activating the repository by using DFSMDA member ............................ 193
JCL and control statement ........................................................................ 196
Segment hierarchy for sample DEDB ....................................................... 197
DBD for sample DEDB ............................................................................. 197
JCL to simultaneously reorganize, analyze, and create image copy ............ 199
JCL to disable automatic pointer analysis, place dependent segments, and stack image copies .............................................................. 200
JCL to specify UOW selection criteria ....................................................... 201
JCL to extend SDEP .................................................................................. 202
JCL to analyze all areas and detect space thresholds ................................... 204
JCL to request repository processing, analyze all areas, and store statistics in repository ................................................................. 205
JCL to analyze all areas and create online image copies .............................. 206
JCL to analyze a single area and request selected reports ........................... 207
JCL to extract and output only selected segments ....................................... 209
JCL to extract to multiple output files and control data selection ................. 210
JCL to extract all areas and generate a subset of area data .......................... 212
JCL to modify segment layout during extract ........................................... 213
JCL to create a customized extract file format ........................................... 214
JCL to simultaneously create online image copy and analyze pointers .......... 218
JCL to disable automatic pointer analysis and stack image copies ............... 219
JCL to simultaneously extend SDEP, analyze, and create image copy .......... 221
JCL to extend SDEP, disable automatic pointer analysis, and stack image copies 221
JCL to extend IOVF ................................................................................. 222
Tables

Synergy offered by Fast Path Online Suite ........................................... 24
Available operational modes by Fast Path/EP command ......................... 32
PFPOPTS OVERRIDE subcommand keywords ........................................ 40
REORGANIZE command keywords and subcommands ................................ 51
ANALYZE command keywords and subcommands ..................................... 76
Fast Path Online Analyzer/EP characteristics, report titles, and keywords ...... 88
Threshold types and keywords ............................................................... 93
EXTRACT command subcommands and keywords .................................... 114
Data extract tailoring options according to value specified for EXTRACT_FORMAT keyword ............................................................. 115
Using INCLUDE, EXCLUDE and OUTPUT subcommands with the EXTRACT command and EXTRACT_FORMAT=EXTRACT or EXTRACT_FORMAT=USER . 130
Using INCLUDE, EXCLUDE and OUTPUT Subcommands with the EXTRACT Command and EXTRACT_FORMAT=HDUNLOAD .................................. 131
Options available for BREAK keyword .................................................... 135
IMAGECOPY command keywords and subcommands ................................ 141
EXTEND command keywords and subcommands ..................................... 156
Reason codes for RC=8 on message BMC110980E ................................... 168
DUMP command set ............................................................................. 177
Expression exception codes .................................................................... 178
Message modification commands and subcommands available with PFPEPR00 ................................................................. 186
Available keywords for the OVERRIDE subcommand ............................... 186
Fast Path Online Suite message severity levels ....................................... 187
Fast Path Online Suite messages available for dynamic modification .......... 187
Descriptive text for JCL ........................................................................... 196
REORGANIZE scenarios ......................................................................... 199
Descriptive text for JCL to simultaneously reorganize, analyze, and create image copy ............................................................................. 200
Descriptive text for JCL to disable automatic pointer analysis, place dependent segments, and stack image copies ............................................... 201
Descriptive text for JCL to specify UOW selection criteria ......................... 202
Descriptive text for JCL to extend SDEP .................................................... 203
ANALYZE scenarios .............................................................................. 203
Descriptive text for JCL to analyze all areas and detect space thresholds ..... 204
Descriptive text for JCL to request repository processing, analyze all areas, and store statistics in repository .................................................. 206
Descriptive text for JCL to analyze all areas and create online image copies .... 206
Descriptive text for JCL to analyze a single area and request selected reports . 207
EXTRACT scenarios ................................................................................ 208
Descriptive text for JCL to extract and output only selected segments ............ 210
Descriptive text for JCL to extract to multiple output files and control data selection . 211
Descriptive text for JCL to extract all areas and generate a subset of area data .... 212
Descriptive text for JCL to modify segment layout during extract ................. 213
Descriptive text for JCL to create a customized extract file format ............... 215
Data output from JCL to create a customized extract file format ................. 216
IMAGECOPY scenarios .............................................................................. 218
Descriptive text for JCL to simultaneously create online image copy and analyze pointers ................................................................. 218
Descriptive text for JCL to disable automatic pointer analysis and stack image copies 219
EXTEND scenarios .................................................................................. 220
Descriptive text for JCL to simultaneously extend SDEP, analyze, and create image copy ................................................................. 221
Descriptive text for JCL to extend SDEP, disable automatic pointer analysis, and stack image copies .................................................. 222
Descriptive text for JCL to extend IOVF ................................................... 222
To meet the growing customer demand for 24x7 availability of data to support business-critical applications, BMC offers the Fast Path Online Suite of products. The products are used by database administrators and technical support personnel who are involved in the management, maintenance, and performance tuning of data entry (DEDB) databases in an IBM® IMS™ environment.

Without taking business-critical databases offline, the Fast Path Online Suite lets you complete a wide range of database maintenance, analysis, and image copying tasks. You can perform many of these tasks simultaneously—without interrupting availability of data to applications that must run continuously.

The Fast Path Online Suite consists of the following products:

- Fast Path Online Reorg/EP
- Fast Path Online Analyzer/EP
- Fast Path Online Image Copy/EP

This book provides detailed procedures for executing the functions that are provided by these products. It also describes how to combine elements of the Fast Path/EP command language to customize these functions to meet your needs.

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 (GUIs).
Related publications

From the BMC Support Central website (http://www.bmc.com/support), you can use either of the following methods to access related publications that support your product or solution:

- Link to the BMC Documentation Center (https://webapps.bmc.com/infocenter/index.jsp) to browse documentation sets, or to view video demos (short overviews of selected product concepts, tasks, or features)

- View individual product documents (books and notices) within the “A – Z Supported Product List”

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.

Conventions

This book 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:

  testsys/instance/fileName

- The symbol => connects items in a menu sequence. For example, Actions => Create Test instructs you to choose the Create Test command from the Actions menu.

- Revision bars in the document mark changes that clarify or correct existing information or that provide new information. Revision bars do not mark editorial changes, formatting changes, or corrections of typographical errors unless these updates significantly affect your use of the information.
Syntax statements

The following example shows a sample syntax statement:

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

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items in italic type represent variables that you must replace with</td>
<td>alias</td>
</tr>
<tr>
<td>a name or value. If a variable is represented by two or more words,</td>
<td>databaseDirectory</td>
</tr>
<tr>
<td>initial capitals distinguish the second and subsequent words.</td>
<td>serverHostName</td>
</tr>
<tr>
<td>Brackets indicate a group of optional items. Do not type the</td>
<td>[tableName, columnName, field]</td>
</tr>
<tr>
<td>brackets when you enter the option. A comma means that you can choose</td>
<td>[-full, -incremental, -level]</td>
</tr>
<tr>
<td>one or more of the listed options. You must use a comma to separate</td>
<td>(UNIX)</td>
</tr>
<tr>
<td>the options if you choose more than one option.</td>
<td>([tableName, columnName, field]</td>
</tr>
<tr>
<td>Braces indicate that at least one of the enclosed items is required</td>
<td>{DBDName</td>
</tr>
<tr>
<td>. Do not type the braces when you enter the item.</td>
<td>UNLOAD device={disk</td>
</tr>
<tr>
<td></td>
<td>fileName</td>
</tr>
<tr>
<td></td>
<td>{-a</td>
</tr>
<tr>
<td>A vertical bar means that you can choose only one of the listed</td>
<td>{commit</td>
</tr>
<tr>
<td>items. In the example, you would choose either commit or cancel.</td>
<td>{-commit</td>
</tr>
<tr>
<td>An ellipsis indicates that you can repeat the previous item or items</td>
<td>columnName ...</td>
</tr>
<tr>
<td>as many times as necessary.</td>
<td></td>
</tr>
</tbody>
</table>

Summary of changes

For detailed information about enhancements, changes, and corrections that are included in your version of the product, see the product release notes. The release notes are available from the BMC Support Central page (http://www.bmc.com/support).
Introduction

The Fast Path Online Suite of products provides online maintenance solutions for IMS data entry databases (DEDBs).

This chapter discusses the following topics:

Overview ................................................................. 21
   Solutions in the Fast Path Online Suite .................................... 22
   Fast Path Online Suite capabilities ........................................... 22
   Product integration ......................................................... 22

Features and functions ....................................................... 24
   Fast Path Online Reorg/EP .................................................... 24
   Fast Path Online Analyzer/EP ................................................. 25
   Fast Path Online Image Copy/EP .............................................. 26

Common features ........................................................... 26

Operational considerations.................................................. 27

More information about Fast Path DEDBs ..................................... 27

Additional functionality for the offline environment ....................... 27

Overview

Applications with high transaction rates demand fast transaction throughput and are quite often the same applications with voluminous storage requirements. Many of the world’s most business-critical applications use IMS Fast Path to take advantages of the transaction rate, large storage capabilities and speed characteristics of DEDBs.

To meet the growing customer demand for 24x7 availability of data to support business-critical applications, BMC has created the Fast Path Online Suite of products to provide more efficient and effective solutions for DEDB analysis, performance, space management, and capacity management. Without taking business-critical databases offline, you can complete a wide range of database maintenance, analysis and image copy tasks. You can also perform many of these tasks simultaneously—without interrupting availability of data to applications that must run continually to serve your business.
Solutions in the Fast Path Online Suite

The Fast Path Online Suite consists of the following products:

- Fast Path Online Reorg/EP
- Fast Path Online Analyzer/EP
- Fast Path Online Image Copy/EP

While these products are available as individual products, they provide significant advantages when applied as a comprehensive online DEDB maintenance suite.

Fast Path Online Suite capabilities

Fast Path Online Suite products create an environment that encourage you to exploit the advantages of DEDBs without the trade-off of downtime associated with traditional batch maintenance processes. The following capabilities are offered with the Fast Path Online Suite of products:

- eliminate manual steps for completing database reorganizations
- perform “intelligent” database reorganizations (reorganize only the portions of the database that actually need reorganizing)
- perform space reclamation online—while the DEDBs are still available to the application
- analyze the content and structure of DEDBs online—while the DEDBs are still available to the application
- create image copies of DEDBs while they are online—without downtime
- significantly reduce I/O through simultaneous reorganization, analysis, and image-copying of a DEDB

Product integration

This section discusses the advantages of combining and using the individual products that comprise the Fast Path Online Suite as a solution suite in an environment where 24x7 data availability is absolutely essential.
Because of the integration among the products in the Fast Path Online Suite, you can execute any combination of reorganization, analysis, and image copy tasks for the same database in a single step. Figure 1 and Figure 2 on page 23 show the functional flow and one-step command JCL that are used for simultaneous execution of DEDB reorganization, analysis, and image copy functions in online mode.

**Figure 1**  Fast Path Online Suite system flow in online mode

**Figure 2**  Sample JCL for multiple Fast Path Online Suite online functions

```plaintext
//PFP  EXEC PGM=DFSRRC00,REGION=0M,
//     PARM=(IFP,dbname,DBF#FPU0)
//STEPLIB  DD DSN=BMC.PFP.LOAD,DISP=SHR
//         DD DSN=IMS.RESLIB,DISP=SHR
//PFPSYSIN DD *
    REORGANIZE DBD=dedb,IAREA=ALL,
        POINTER_VALIDATION=QUICK
    IC DSN='your_dsname',
        UNIT=TAPE,DISP=(CATLG)
/*
```

Table 1 summarizes how the products in the Fast Path Online Suite can be used with each other to save time and resources when performing DEDB database maintenance, analysis, and image–copying tasks.
Features and functions

Table 1  Synergy offered by Fast Path Online Suite

<table>
<thead>
<tr>
<th>Fast Path Online maintenance function</th>
<th>Can concurrently invoke</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>online DEDB reorganization</td>
<td>online DEDB analysis</td>
<td>You can gather statistical information about your Fast Path database area and validate its pointers at the same time you are implementing an online, non-disruptive database reorganization.</td>
</tr>
<tr>
<td>online DEDB reorganization</td>
<td>creation of online image copy</td>
<td>During an online reorganization, you can simultaneously image copy a Fast Path database area.</td>
</tr>
<tr>
<td>creation of online image copy</td>
<td>online DEDB analysis</td>
<td>You can gather statistical information about your Fast Path database area and validate its pointers simultaneously with the creation of an online image copy.</td>
</tr>
<tr>
<td>online DEDB analysis</td>
<td>creation of online image copy</td>
<td>You can gather statistical information about your Fast Path database area and validate its pointers simultaneously with the creation of an online image copy.</td>
</tr>
<tr>
<td>writing of statistics to MAXM Database Advisor for IMS repositories</td>
<td>Fast Path Online Analyzer/EP statistics are written to the Database Advisor repositories if Database Advisor is active.</td>
<td></td>
</tr>
</tbody>
</table>

**Features and functions**

The Fast Path Online Suite of products let you complete a broad range of database maintenance and analysis tasks. This section provides additional detail on the features and benefits of the individual products that comprise the Fast Path Online Suite.

**Fast Path Online Reorg/EP**

The Fast Path Online Reorg/EP product provides reorganization functions that solve the issues of how to reclaim space or extend space in DEDBs while they are online, without database downtime. The utility shares control of the database with IMS during the online reorganization.
Key product functions are summarized as follows:

- The patented Intelligent Reorg feature recognizes and reorganizes only UOWs that require reorganization without taking the database offline. This technique eliminates unnecessary I/O, and more importantly, dramatically reduces logging overhead on the IMS subsystem during an online reorganization. Intelligent Reorg also enables you to fine-tune a DEDB reorganization by selecting only certain UOWs for reorganization with either of the following criteria:
  - only UOWs that extend into IOVF
  - only UOWs that exceed a threshold percent of disorganization

- The online reorganization function lets you optimize performance and reclaim space by processing only UOWs that are disorganized, leaving organized UOWs unaffected.

- The patented online Extend function lets you increase the size of IOVF and SDEP storage portions of a DEDB during reorganization without taking the database offline.

---

**Fast Path Online Analyzer/EP**

The Fast Path Online Analyzer/EP product provides functions that analyze the content and structure of DEDBs while they are online, without database downtime. It lets you verify the integrity of all pointer values, free space element chains, and VSAM control fields, and determine space utilization needs. The utility shares control of the database with IMS during the online analysis. Fast Path Online Analyzer/EP helps you analyze your DEDBs faster, without requiring resource-intensive sorts and work files.

Key product functions are summarized as follows:

- The validation function examines, verifies and reports on pointer data, including segment pointers, root addressable pointers (RAPs), and SDEP pointers.

- The analysis function computes and analyzes information about single or multiple areas. User-specified thresholds can be used to set statistical analysis limits. An area data set or an image copy can be used as input to an analyze function.

- A variety of reports provide comprehensive analysis data to help you monitor and manage performance characteristics, space usage, and physical attributes of DEDBs.

- The online DEDB data extract function lets you extract full or partial segment data from DEDBs while they are online to IMS.

The optional concurrent analysis function enables online analysis of a DEDB with reduced overhead.

Supporting utilities provided with Fast Path Online Analyzer/EP let you dump or modify a database control interval and print a listing of the DMAC.

Fast Path Online Image Copy/EP

Fast Path Online Image Copy/EP lets you take an image copy of the DEDB while the DEDB is online, without any database downtime.

Common features

Several important features that are common to all the Fast Path Online Suite products facilitate product implementation and use:

- For customers who receive the Fast Path Online Suite products on the I-series cartridge tape, installation is accomplished with the BMC Installation System. For more information, see the Installation System User Guide.

- Fast Path Online Suite customers have the option of using the DBA Toolkit, which is included with the MAXM Database Advisor for IMS product on the I-series cartridge tape. Installation is accomplished with the BMC Installation System. For more information, see the Installation System User Guide.

- Fast Path Online Suite products incorporate an easy-to-use command language. Commands, subcommands, keywords, and keyword parameter options provide control and customization of DEDB maintenance and analysis functions. Command sets are composed of a command, optional subcommands, and optional keywords with their parameter values.
Operational considerations

The Fast Path Online Suite products encompass certain requirements:

- IBM z/Architecture® mainframes
  
  BMC licenses each product in the Fast Path Online Suite to run on specific CPUs.

- IBM z/OS 1.10 or later

- supported releases of IMS

- ISPF 3.3 or later

- APF authorization for the STEPLIB containing the Fast Path Online Suite load library

- standard IMS RESLIB (SDFSRESL) for execution

More information about Fast Path DEDBs

For more information about Fast Path DEDBs, see the IBM Redbooks® IMS Fast Path Solutions Guide, reference number SG24-4301-00.

Additional functionality for the offline environment

When you purchase a license for the Fast Path Online Reorg/EP product, you also receive a license for the Fast Path Reorg/EP product. When you purchase a license for the Fast Path Online Analyzer/EP product, you also receive a license for the Fast Path Analyzer/EP product. This integration of batch capabilities with the online solutions provides added benefits for customers that maintain Fast Path databases in online and offline environments.
With the documentation set for the Fast Path Online Suite, you also received the *Fast Path Offline Suite User Guide*, which supports Fast Path Reorg/EP and Fast Path Analyzer/EP. This separate user guide discusses several additional functions offered by Fast Path Reorg/EP and Fast Path Analyzer/EP:

- database change function, which lets you rapidly alter a database, changing the DBD, without I/O-intensive unload and reload of the DEDB

  The function reads the DEDB from DASD or from an image copy data set, and writes the restructured DEDB directly to DASD.

- area change modeling utility that lets you model the effect of a potential DEDB change before actually performing it

- unload and reload functions, which provide the capability to execute the more traditional offline DEDB unload and reload

  When their use is appropriate, the unload and reload functions provide advanced features and controls as compared to traditional unload and reload processing.

- statistics repository, which stores information for immediate retrieval and complete record-keeping of historical DEDB activity gathered by an online or offline analysis process

  User-specified thresholds allow statistical reporting limits and close monitoring of database performance.

- supporting utilities that allow offline control interval dump and modification, reporting on SDEP space usage, and invocation of DEDB randomizing routines

Using Fast Path Reorg/EP with other BMC products, you can simultaneously perform the following multiple tasks:

- reorganize, change or reload the DEDB

- analyze the DEDB (with Fast Path Analyzer/EP) at a user-specified level ranging from checksum pointer validation to full validation

- take one or more image copies of the DEDB

- build or rebuild an index database associated with the DEDB during a change or reload operation (with the BMC Fast Path Indexer/EP product)
Using Fast Path Analyzer/EP with other BMC products, you can simultaneously analyze, reorganize (with Fast Path Reorg/EP), and take one or more image copies of an offline DEDB.

For complete details on these product functions and synergies, see the *Fast Path Offline Suite User Guide*. 
Additional functionality for the offline environment
System resources and performance

This chapter discusses considerations that are related to the use of system resources and execution of the Fast Path Online Suite products. These considerations are prerequisite to optimizing product performance.

This chapter discusses the following topics:

- Processing modes .......................................................... 31
- Online processing ......................................................... 33
  - Execution JCL ............................................................. 33
  - Region size ............................................................... 33
- STEPLIB requirements .................................................. 33
- Identifying the functions to be performed .......................... 34
- Identifying the DEDB area to be processed ........................ 35
- Access to online data sets ............................................... 36
- IMS commands ............................................................ 36
- Simplifying the command set .......................................... 37
- Randomizer module interface ........................................ 38
- Dynamically modifying messages .................................... 38
  - Permanently modifying messages ................................. 39
  - Temporarily modifying messages ................................. 40
- Temporarily restoring permanent message customizations .... 41
- Suppressing repetitious messages by suffix type .................. 42

Processing modes

Each key command-driven function that is provided by the Fast Path Online Suite products can operate in the processing modes that are listed in Table 2. This table also lists products that can execute the command in the indicated mode. Because this document deals only with online processes, the online mode is the only mode that is discussed in this chapter.
NOTE

Because Fast Path Online Reorg/EP includes all offline functionality of the Fast Path Reorg/EP product, Table 2 lists several offline processes that can be executed if you have a license for Fast Path Online Reorg/EP. Because Fast Path Online Analyzer/EP encompasses all offline functionality of the Fast Path Analyzer/EP product, Table 2 indicates that the offline ANALYZE process can be executed if you have a license for Fast Path Online Analyzer/EP. For more information about these offline processes, see the Fast Path Offline Suite User Guide.

<table>
<thead>
<tr>
<th>Command</th>
<th>Mode</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Offline</td>
<td>Online</td>
</tr>
<tr>
<td>ANALYZE</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CHANGE</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>DMAC_PRINT</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>EXTEND</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>EXTRACT</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>IMAGECOPY IC (subcommand)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INITIALIZE</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PFPSORT</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PROCESS_AREA</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RELOAD</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>REORGANIZE</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>RETRIEVE</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>UNLOAD</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

System resource and performance considerations for the execution of the Fast Path Online Suite product functions are dependent on the processing mode, the characteristics of the area being processed, and the optional processing that is requested by using command and subcommand keywords.
Online processing

Online mode is characterized by processing DEDB areas that are online to one (or more) IMS control regions. DEDB areas that are processed by the Fast Path Online Suite product functions are available to applications executing simultaneously.

Execution JCL

Figure 3 shows the basic JCL required to execute a Fast Path Online Suite product function. The functions execute in an IFP utility region.

Figure 3  Online mode execution JCL

```plaintext
//PFP  EXEC  PGM=DFSRRC00,REGION=0M,
  PARM=(IFP,dbname,DFB#FPUO)
//STEPLIB  DD DISP=SHR,DSN=BMC.PFP.LOAD
  DD DISP=SHR,DSN=IMSVS.RESLIB
//PFPSYSIN DD *
--- control statements ---
/*
```

Other parameter values for the PARM keyword might be required for IFP regions at your site. The standard cataloged procedure that is supplied with IMS (FPUTIL) can be used for the Fast Path Online Suite products.

Region size

To ensure optimum performance, BMC recommends that you request the maximum available storage for execution of the Fast Path Online Suite product functions. Maximum storage is normally requested by specifying REGION=0M on the EXEC statement for the JOB step. Some sites might have different requirements. To determine how to request the maximum region size, check with your z/OS systems programmer.

STEPLIB requirements

The Fast Path Online Suite product functions must execute as an APF-authorized program. The product installation library (BMC.PFP.LOAD in all examples) must be identified as an APF-authorized library. All data sets that are listed within the STEPLIB concatenation must also be APF-authorized.
The Fast Path/EP online products require that the installation library containing the BMC-supplied region controller module, BMCRRC00 (alias DFSRRC00), be concatenated preceding the IMS RESLIB data set in the STEPLIB DD statement. The BMC DFSRRC00 module is a replacement of the IMS region controller module.

If the IMS RESLIB precedes the library containing DFSRRC00, you must modify the PGM keyword to execute the BMC–supplied region controller (BMCRRC00) directly, rather than by its alias name (DFSRRC00). JCL for executing BMCRRC00 directly is shown in the example in Figure 4. For more information, see the Database Products for IMS Configuration Guide.

Figure 4  Online mode execution JCL with RESLIB preceding the product library

```plaintext
//PFP EXEC PGM=BMCRRC00,REGION=OM,
//      PARM=(IFP,dbname,DFB#FPUO)
//STEPLIB DD DISP=SHR,DSN=IMSVS.RESLIB
//      DD DISP=SHR,DSN=BMC.PFP.LOAD
//PFPSYSIN DD *
--- control statements ---
/*
```

All example JCL shown in this and other Fast Path Online Suite product documentation assumes that the BMC product library precedes the IMS RESLIB in the STEPLIB concatenation, and that the BMC–supplied region controller module is being executed by its alias name (DFSRRC00).

If any other library precedes the product installation library, you must ensure that this library does not contain a member name that duplicates a member within the PFP product library.

**Identifying the functions to be performed**

The PFPSYSIN control statements identify the functions to be performed. The DBD keyword is optional for online functions and identifies the database (DBD name) to be processed by the command. If the DBD keyword is specified, the DBD name must match the DBD name that is specified in the PARM keyword on the EXEC statement for the JOB step. If you omit the DBD keyword, the command function defaults to the DBD name that is specified in the PARM keyword. Multiple online function commands can be specified within the PFPSYSIN control statements, but they must all refer to the same database. If multiple online function commands do not all refer to the same database, an error message is issued and Fast Path Online Suite processing terminates.
Identifying the DEDB area to be processed

The IAREA keyword identifies the names of the areas that the function will process. You can specify one or more area names, in any order, to be processed by the command. If you omit the IAREA keyword, the function processes all areas that are defined as part of the database. Because of the default values for the DBD and IAREA keywords, you can omit them as shown in Figure 5.

Areas also can be specified on the IAREA keyword by using any combination of area names, area numbers, or area ranges. The following parameters are available for the IAREA keyword:

- **IAREA=ALL (default) or IAREA=: specifies all areas of the DEDB.**

- **IAREA=areaname** specifies one or more areas by using the one-character to eight-character area name for each area specified. Multiple area names must be enclosed in parentheses and separated by commas.

- **IAREA=areanumber** specifies one or more areas by using the one-character to five-character area number for each area specified. Multiple area numbers must be enclosed in parentheses and separated by commas.

- **IAREA=(RANGE=(startarea,endarea))** specifies a consecutive range of areas using either areaname or areanumber parameters. The area number associated with startarea must be less than the area number associated with endarea.

An asterisk (*) can be used to specify all areas of the DEDB. When the * character is used with the RANGE keyword, it can be used to specify the beginning or ending range for specific areas of the DEDB.

Detailed examples of complex IAREA statements are provided in other chapters in this book.
Access to online data sets

The Fast Path Online Suite product functions access the data sets by using IMS control region services. The database definitions (DMB) for the database areas to be processed are obtained from the IMS control region. For this reason, no IMSACB DD statement is required, and any IMSACB DD statement that is supplied is ignored.

All facilities of the online IMS control region are supported, including Multiple Area Data Sets (MADS) and the Virtual Storage Option (VSO). No areaname DD statement is needed, and any areaname DD statement that is supplied is ignored.

The functions access the RECON data sets by using DBRC region services. The RECON1, RECON2, and RECON3 DD statements are not needed, and any of these statements that are supplied are ignored.

IMS commands

IMS/VS commands are available to monitor and control an online DEDB during processing of a Fast Path Online Suite function:

- `/DISPLAY ACTIVE` lets you display the status of all active IMS regions. This display includes the IFP region that is used to execute the Fast Path Online Suite functions.

- `/STOP REGION regid` lets you terminate the Fast Path region at the next utility checkpoint. `Regid` is the region ID number that is assigned to the Fast Path Region. You can determine the `regid` by displaying it with the `/DISPLAY ACTIVE` command.

  **WARNING**

  `/STOP REGION ABDUMP` or `CANCEL` can cause a U113 abend in the IMS control region.

- `/DISPLAY AREA areaname` lets you see the current UOW# that the utility is processing.
Simplifying the command set

Fast Path Online Suite products provide an extensive and powerful command language. Simple keywords and subcommands request available features and functions. Specifying these keywords and subcommands on each command can become repetitious. You can simplify the command set and reduce tedious repetition of keywords and subcommands by setting global values for many common keywords.

If a value for a keyword is specified on the GLOBAL command, the specified value is applied globally on all commands in the command set for which the keyword is valid.

The example in Figure 6 shows how to set a value for the HISTORY_DDNAME keyword that applies to all other commands in the set. The commands to analyze areaname1 and areaname4 inherit this value as if it had been explicitly specified on the ANALYZE commands. Because the command to analyze areaname2 contains an explicit specification for the HISTORY_DDNAME keyword, the value abc overrides the value xyz for that database only.

You can also specify the IC, REPORT, and THRESHOLD subcommands under the GLOBAL command. These subcommands are globally applied as the values on all commands in the command set where they are valid.

The example in Figure 7 shows how to specify a REPORT subcommand that applies to all other commands in the set. The commands to analyze areaname1 and reorganize areaname3 (parallel analysis) inherit this subcommand as if it had been specified explicitly on the commands. Because the command to analyze areaname2 contains an explicit REPORT subcommand of its own, that explicit subcommand overrides the REPORT subcommand on the GLOBAL command.

Figure 6 Specifying the GLOBAL command

| GLOBAL | HISTORY_DDNAME=xyz |
| ANALYZE | IAREA=areaname1 |
| ANALYZE | IAREA=areaname2,HISTORY_DDNAME=abc |
| ANALYZE | IAREA=areaname4 |

Figure 7 Setting a report default

| GLOBAL |
| REPORT DEFAULT=NO,FREE_SPACE_ANALYSIS=YES |
| ANALYZE | IAREA=areaname1 |
| ANALYZE | IAREA=areaname2 |
| REPORT |
| REORGANIZE | IAREA=areaname3,POINTER_VALIDATION=QUICK |
The example in Figure 8 shows how to request the reorganization of three areas. During reorganization, full pointer validation is performed and an output image copy is generated for each area.

Figure 8  Reorganizing areas with pointer validation and image copy

GLOBAL POINTER_VALIDATION=FULL
    IC DSNNAME=‘dataset-name-mask’,DISP=SHR
REORGANIZE IAREA=areaname1
REORGANIZE IAREA=areaname2
REORGANIZE IAREA=areaname3

Randomizer module interface

The randomizer module interface environment conforms to the published interface, with the following exceptions:

- When a randomizer is invoked by IMS, register 10 contains the address of the EPST and register 11 contains the address of the ESCD. Some user-written or user-customized randomizers have been designed to use these addresses to gain access to IMS/VS control blocks other than those that are passed as part of the published interface.

- When a Fast Path Online Suite product issues a call to a randomizing module, register 10 is set to -1 and register 11 is set to 0 to indicate that the call is not being issued in a live IMS environment.

Dynamically modifying messages

Fast Path Online Suite products provide the capability to temporarily or permanently customize eligible messages that are issued by any Fast Path Online Suite primary command. Eligible refers to messages that are available for customization as defined within the product. Temporarily means that you can specify customizations that will apply only to the job input where the changes are requested. Permanently means that you can specify message customizations that will apply to all subsequent Fast Path Online Suite job execution until you choose to return customizations to product defaults.

You might find message customization useful for the following situations:

- to reduce the “nuisance factor” of a message, such as the number of times a message about segment length errors is issued for a very large area
Permanently modifying messages

- to reduce the severity level of a message from E (error) to W (warning) so that issuance of the message will not terminate processing (such as an error message that would otherwise terminate an online reorganization process)

- to increase the severity level of a condition-specific message to force an automatic snap dump in response to a detected condition within the DEDB or area

Permanent changes that you request for eligible messages are stored in the Fast Path/EP statistics repository and are implemented by coding command language executed by the PFPEPR00 repository program. Changes to messages that are specified in this manner will apply as long as they remain stored in the repository.

When you have defined message customizations and stored them in the repository catalog, you must define the repository catalog data set name in your job input to activate message customizations. If you decide later to undo any (or all) message customizations, you can execute PFPEPR00 to return all (or selected) messages to product defaults.

Temporary changes that you request for eligible messages are specified in the JCL under the PFPOPTS DD statement. After job execution, any specified changes will restore the previous settings automatically.

Permanently modifying messages

Using the PFPEPR00 utility, you can change the following functions for eligible messages that are issued by the Fast Path Online Suite products:

- override (change) the default suffix level (informational, warning, error or critical)

- suppress issuance of the specified message by all subsequent primary command processes after reaching a specified occurrence threshold

- restore any (or all) customizations to product defaults

- list all active message customizations you already specified by a previous execution of PFPEPR00

Any customizations that you specify for messages by using the PFPEPR00 utility will apply to all Fast Path Online Suite primary command processes until you execute PFPEPR00 again to change customizations or restore them to product defaults.

For detailed instructions for using the PFPEPR00 utility to specify, store, list, and remove permanent message customizations, see Appendix B, “Dynamic message modification.”
Temporarily modifying messages

Using the PFPOPTS DD statement, you can combine the OPTIONS command and the OVERRIDE subcommand to customize messages for the job input in the associated PFPSYSIN DD statement:

- override (change) the default suffix level (informational, warning, error or critical)
- suppress issuance of the specified message by the primary command process after reaching a specified occurrence threshold

Table 3 shows the functions of the keywords that are available on the PFPOPTS OVERRIDE subcommand to customize messages for the process defined by the associated PFPSYSIN input.

**Table 3  PFPOPTS OVERRIDE subcommand keywords**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>MESSAGE_LEVEL</td>
<td>change default severity level for message</td>
</tr>
<tr>
<td>MESSAGE_LIMIT</td>
<td>set threshold level for suppression of message</td>
</tr>
<tr>
<td>MESSAGE_NUMBER</td>
<td>specify product ID number of message to be customized</td>
</tr>
</tbody>
</table>

You must specify a separate OVERRIDE subcommand for each message you want to customize. Figure 9 specifies two sets of message customization syntax on the PFPOPTS DD statement. The first OVERRIDE subcommand is used to reduce the severity level of message BMC111162 from E (the default) to W. The second OVERRIDE subcommand will suppress the issuance of message BMC111162 after it has been issued 10 times by the reorganization process.

**Figure 9  Specifying message overrides on the PFPOPTS DD statement**

```plaintext
//PFP EXEC PGM=DFSRRC00,REGION=0M,
  //   PARM=(IFP,DB100,DBF#FPU0)
//STEPLIB DD DSN=BMC.PFP.LOAD,DISP=SHR
//     DD DSN=IMSVS.RESLIB,DISP=SHR
//PFPOPTS DD *
  OPTIONS
    OVERRIDE MESSAGE_NUMBER=111162
    MESSAGE_LEVEL=WARNING
    OVERRIDE MESSAGE_NUMBER=121150
    MESSAGE_LIMIT=10
/*
//PFPSYSIN DD *
  REORGANIZE DBD=dbdnam1,IAREA=areaname
/*
```
Temporarily restoring permanent message customizations

If you have already used the PFPEPR00 utility to set permanent modifications for eligible messages, you can specify the RESET subcommand on the PFPOPTS DD statement to temporarily restore any (or all) customized messages to product defaults. RESET can be used to temporarily “override the permanent override” that you stored in the Fast Path/EP repository.

For detailed instructions for using the PFPEPR00 utility to specify, store, list, and remove permanent message customizations, see Appendix B, “Dynamic message modification.”

The example shown in Figure 10 assumes that PFPEPR00 has been executed previously to override the suffix level for message BMC111162, and to set a suppression threshold for message BMC121150. Separate RESET subcommands are specified to restore both messages to product defaults for the specified job input only.

**Figure 10  Restoring selected permanent message overrides on the PFPOPTS DD statement**

```plaintext
//PFP EXEC PGM=DFSRRC00,REGION=0M,
// PARM=(IFP,DB100,DBF#FPU0)
//STEPLIB DD DSN=BMC.PFP.LOAD,DISP=SHR
//      DD DSN=IMSVS.RESLIB,DISP=SHR
//PFPOPTS DD *
  OPTIONS
      RESET MESSAGE_NUMBER=111162
      RESET MESSAGE_NUMBER=121150
/*
//PFPSYSIN DD *
  REORGANIZE DBD=dbdnam1,IAREA=areaname
/*
```

The example shown in Figure 11 assumes that PFPEPR00 has been executed previously to specify customizations (suffix overrides or suppression thresholds) for multiple messages. The parameter ALL is specified on the RESET subcommand to restore all customized messages stored in the repository to product defaults for the specified job input only.

**Figure 11  Restoring all permanent message overrides on the PFPOPTS DD statement (part 1 of 2)**

```plaintext
//PFP EXEC PGM=DFSRRC00,REGION=0M,
// PARM=(IFP,DB100,DBF#FPU0)
//STEPLIB DD DSN=BMC.PFP.LOAD,DISP=SHR
//      DD DSN=IMSVS.RESLIB,DISP=SHR
//PFPOPTS DD *
  OPTIONS
      RESET MESSAGE_NUMBER=ALL
```
Suppressing repetitious messages by suffix type

For each anomaly encountered in an area, the command function might generate warning, error, informational, or critical messages. When the number of places that a particular condition exists is high, a large number of messages are produced.

You can use the MESSAGE_SUPPRESSION keyword on the primary command to set a threshold level for all messages with a certain suffix. You can reduce the number of repetitious messages by specifying a threshold value on the MESSAGE_SUPPRESSION keyword. Use this keyword to specify the maximum number of times that any warning or error message is to be produced before it is suppressed.

In the example in Figure 12, the MESSAGE_SUPPRESSION keyword specifies that any given warning message is to be issued no more than 10 times, that any given error message is to be produced no more than 15 times, and that any given informational message will be issued no more than five times by the reorganization process.

Figure 12   Using the MESSAGE_SUPPRESSION keyword

```
REORGANIZE DBD=dbdname, IAREA=areaname,
  MESSAGE_SUPPRESSION=(WARNING=10,ERROR=15,
                       INFORMATIONAL=5)
```

When messages are suppressed in this way, the Fast Path Online Suite product produces a summary of the number of times that each message has been suppressed.

**WARNING**

The MESSAGE_LIMIT keyword takes precedence over the MESSAGE_SUPPRESSION keyword. In other words, if you specify a suppression threshold for a specific message using the OVERRIDE subcommand (with the MESSAGE_LIMIT keyword) on the PFPOPTS DD statement, the product will comply with this threshold, even if you specify a different threshold for all messages with the same suffix level (with the MESSAGE_SUPPRESSION keyword) on the primary command.
Reorganizing a DEDB online

This chapter provides information on the capabilities, setup and use of the online reorganization function that is provided by Fast Path Online Reorg/EP. This function lets you perform space reclaim and other tasks on DEDBs while they are online to IMS.

This chapter discusses the following topics:

Reorganization function overview .......................................................... 44
   Selecting UOWs with patented Intelligent Reorg .................................. 44
   User-specified control of the reorganization ....................................... 45
   Automatic pointer validation during reorganization ............................ 45
Reorganization function restrictions ....................................................... 46
   Control region requirement ......................................................... 46
Reorganization function inputs and outputs ........................................... 46
Selecting the type of reorganization ..................................................... 47
   Online reorganization ............................................................... 48
   Control statement ................................................................. 48
   Performance considerations ..................................................... 48
   Shutting down the reorganization function .................................... 50
   MADS considerations ............................................................ 50
REORGANIZE command keywords and subcommands ............................. 51
Selecting the database and areas .......................................................... 52
Allocating the area data set ............................................................... 53
Setting UOW selection criteria ............................................................ 53
   Selectively specifying UOWs for reorganization ................................. 53
   Selecting all UOWs for reorganization ......................................... 55
Controlling segment compression ........................................................ 56
Extending IOVF and SDEP storage portions .......................................... 57
   Extending IOVF ..................................................................... 57
   Extending SDEP .................................................................... 59
Placing segments with load control ......................................................... 60
   Defining segments for load control placement .................................. 61
   Placement control using segment content .................................... 63
   Placement control using segment counts ..................................... 65
Reorganization function overview

In some cases, the need to reorganize an area does not require restructuring the area, but simply the elimination of fragmented free space elements (FSEs) and scraps in the root addressable area (RAA), and the movement of as many segments as possible from independent overflow (IOVF) into their respective RAA or DOVF control intervals (CIs). The online reorganization function of Fast Path Online Reorg/EP offers a high-performance alternative for reclaiming fragmented space in a unit of work (UOW) when compared to the traditional reorganization process of unloading and reloading the DEDB area. The online reorganization function, which uses the REORGANIZE command, effectively takes advantage of any free space in the UOW and frees as many IOVF CIs as possible with a minimum of I/O. The reorganization function performs the following tasks:

- reclaims space
- resequences roots
- controls segment placement on the reorganized DEDB
- enables reorganizations to be executed without taking the database offline
- optionally extends the size of the IOVF and SDEP portions of an area during space reclaim without changing the DBD

Selecting UOWs with patented Intelligent Reorg

Certain characteristics of a UOW can determine whether reorganization of the UOW might result in significant performance improvement. Fast Path Online Reorg/EP incorporates the patented Intelligent Reorg feature. Intelligent Reorg selects UOWs based on certain user-specified criteria and reorganizes only those UOWs whose characteristics meet the specified criteria. You can specify criteria to answer the following questions:

- Does the UOW extend into IOVF?
- Is UOW fragmentation over a specified threshold?
- Can reorganization of the UOW save a specified amount of IOVF?

High performance is achieved by reorganizing only UOWs that meet the criteria, which saves time and resources during space reclaim maintenance.
Under ordinary conditions, not all UOWs in an area extend into and use IOVF CIs. The Intelligent Reorg feature selects only UOWs that use IOVF, giving the greatest return with the least I/O to the area. The Intelligent Reorg feature can be turned off, allowing all UOWs to be reorganized, even if the UOW does not extend into IOVF.

Intelligent Reorg examines each UOW to determine whether it is disorganized, so that only UOWs that require reorganization are reorganized. This feature reduces I/O and processing time. To perform space reclaim efficiently, the reorganization function follows these steps:

1. reads each UOW of the selected area or areas
2. examines the UOW to determine level of disorganization
3. reorganizes the UOW only if the UOW is disorganized
4. rewrites the UOW to the same data set only if more than a specified amount of IOVF space was reclaimed

**User-specified control of the reorganization**

User-specified keywords are available to let you control the DEDB reorganization process manually. You can reorganize all areas or selected areas, and you can reorganize all UOWs or selected UOWs by using selected keywords with the REORGANIZE command.

**Automatic pointer validation during reorganization**

If you have a license for the Fast Path Online Analyzer/EP product, the product performs quick (checksum) pointer validation *by default* when you execute the online reorganization function.
Reorganization function restrictions

The following restrictions apply to the online reorganization function:

- The SELECT_UOW keyword enables you to specify the technique to be used by the online reorganization function in the selection of UOWs to be reorganized. The normal default for this keyword is SELECT_UOW=IOVF, which causes reorganization of only UOWs that extend into IOVF. However, if compression is requested (either by specifying COMPRESS=YES or COMPRESS=segment name), the online reorganization function will automatically use a value of SELECT_UOW=ALL.

- When the online reorganization function defaults to SELECT_UOW=ALL because compression is requested by the COMPRESS keyword, the online reorganization function will ignore any value that you specify on the FRAGMENTATION_PERCENT keyword and any alternate value that you specify on the SELECT_UOW keyword. Under this condition, Fast Path Online Reorg/EP will issue message BMC111197I to indicate that all UOWs will be processed.

- You cannot use the online REORGANIZE command to extend IOVF for an area when SDEP segments are defined in the DBD. However, you can use the online EXTEND command to extend IOVF when SDEPs are defined in the DBD. For more information, see Chapter 7, “Extending a DEDB online.”

Control region requirement

When performing an online reorganization, you must define the database to the IMS Control Region with ACCESS=UP.

Reorganization function inputs and outputs

The reorganization function reorganizes the area data set in place, UOW by UOW. For this reason, the same area data set is used as input to and (output from) the reorganization function.

Possible inputs and outputs for an online reorganization function are shown in Figure 13.
The reorganization function always issues a summary report to the report log when reorganization activities are complete. For an example of the Reorganization Report, see the PFPREORG member in the REPORTS data set.

Selecting the type of reorganization

Fast Path Online Reorg/EP reorganizes a DEDB without taking it offline from IMS. Only the UOW that is being reorganized is held exclusively, and it is inaccessible to applications only during the time that it is being reorganized.

Because Fast Path Online Reorg/EP incorporates all batch functionality that is provided by the Fast Path Reorg/EP product, you can also perform offline reorganizations by using Fast Path Online Reorg/EP. For more information, see the Fast Path Offline Suite User Guide.
Online reorganization

The online reorganization process reorganizes a DEDB without taking it offline. Only the UOW that is being reorganized is held exclusively, and it is inaccessible to applications only during the time that it is being reorganized. Performing an online reorganization requires a license for Fast Path Online Reorg/EP.

Control statement

The PFPSYSIN control statements include the command set necessary to run the online reorganization function. A sample control statement for an online reorganization is shown in Figure 14. Reorganization is limited to only the area that is specified with the IAREA keyword.

Figure 14  Sample control statement for online reorganization

```
//PFP      EXEC PGM=DFSRRC00,REGION=0M,
//         PARM=(IFP,dbdname,DBF#FPU0)
//STEPLIB  DD   DSN=BMC.PFP.LOAD,DISP=SHR
//         DD   DSN=IMS.RESLIB,DISP=SHR
//PFPSYSIN DD   *
REORGANIZE DBD=dbdname,IAREA=areaname
/*
```

Performance considerations

This section discusses performance considerations for online reorganization function.

UOW locking

The online reorganization function locks UOWs one at a time while they are being reorganized. UOWs that are selected for reorganization are locked with exclusive access (EX). UOWs that are not selected for reorganization, but are needed for analysis or image copy processing, are locked with shared access (SH).

IMS/VS logging

When a UOW has been reorganized, the lock is released and the reorganization function logs the UOW updates to the IMS/VS log.
Buffer usage

The online reorganization function automatically determines the number of buffers that will be required before reorganization. Automatic determination of buffer requirements is an advantage over other reorganization utilities. Other utilities require the user to estimate the number of buffers that are required before reorganization and will fail if the estimate is insufficient. If you want to allocate and limit the number of buffers used by the online reorganization function because of extended common storage area (ECSA) considerations, use the SELECT_UOW keyword. For more information about using the SELECT_UOW keyword, see “Selectively specifying UOWs for reorganization” on page 53 and the Fast Path/EP Series Reference Manual.

Analyzing the area while reorganizing

If you have a license for the Fast Path Online Analyzer/EP product, the product performs quick (checksum) pointer validation by default during the online reorganization function. The I/O that is required to read the UOW and IOVF control intervals is shared between the two functions.

Because the reorganization process does not normally access SDEP control intervals, additional I/O is required to read these control intervals if you request SDEP pointer validation. This I/O, however, is shared with the image copy function.

For more information, see “Analyzing the DEDB during reorganization” on page 66. For example, you can control the level of pointer validation or turn pointer validation off, depending on your performance considerations.

Creating an output image copy while reorganizing

If you create one or more output image copies while performing an online reorganization, the I/O that is required to read the UOW control intervals is shared between the two functions. A license for the Fast Path Online Image Copy/EP product is required.

Unlike the reorganization process, the image copy process requires the IOVF control intervals to be processed sequentially. The I/O that is required to read the IOVF control intervals for the image copy is not shared. The I/O that is required for processing of the SDEP control intervals is shared between image copy and analysis functions.

You can use the IC subcommand to request creation of an image copy for each area during the reorganization. DBRC is informed (NOTIFY.IC) that a concurrent image copy has been created.
Message suppression

For each anomaly that is encountered in an area, the reorganization function generates a message with a specific suffix (severity) level. When the number of places that a particular condition exists is large, a large number of messages is produced. You can reduce the number of repetitious messages produced by using the MESSAGE_SUPPRESSION keyword to specify the maximum number of times that an informational, warning, error, or critical message is to be produced. For the online reorganization process, this keyword functions in the same manner as it does for the Fast Path/EP online analysis process. For more information, see “Suppressing repetitious messages” on page 95.

Shutting down the reorganization function

When you are running an online reorganization, you can terminate the reorganization function before completion. Use the /STOP REGION regid IMS command.

Because the reorganization function reorganizes one UOW at a time, reorganization that has already been performed does not have to be repeated if the reorganization function is interrupted. UOWs that have already been reorganized remain reorganized. After restart, the reorganization function begins processing at the next disorganized UOW. Special restart JCL is not required.

MADS considerations

Fast Path Online Reorg/EP fully supports multiple area data sets (MADS) for online reorganizations. All online MADS that are registered for each area are reorganized simultaneously.
Table 4 lists the keywords and subcommands that are available for the REORGANIZE command.

Table 4  REORGANIZE command keywords and subcommands

<table>
<thead>
<tr>
<th>Function</th>
<th>Command or Subcommand</th>
<th>Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>selecting the database and area</td>
<td>REORGANIZE</td>
<td>DBD IAREA</td>
</tr>
<tr>
<td>setting UOW selection criteria</td>
<td>REORGANIZE SELECT_UOW</td>
<td>FRAGMENTATION_PERCENT IOVF_SAVE_THRESHOLD</td>
</tr>
<tr>
<td>controlling segment compression</td>
<td>REORGANIZE COMPRESS</td>
<td></td>
</tr>
<tr>
<td>extending IOVF and SDEP storage portions</td>
<td>REORGANIZE EXTEND_IOVF_#UOWS EXTEND_SDEP_#CIS</td>
<td></td>
</tr>
<tr>
<td>placing segments with Load Control</td>
<td>LOADCTL all keywords</td>
<td></td>
</tr>
<tr>
<td>controlling error tolerance</td>
<td>REORGANIZE ERROR_THRESHOLD</td>
<td></td>
</tr>
<tr>
<td>analyzing the DEDB while reorganizing</td>
<td>REORGANIZE REPORT_THRESHOLD POINTER_VALIDATION RAP_VALIDATION SDEP_VALIDATION LARGEST_DATABASE_RECORDS all keywords</td>
<td></td>
</tr>
<tr>
<td>making an image copy of a DEDB while reorganizing</td>
<td>IC all keywords</td>
<td></td>
</tr>
</tbody>
</table>

For more information about the following topics, see the Fast Path/EP Series Reference Manual:

- syntax of commands, subcommands, and keywords that are discussed in this book
- diagrams that show the syntax and available parameters and values for Fast Path Online Suite commands and subcommands
Selecting the database and areas

The DBD keyword identifies the name of the DEDB (DBD name) to be reorganized. For an online reorganization, the DBD name is supplied as an execution parameter, and the DBD keyword can be omitted. If the DBD keyword is coded, it must specify the same DBD name as supplied in the execution parameter.

**WARNING**

If the DBD keyword does not specify the same DBD name as supplied in the execution parameter, an error message is issued and the online reorganization function terminates.

The IAREA keyword can be used to select specific areas to be reorganized. If you omit the IAREA keyword, all areas that are defined in the DEDB are reorganized.

Areas can be specified on the IAREA keyword by using any combination of area names, area numbers, or area ranges. The following parameters are available for the IAREA keyword:

- **IAREA=ALL** (default) or **IAREA=*** specifies all areas of the DEDB.
- **IAREA=areaname** specifies one or more areas by using the one-character to eight-character area name for each area specified. Multiple area names must be enclosed in parentheses and separated by commas.
- **IAREA=areanumber** specifies one or more areas by using the one-character to five-character area number for each area specified. Multiple area numbers must be enclosed in parentheses and separated by commas.
- **IAREA=(RANGE=(startarea,endarea))** specifies a consecutive range of areas using either areaname or areanumber parameters. The area number associated with startarea must be less than the area number associated with endarea.

An asterisk (*) can be used to specify all areas of the DEDB. When the * character is used with the RANGE keyword, it can be used to specify the beginning or ending range for specific areas of the DEDB.

To reorganize an entire DEDB, use a command set like the example shown in Figure 15.

**Figure 15 Sample control statement for reorganizing all areas**

```
REORGANIZE DBD=dbdname, IAREA=ALL
```

To reorganize specific areas, use a command set like the example shown in Figure 16.
Allocating the area data set

For an online reorganization, the area data set is accessed by using IMS control region services. No JCL is needed for the area data set; if present, it is ignored.

Setting UOW selection criteria

You can control which UOWs are selected for reorganization by using the SELECT_UOW keyword. Available values for this keyword are discussed in the following sections.

Selectively specifying UOWs for reorganization

SELECT_UOW=IOVF (the default) selects UOWs based on IOVF usage. When this value is specified, the reorganization function identifies and reorganizes only UOWs that extend into IOVF. By limiting the scope of reorganization, SELECT_UOW=IOVF significantly reduces processing I/O. In the example shown in Figure 17 for database dbdname, the reorganization function considers UOWs for reorganization in area areaname only if the UOW extends into IOVF.

SELECT_UOW=(IOVF, n1) selects UOWs based on a minimum specified number of IOVF control intervals (CIs) used. When this value is selected and you specify a numeric value for n1, the reorganization function will identify and reorganize only UOWs that have used at least that many IOVF CIs. This value can be used to exclude
UOWs from the reorganization that have used a minimum amount of IOVF based on your assessment of CI usage. An example is shown in Figure 18. For database dbdname, the reorganization function considers UOWs for reorganization in area areaname only if the UOW has used at least five IOVF CIs.

SELECT_UOW=(IOVF,n1,n2) selects UOWs based on a minimum and maximum specified number of IOVF control intervals (CIs) used. When this value is selected and you specify a numeric value for n1 and n2, the reorganization function will identify and reorganize only UOWs that have used at least the minimum (n1), and no more than the maximum (n2) number of IOVF CIs. This value can be used to exclude UOWs from the reorganization that have used a minimum and maximum amount of IOVFs based on your assessment of CI usage. SELECT_UOW=(IOVF,n1,n2) also allocates and limits the number of buffers used by the online reorganization function to a maximum value for n2. This feature restricts the amount of ECSA used by the online reorganization function.

NOTE
Specifying a minimum number (n1) and a maximum number (n2) of IOVF CIs on the SELECT_UOW keyword is optional. If only n2 is specified, use the following syntax: (IOVF,,n2).

Figure 18  Sample control statements for selective UOW reorganization based on minimum CIs used

| REORGANIZE DBD=dbdname, AREA=areaname, SELECT_UOW=(IOVF,5) |

Either of the preceding values for the SELECT_UOW keyword can be combined with additional keywords to further limit UOW selection criteria.

The FRAGMENTATION_PERCENT keyword is used to specify the percentage of disorganization that must be detected in a UOW that extends into IOVF to qualify the UOW for reorganization.

NOTE
If you request compression (either by specifying COMPRESS=YES or COMPRESS=segment name) on the REORGANIZE command, any value that you specify for the FRAGMENTATION_PERCENT keyword is ignored. Under this condition, the REORGANIZE command must process all UOWs to ensure that all segments are expanded and recompressed.

An example of the FRAGMENTATION_PERCENT keyword is shown in Figure 19. For DBD dbdname, the command set reorganizes UOWs in areas areanam1 and areanam2 if the UOW extends into IOVF and the minimum user-specified percentage of fragmentation, (2 percent) is detected. Reasonable fragmentation percentage values range from 1 to 5 percent. Fragmentation greater than 5 percent is unlikely under ordinary circumstances.
The `IOVF_SAVE_THRESHOLD` keyword specifies the minimum amount of IOVF that must be saved before the UOW is selected for reorganization. The amount can be specified as the number of CIs or as a percentage of the IOVF that is used by the UOW. Figure 20 specifies that UOWs will not be reorganized unless five or more IOVF CIs are released.

Figure 21 specifies that UOWs will not be reorganized unless 10 percent or more of the IOVF control intervals that are allocated to the UOW are released.

**Selecting all UOWs for reorganization**

`SELECT_UOW=ALL` selects all UOWs for reorganization. This keyword and value specify that all UOWs in an area be selected for reorganization and that all be rewritten. Reorganizing all UOWs increases your reorganization execution time but results in an area with no fragmentation.

**NOTE**

If compression is requested (either by specifying `COMPRESS=YES` or `COMPRESS=segment name`) on the `REORGANIZE` command, the reorganization function will automatically use a value of `SELECT_UOW=ALL`.

An example for `SELECT_UOW=ALL` is shown in **Figure 22**. For database `dbname`, all UOWs in area `areaname` are reorganized.

**Figure 22** Sample control statements for reorganizing All UOWs

```plaintext
REORGANIZE DBD=dbname, IAREA=areaname,
SELECT_UOW=ALL
```
Controlling segment compression

If the BMC DATA PACKER/IMS product is installed in the database that you are reorganizing, you can use the COMPRESS keyword to dynamically expand and compress segment data during the reorganization. SDEPs are excluded because the REORGANIZE command does not process SDEP segments.

The COMPRESS keyword feature lets you change compression techniques or other DATA PACKER/IMS product parameters. For more information, see the DATA PACKER/IMS User Guide or contact BMC DATA PACKER/IMS technical support.

The following values are available for the COMPRESS keyword:

- COMPRESS=NO (the default) causes segment data to be left unchanged.
- COMPRESS=YES causes compressed segment data to be expanded then recompressed.
- COMPRESS=(segment name1, segment name2, ..., segment namen) causes only the segments which are listed by name to be expanded then recompressed.

**NOTE**

When COMPRESS=YES or COMPRESS=(segment name) is specified, the online reorganization function automatically uses the keyword value SELECT_UOW=ALL to ensure that all segments are expanded and recompressed.

In the example in Figure 23, IAREA defaults to ALL. The segments that have a compression exit specified in the DBD are expanded and recompressed.

**Figure 23  Using the COMPRESS keyword with the REORGANIZE command**

```
REORGANIZE DBD=dbdname,COMPRESS=YES
```

**Figure 24** shows an example of using the COMPRESS keyword to selectively specify segment compression for a specified area. Segment1, segment4 and segment5 in area1 are expanded and recompressed only if these segments have a compression exit specified in the DBD. Any other (unspecified) segments that have a compression exit specified in the DBD remain unchanged.

**Figure 24  Selectively specifying segments for compression with REORGANIZE**

```
REORGANIZE DBD=dbdname,IAREA=area1,
       COMPRESS=(segment1,segment4,segment5)
```
Extending IOVF and SDEP storage portions

If an out-of-space condition occurs in IOVF or SDEP, you can use the Fast Path Online Reorg/EP patented Extend feature to extend the IOVF or SDEP portions of a DEDB during a reorganization. The ability to extend the IOVF or SDEP portions of the area during the reorganization process can save time and resources in comparison to using offline CHANGE or UNLOAD/RELOAD commands. The extend feature is available during an online reorganization or an offline reorganization.

If you use the extend feature with an online reorganization and the area is registered with Virtual Storage Option (VSO), the IMS VUNLOAD command must be issued before executing the reorganization.

Depending on how many IOVF or SDEP portions you request, the number of control intervals (CIs) that are required to accommodate your request is rounded up to the next control area (CA) boundary. If additional CIs must be added as a result of the rounding, the additional CIs are added to the SDEP portion of the database.

The following sections present step-by-step procedures for extending IOVF, extending SDEP, and extending IOVF or SDEP without reorganizing the area.

Extending IOVF

The IOVF portion is extended in “UOW units” of CIs. The optional EXTEND_IOVF_#UOWS keyword allows the REORGANIZE command to extend the IOVF portion.

**WARNING**
You cannot use the online REORGANIZE command to extend IOVF for an area with SDEPs defined. However, you can use the online EXTEND command to extend IOVF when SDEPs are defined. For more information, see Chapter 7, “Extending a DEDB online.”

**To extend the IOVF during reorganization**

1. Ensure that adequate space is present on the primary volume of the target area data set to accommodate the request before trying to increase the IOVF portion during the reorganization function. If adequate space is not present, allocate additional volumes to the area data set by using IDCAMS ALTER ADDVOLUME.
Extending IOVF

**NOTE**
If an IDCAMS ALTER ADDVOLUME is used while the area data set is opened and allocated (online to IMS), the data set must be closed and reopened so that VSAM can recognize the added volume. Use the /DBR command to close the data set, and the /STA command to reopen the data set.

**2** Run the reorganization function by using the EXTEND_IOVF_#UOWS keyword to specify the number of UOWs to add.

**3** When the reorganization function is complete, message BMC111193I is issued to the processing log, showing the new ROOT parameter for the DBD source. Use the values that are indicated in message BMC111193I to update the DBD definition to include the IOVF extension. Message BMC110000I is also issued, stating that the extension has been committed.

**Sample control statements for extending IOVF**

Sample control statements for extending IOVF during reorganization are shown in Figure 25.

**Figure 25 Sample control statements for extending IOVF during reorganization**

REORGANIZE DBD=dbdname, IAREA=areaname,
    EXTEND_IOVF_#UOWS=50

**Sample output**

The following example shows messages BMC111193I and BMC110000I that are generated when adding 50 UOWs to the IOVF portion:

BMC111193I <dbdname>, <areaname>: ROOT parameter changed from (200,100) to (250,150)

BMC110000I <dbdname>, <areaname>: AREA extension committed <date>, <time>

**Subsequent maintenance on the DEDB**

When the extend for IOVF has been committed, the in-core DMAC is updated with the increased ROOT parameter. However, because the ACB is not updated with the increased ROOT parameter, you must update the DBD and the ACB with the increased ROOT size.

If the ACB is not updated and the area is ever recovered, reloaded, or reinitialized to a smaller sized IOVF, IMS will refuse to bring the area data set online.
To ensure that the extended IOVF is applied to the DBD definition and that the ACB definition is revised

1. Update the ROOT parameter of the AREA statement in the DBD source with the values that you recorded from message BMC111193I.

2. Run DBDGEN.

3. Run ACBGEN.

---

**NOTE**
You might need to change your area data set VSAM cluster definition to reflect the extended size.

---

### Extending SDEP

The SDEP storage portion is extended in CIs. To extend the SDEP storage area, use the optional EXTEND_SDEP_#CIS keyword with the REORGANIZE command. Specify the number of CIs to add. When SDEP storage is increased, the additional CIs are added to the end of the area.

---

#### To extend SDEP storage

1. Ensure that adequate space is present on the primary volume of the target area data set to accommodate the request before trying to increase the SDEP portion during the reorganization function. If adequate space is not present, allocate additional volumes to the area data set by using IDCAMS ALTER ADDVOLUME.

---

**NOTE**
If an IDCAMS ALTER ADDVOLUME is used while the area data set is opened and allocated (online to IMS), the data set must be closed and reopened so that VSAM can recognize the added volume. Use the /DBR command to close the data set, and the /STA command to reopen the data set.

---

2. Before running the reorganization function with the EXTEND_SDEP_#CIS keyword, BMC recommends that you execute the SDEP Scan/Delete utilities.

---

**NOTE**
If the SDEP storage area is in a logically “wrapped” condition when the extend is performed, the new CIs will appear to be “in use” (even though they are empty) until you execute the SDEP Scan/Delete utilities.
3 Run the reorganization function with the EXTEND_SDEP_#CIS keyword to specify the number of CIs to add.

Sample control statements for extending SDEP during reorganization

Sample control statements for extending SDEP during reorganization are shown in Figure 26.

**Figure 26 Sample control statements for extending SDEP during reorganization**

```
REORGANIZE DBD=dbdname, IAREA=areaname,
              EXTEND_SDEP_#CIS=100
```

Sample output

The following example shows message BMC110000I that is generated when adding control intervals to the SDEP portion:

```
BMC110000I <dbdname>, <areaname>: AREA extension committed <date>, <time>
```

Subsequent maintenance on the DEDB

The next time that you perform maintenance on the DEDB, you might need to change your area data set VSAM cluster definition to allow for the extended size of the SDEP portion.

Placing segments with load control

The normal space search algorithm places segments into the control intervals in a tightly packed manner. The root segment is placed within the RAA block (if room is available) that is determined by the randomizer. Direct dependent segments belonging to the root are placed into the same RAA block or DOVF blocks within the UOW until they are all used. IOVF blocks are selected and filled as needed.

Placement of segments within the types of control intervals (RAA, DOVF, and IOVF) has a significant impact on performance. Consequently, it is sometimes useful to influence the space search algorithm during the reorganization function to relegate infrequently used segments into IOVF (or DOVF) blocks. Segment placement will preserve space in the RAA (or DOVF) blocks for more active data.
Placement of selected segments during the reorganization function can be accomplished by using the LOADCTL subcommand and its related keywords. The LOADCTL subcommand lets you control segment placement in the following ways:

- select a segment, a segment and its dependents, or its dependents only for placement
- control the placement of selected segments into IOVF or DOVF
- control placement of segments, based on their data content
- control placement of segments, based on a specified number of segment occurrences
- preserve space in the RAA (or DOVF) blocks for more active data

**NOTE**

To maintain this data placement, you must continue to use the same LOADCTL subcommand (or subcommands) for any subsequent reorganization processing on the DEDB.

**Figure 27** shows a typical DEDB hierarchy. This hierarchy is used in subsequent examples in this section that show how the LOADCTL subcommand can be used with its associated keywords.

**Figure 27  Database hierarchical structure in LOADCTL examples**

---

**Defining segments for load control placement**

The SEGMENT keyword is required with the LOADCTL subcommand to identify segments for load control placement. The following subparameters are available for the SEGMENT keyword to provide selection versatility:
Defining segments for load control placement

- **SEGMENT=(segname,ONLY)** (the default) specifies that the LOADCTL subcommand applies to the named segment only. Its dependent segments (if any) are not affected.

- **SEGMENT=(segname,DEPENDENTS)** specifies that the LOADCTL subcommand applies to the dependents of the named segment only. The named segment itself is not affected.

- **SEGMENT=(segname,BOTH)** specifies that the LOADCTL subcommand applies to both the named segment and to all of its dependents.

The LOCATION keyword is used to specify where you want the segments placed. The following values are available:

- **LOCATION=IOVF** (the default) places segments in IOVF storage regardless of the amount of space available in RAA and/or DOVF blocks.

- **LOCATION=DOVF** places segments in DOVF storage regardless of the amount of space available in the RAA block. If an inadequate amount of space exists in the DOVF blocks, the segment is placed into IOVF.

### Placing a specific segment type

If you specify the SEGMENT keyword with the ONLY subparameter, the named segment *only* is placed by using load control. For example, in Figure 28, all occurrences of SEGB are placed in IOVF.

**Figure 28  LOADCTL with SEGMENT only**

```
REORGANIZE DBD=dbdname
LOADCTL SEGMENT=(SEGB,ONLY),LOCATION=IOVF
```

### Placing dependents of a segment type

If you specify the SEGMENT keyword with the DEPENDENTS subparameter, all dependents of the named segment are placed by using load control. The named segment is placed normally. In the example in Figure 29, the DEPENDENTS subparameter is specified. Segment placement control is applied to all dependent segment types of the named segment (SEGB). All dependent segment occurrences (SEGC, SEGD, SEGE, and SEGF) are placed within an IOVF block, regardless of the amount of space that might remain within the RAA and DOVF blocks.

**Figure 29  Sample LOADCTL for segment dependents**

```
REORGANIZE DBD=dbdname,IAREA=areaname
LOADCTL SEGMENT=(SEGB,DEPENDENTS)
```
Placing a segment and its dependents

If you specify the SEGMENT keyword with the BOTH subparameter, the named segment and all dependents are placed by using load control. In the example in Figure 30, the BOTH subparameter is specified. Segment placement control is applied to SEGG and its dependent segment (SEGH).

![Figure 30 Sample LOADCTL for segment and its dependents](image)

Placement control using segment content

You can specify that placement control be applied, based on the data values contained in segments, by specifying the WHERE keyword. The WHERE keyword lets you specify field names or field positions and lengths, an operand, a value, and optional Boolean operators. For details about the operations and syntax of the WHERE keyword and detailed syntax rules for coding these expressions, see the Fast Path/EP Series Reference Manual.

Placing specific segments based on segment data

You can specify that placement control be applied depending on values that appear within the named segment data. The example in Figure 31 requests that placement control be applied to the segment type that is specified explicitly by using the SEGMENT keyword. Selection criteria is specified by using the WHERE keyword.

![Figure 31 Sample LOADCTL basic selection criteria](image)

If the criteria is met, SEGB is placed within an IOVF block, regardless of the amount of space that might remain within the RAA and DOVF blocks. If the criteria is not met, the segment occurrence is placed normally.
Placing specific segments based on segment data in a different segment

You can specify that criteria be based on data within a different segment with the WHERE keyword by using a qualified field reference. The qualified field must lie within the segment parentage path from the root segment to the segment specified using the SEGMENT keyword.

The example in Figure 32 requests that placement control be applied to SEGD specified by the SEGMENT keyword. Selection criteria is specified by using the WHERE keyword that references a field within the parent segment (SEGB). If the criteria is met, the SEGD segment occurrences are placed within an IOVF block, regardless of the amount of space that might remain within the RAA and DOVF blocks. If the criteria is not met, the SEGD segments are placed normally.

Figure 32 Sample LOADCTL using qualified selection criteria

| REORGANIZE DBD=dbdname,IAREA=areaname |
| LOADCTL SEGMENT=SEGD,LOCATION=IOVF, |
| WHERE=(SEGB.3:1 EQ 'X') |

Placing multiple segments based on different criteria

You can specify as many LOADCTL subcommands as are required to obtain the desired segment placement. The example in Figure 33 requests that placement control be used for four separate segments. The WHERE keyword is used for each segment to specify a Boolean expression that controls the placement of each segment occurrence.

Figure 33 Using multiple LOADCTL subcommands to control segment placement

| REORGANIZE DBD=dbdname,IAREA=areaname |
| LOADCTL SEGMENT=SEGB,LOCATION=IOVF, |
| WHERE=(3:2 EQ C'80' OR 3:2 EQ C'90') |
| LOADCTL SEGMENT=SEGF,LOCATION=IOVF, |
| WHERE=(SEGA.16:1 EQ C'N') |
| LOADCTL SEGMENT=SEGG,LOCATION=IOVF, |
| WHERE=(16:4P LE P'19970101' AND 20:1 EQ C'Y') |
| LOADCTL SEGMENT=SEGI,LOCATION=IOVF, |
| WHERE=(13:2X GT X'4040') |

If multiple LOADCTL subcommands are specified for the same segment type, they are evaluated in the order in which they are specified. The first LOADCTL subcommand for the segment for which the WHERE criteria is met (or if no WHERE criteria is specified) will be used to control placement.
Figure 34 shows an example of multiple LOADCTL subcommands specified for the same segment. If the first WHERE criteria is met, SEGB is placed in DOVF. If the second WHERE criteria is met, SEGB is placed in IOVF. If neither WHERE criteria is met, the segment is placed normally.

Figure 34   Sample LOADCTL using multiple WHERE statements for the same segment type

```sql
REORGANIZE DBD=dbdname, IAREA=areaname
LOADCTL SEGMENT=SEGB, WHERE=(3:2 EQ '80'), LOCATION=IOVF
LOADCTL SEGMENT=SEGB, WHERE=(3:2 GE '80'), LOCATION=DOVF
```

Placement control using segment counts

You can specify that placement control be applied after a number of segment occurrences for a root segment have been processed by using the INSERT_LIMIT_COUNT keyword. The example in Figure 35 requests that placement control be used for a specified segment type. The first five segments of this type for each root segment are placed normally. The sixth occurrence (and all subsequent occurrences) of the segment type are placed within an IOVF block, regardless of the amount of space that might remain within the RAA and DOVF blocks within the UOW.

Figure 35   Sample LOADCTL using segment occurrences

```sql
REORGANIZE DBD=dbdname, IAREA=areaname
LOADCTL SEGMENT=SEGE, LOCATION=IOVF,
INSERT_LIMIT_COUNT=5
```

The example in Figure 36 combines WHERE and INSERT_LIMIT_COUNT keywords on a single LOADCTL subcommand. All segment types that do not meet the WHERE criteria are placed normally. The first three segments that do meet the WHERE criteria are placed normally. The fourth occurrence (and all subsequent occurrences) of the segment type that meet the WHERE criteria are placed within an IOVF block, regardless of the amount of space that might remain within the RAA and DOVF blocks within the UOW.

Figure 36   Sample LOADCTL using segment occurrences and qualified selection criteria

```sql
REORGANIZE DBD=dbdname, IAREA=areaname
LOADCTL SEGMENT=SEGE, LOCATION=IOVF,
WHERE=(3:2 EQ C'80'),
INSERT_LIMIT_COUNT=3
```
Controlling error tolerance

A UOW can contain a pointer error that prevents the reorganization function from processing root or dependent segment chains. If such an error is encountered in a UOW, the reorganization function terminates.

The `ERROR_THRESHOLD` keyword can be used to enable the reorganization function to encounter and bypass a specified number of UOWs containing pointer errors without terminating. When `ERROR_THRESHOLD` is used with the `REORGANIZE` command, any UOW where a pointer error is detected is not reorganized (remains unchanged). Processing continues with the next eligible UOW. When the number of UOWs where errors are encountered exceeds the value specified on the `ERROR_THRESHOLD` keyword, processing of the area terminates.

When the default value of 0 (zero) is in effect for the `ERROR_THRESHOLD` keyword, reorganization of the area terminates when the first pointer error is encountered.

The example in Figure 37 requests a reorganization and will bypass up to three UOWs containing pointer errors. The UOWs that contain pointer errors are not reorganized. If a fourth UOW that contains a pointer error is encountered, the reorganization is terminated.

![Figure 37 Using the ERROR_THRESHOLD keyword to control error tolerance](REORGANIZE DBD=dbdname,SELECT_UOW=ALL,ERROR_THRESHOLD=3,POINTER_VALIDATION=NONE)

Analyzing the DEDB during reorganization

If your site has a license for the Fast Path Online Analyzer/EP product, pointer validation will occur automatically (by default) when you execute an online reorganization.

This automatic process applies the default value `QUICK` for the `POINTER_VALIDATION` keyword. The analysis function performs a checksum validation of pointers for each segment type within each UOW of the database that is specified on the `REORGANIZE` command. Automatic pointer validation provides assurance of the area’s pointer integrity, while providing statistics that show how the reorganization process affected space usage.
You can manually specify any valid value for the SDEP_VALIDATION keyword to control how SDEP pointers are to be validated.

You can also specify additional functions that are associated with the analysis function:

- Use the RAP_VALIDATION keyword to control how RAPs are processed.
- Use the LARGEST_DATABASE_RECORDS keyword to specify the number of largest database records to be tracked by the analysis process.
- Use the REPORT and THRESHOLD subcommands to control the generation of analysis reports and exception testing.

For more information about these keywords and subcommands, see Chapter 4, “Analyzing a DEDB online.”

Because no values are specified for the POINTER_VALIDATION and SDEP_VALIDATION keywords, the example in Figure 38 requests that pointer validation be performed during reorganization of the area by using the quick (checksum) technique and that no validation of SDEP pointers is performed. Full validation for RAP pointers is requested explicitly. The example produces only the Free Space Analysis report and does not check threshold conditions.

**Figure 38**  Control statements for requesting analysis during reorganization

```
REORGANIZE DBD=dbname.
   RAP_VALIDATION=XREF.
REPORT
   REPORT_DEFAULT=NO.
   FREESPACE_ANALYSIS=YES
```
Making an image copy during reorganization

You can request that one or more output image copy data sets be produced during reorganization. Specify the IC subcommand for each image copy to be produced. For details about the IC subcommand, see Chapter 6, “Making an online image copy.”

When an image copy is requested during an online reorganization, a license for Fast Path Online Image Copy/EP is required.

The example in Figure 39 requests the creation of a single output image copy data set for the area being reorganized.

Figure 39  Requesting an image copy during reorganization

```sql
REORGANIZE DBD=dbname,IAREA=areaname
   IC DSNAME='dataset-name-mask',
   UNIT=TAPE,DISP=(NEW,CATLG)
```

Sample REORGANIZE command scenarios

For scenarios that show how to use the online REORGANIZE command with key related keywords and subcommands, see Appendix C, “Sample online command scenarios.”
Analyzing a DEDB online

This chapter provides information on the capabilities, setup, and use of the online analysis function that is provided by Fast Path Online Analyzer/EP. This function lets you analyze and validate the content and structure of DEDBs while they are online to IMS. It also lets you perform reporting and related tasks.

This chapter discusses the following topics:

- Analysis function overview .................................................. 70
- Selecting the type of analysis .................................................. 72
- Online analysis ...................................................................... 72
  - Control region requirement ...................................................... 73
  - Inputs and outputs ................................................................. 73
  - Control statement ................................................................. 73
  - MADS considerations ............................................................ 74
- Concurrent analysis ............................................................... 74
  - Inputs and outputs ................................................................. 74
  - Control statement ................................................................. 75
  - DBRC considerations ............................................................. 76
  - MADS considerations ............................................................ 76
- ANALYZE command keywords and subcommands .................... 76
- Selecting the Database and Areas .............................................. 77
- Allocating the area data set ..................................................... 78
  - Online analysis ................................................................. 78
  - Concurrent analysis ............................................................. 79
- Verifying area integrity ........................................................... 79
  - Physical validation ............................................................... 79
  - Pointer validation ............................................................... 80
  - RAP validation ..................................................................... 82
  - SDEP validation .................................................................... 82
- Correcting pointer errors ......................................................... 84
  - Restrictions for correcting pointer errors ............................... 85
  - Allocating the pointer error corrections data set ...................... 86
  - Processing the generated output ........................................... 87
The Fast Path Online Analyzer/EP analysis function lets you gather the data that is required for assessing data entry database (DEDB) capacity and performance characteristics and for making DEDB maintenance decisions. Using the ANALYZE command, subcommands, and keyword parameters, you can perform the following tasks:

- analyze information about single or multiple areas
- validate pointer data
- generate reports
- set thresholds for exception condition detection
- retain data in a statistical repository
- retrieve the data for historical reporting
- generate input to the Area Change Modeling process

The analysis function analyzes and validates the content and structure of DEDBs. The analysis function includes the following features:

- four types of analysis
  - offline
  - online
  - concurrently
  - with image copy as input

- user-specified keywords

  Keywords and keyword parameters let you control and customize the analysis.

- pointer validation

  The validation function examines and validates pointer data during every analysis.

- reporting

  The analysis function computes statistics and analyzes information about single or multiple areas and presents the results in reports.

- thresholds

  User-specified values set threshold criteria by which to perform analysis to detect exception conditions.

- statistics repository

  The statistics repository can store information for historical reporting.

Fast Path Online Analyzer/EP includes the following utilities that can use input from the analysis function to improve database performance and integrity:

- Area Change Modeling Utility

  The Area Change Modeling utility lets you model a potential DEDB change before performing it.

- SDEP Space Utilization Utility

  The SDEP Space Utilization Utility lets you track the utilization of the SDEP portions of your area.
For more information about these utilities, see the *Fast Path/EP Series Reference Manual*.

## Selecting the type of analysis

Fast Path Online Analyzer/EP lets you perform the analysis function in the following ways:

- offline
- online
- concurrent
- with image copy as input

This chapter presents information on performing online analysis and concurrent analysis.

Because Fast Path Online Analyzer/EP incorporates all batch functionality that is provided by the Fast Path Analyzer/EP product, you can also perform offline analysis by using Fast Path Online Analyzer/EP. For information about performing an offline analysis or analysis of an image copy, see the *Fast Path Offline Suite User Guide*.

## Online analysis

The Fast Path Online Analyzer/EP product can read and analyze a DEDB while the DEDB is online. Online analysis operating characteristics are as follows:

- The DEDB is online to IMS when analysis occurs.
- Read-with-integrity access (shared locks) is used. IMS services are used to read the DEDB.
- Other applications can update the DEDB while analysis is in progress.

Fast Path Online Analyzer/EP locks each UOW briefly while it is read and analyzed. Another application momentarily cannot read that one UOW; however, other applications can access the remainder of the database during that time. A UOW that is being processed by an application is locked by the IMS system, preventing Fast Path Online Analyzer/EP from accessing the UOW until it is unlocked by the application.
Control region requirement

When performing an online analysis, you must define the database to the IMS Control Region with ACCESS=UP.

Inputs and outputs

Figure 40 shows the inputs and outputs for the online analysis function.

![Diagram of online analysis function]

Control statement

A sample control statement for an online analysis is shown in Figure 41. Analysis is limited only to the area that is specified with the IAREA keyword.

```
//PFP EXEC PGM=DFSRRC00,REGION=0M,
   PARM=(IFP,dbname,DFB#FP0)
//STEPLIB DD DSN=BMC.PFP.LOAD,DISP=SHR
//@ DD DSN=IMS.RESLIB,DISP=SHR
//@PFPSSYN DD *
   ANALYZE DBD=dbname,IAREA=areaname
/*
```
MADS considerations

Multiple area data sets (MADS) are fully supported by the online analysis process.

Concurrent analysis

Fast Path Online Analyzer/EP can also read and analyze a DEDB that is online to the IMS system without using any IMS system resources. This lower overhead method of operation is called concurrent analysis.

Concurrent analysis operating characteristics are as follows:

- The DEDB is online during analysis.
- Read-without-integrity access is used.
- Unprotected updates can occur during analysis.
- Performance of the online IMS system is not affected.

During concurrent analysis, the DEDB might be under the active control of an IMS online control region, but IMS services are not used to read the area. If updates occur during analysis, the analysis function might report transient errors as a result of analysis and update lag time. Running a concurrent analysis again probably will not report the same errors. Subsequent concurrent analysis runs can help determine whether any reported errors resulted from such transient conditions.

Inputs and outputs

Figure 42 shows the inputs and outputs for the concurrent analysis function.
To perform concurrent analysis of a DEDB, you must specify the ACCESS=CONCURRENT keyword. DBRC must also be active during this analysis.

A sample control statement for concurrent analysis is shown in Figure 43. In this example, the area data set, ACB library, and RECON data sets are accessed by using dynamic allocation. Analysis is limited only to the area that is specified with the IAREA keyword.

```plaintext
//ANALYZE EXEC PGM=PFPMAIN,REGION=0M
//STEPLIB DD DSN=BMC.PFP.LOAD,DISP=SHR
//         DD DSN=IMS.RESLIB,DISP=SHR
//          DD DSN=IMS.RESLIB,DISP=SHR
//PFPSYSIN DD *
  GLOBAL DBRC=YES
END
ANALYZE DBD=dbdname,IAREA=areaname.
  ACCESS=CONCURRENT
END
/*
```
DBRC considerations

DBRC must be active during execution of the concurrent analysis function. If the area is registered, a read-without-integrity authorization level for the area data set is requested. The data set name must match the name that is registered with DBRC.

MADS considerations

Multiple area data sets (MADS) are not supported by the concurrent analysis process. The product searches the area data set (ADS) list that is registered for each area (in collating sequence by DD name). The product selects the first ADS that is marked as available for use and that has no error queue elements (EQEs). If an ADS is found that meets both of these criteria, it is the only ADS is analyzed. All other area data sets are ignored.

ANALYZE command keywords and subcommands

Table 5 lists the keywords and subcommands that are available for the ANALYZE command.

<table>
<thead>
<tr>
<th>Function</th>
<th>Command or Subcommand</th>
<th>Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>selecting the database and areas</td>
<td>ANALYZE</td>
<td>DBD IAREA</td>
</tr>
<tr>
<td>allocating the area data set</td>
<td>ANALYZE</td>
<td>INPUT_DSN_MASK</td>
</tr>
<tr>
<td>selecting the mode of analysis</td>
<td>ANALYZE</td>
<td>ACCESS</td>
</tr>
<tr>
<td>verifying area integrity</td>
<td>ANALYZE</td>
<td>POINTER_VALIDATION RAP_VALIDATION SDEP_VALIDATION ORPHANED_SDEP_MSG</td>
</tr>
<tr>
<td>correcting pointer errors</td>
<td>CORRECTIONS_FILECTL</td>
<td>DDNAME DSNAME</td>
</tr>
<tr>
<td>specifying the number of largest database records to be tracked during analysis</td>
<td>ANALYZE</td>
<td>LARGEST_DATABASE_RECORDS</td>
</tr>
<tr>
<td>generating analysis reports</td>
<td>REPORT</td>
<td>all associated keywords</td>
</tr>
</tbody>
</table>
Selecting the Database and Areas

The DBD keyword identifies the name of the DEDB (DBD name) to be analyzed. For online analysis, the DBD name is supplied as an execution parameter, and the DBD keyword can be omitted. If the DBD keyword is coded for an online analysis, it must specify the same DBD name as supplied in the execution parameter.

**WARNING**

If the DBD keyword does not specify the same DBD name as supplied in the execution parameter, an error message is issued and the online analysis function terminates.

The IAREA keyword can be used to select specific areas to be analyzed. If you omit the IAREA keyword, all areas that are defined in the DEDB are analyzed.

Areas can be specified on the IAREA keyword by using any combination of area names, area numbers, or area ranges. The following parameters are available for the IAREA keyword:

- IAREA=ALL (default) or IAREA=* specifies all areas of the DEDB.

### Table 5  ANALYZE command keywords and subcommands (part 2 of 2)

<table>
<thead>
<tr>
<th>Function</th>
<th>Command or Subcommand</th>
<th>Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>detecting exception conditions</td>
<td>THRESHOLD</td>
<td>all associated keywords</td>
</tr>
<tr>
<td>generating input for the Area Change Modeling utility</td>
<td>ANALYZE MODEL_DDNAME</td>
<td></td>
</tr>
<tr>
<td>suppressing repetitious messages</td>
<td>ANALYZE MESSAGE_SUPPRESSION ORPHANED_SDEP_MSG</td>
<td></td>
</tr>
<tr>
<td>using a history file</td>
<td>ANALYZE HISTORY_DDNAME</td>
<td></td>
</tr>
<tr>
<td>enhancing performance</td>
<td>ANALYZE ICACHE</td>
<td></td>
</tr>
<tr>
<td>producing an image copy during analysis</td>
<td>IC</td>
<td>all associated keywords</td>
</tr>
</tbody>
</table>

For more information about the following topics, see the *Fast Path/EP Series Reference Manual*:

- syntax of commands, subcommands, and keywords that are discussed in this book
- diagrams that show the syntax and available parameters and values for Fast Path Online Suite commands and subcommands

### Selecting the Database and Areas

The DBD keyword identifies the name of the DEDB (DBD name) to be analyzed. For online analysis, the DBD name is supplied as an execution parameter, and the DBD keyword can be omitted. If the DBD keyword is coded for an online analysis, it must specify the same DBD name as supplied in the execution parameter.

**WARNING**

If the DBD keyword does not specify the same DBD name as supplied in the execution parameter, an error message is issued and the online analysis function terminates.

The IAREA keyword can be used to select specific areas to be analyzed. If you omit the IAREA keyword, all areas that are defined in the DEDB are analyzed.

Areas can be specified on the IAREA keyword by using any combination of area names, area numbers, or area ranges. The following parameters are available for the IAREA keyword:

- IAREA=ALL (default) or IAREA=* specifies all areas of the DEDB.
Allocating the area data set

- IAREA=areaaname specifies one or more areas by using the one-character to eight-character area name for each area specified. Multiple area names must be enclosed in parentheses and separated by commas.

- IAREA=areanumber specifies one or more areas by using the one-character to five-character area number for each area specified. Multiple area numbers must be enclosed in parentheses and separated by commas.

- IAREA=(RANGE=(startarea,endarea)) specifies a consecutive range of areas using either areaname or areanumber parameters. The area number associated with startarea must be less than the area number associated with endarea.

An asterisk (*) can be used to specify all areas of the DEDB. When the * character is used with the RANGE keyword, it can be used to specify the beginning or ending range for specific areas of the DEDB.

To analyze an entire DEDB, use a command set like the example shown in Figure 44.

**Figure 44  Sample control statement for analyzing all areas**

```
ANALYZE DBD=dbdname,IAREA=ALL
```

To analyze specific areas, use a command set like the example shown in Figure 45.

**Figure 45  Sample control statement for analyzing specific areas**

```
ANALYZE DBD=dbdname,IAREA=(AREANAM1,AREANAM3,RANGE=(5,8))
```

This control statement uses a combination of area names and area numbers to request that the specified areas are to be analyzed. The areas named AREANAM1 and AREANAM3 will be analyzed. Also, all consecutive areas from area number 5 to area number 8 (area5, area6, area7 and area8) will be analyzed.

### Allocating the area data set

This section discusses allocation of the area data set for the online and concurrent analysis methods.

### Online analysis

For online analysis processing, the area data set is accessed by using IMS control region services. No JCL is needed for the area data set; if present, it is ignored.
Concurrent analysis

For concurrent analysis, the area data set can be supplied in the JCL, or accessed by using dynamic allocation.

The areaname DD statement identifies the area data set to be analyzed. If the areaname DD statement is omitted from the JCL, Fast Path/EP tries to dynamically allocate it.

The IMSACB DD statement identifies the ACB library containing the database definition that describes the area which is referenced by the areaname DD statement.

If you are using dynamic allocation, do not include the areaname DD statement. Fast Path/EP tries to obtain the data set name for allocation in the following order:

1. If the INPUT_DSN_MASK keyword is specified, it is used to generate the data set name.
2. If DBRC is active and the area is registered, the registered area data set name is obtained from DBRC.
3. The STEPLIB is searched for the DFSMDA member that contains the data set name for this area.

If the area is registered with DBRC, the allocated data set name must match the registered data set name, regardless of how the area is allocated.

Verifying area integrity

The analysis function validates the physical and logical integrity of the area. Physical verification is the process of validating that the area can be read and interpreted correctly. Logical verification is the process of validating that the pointer chains among segment types are correct.

Physical validation

Each time you execute an analysis process, the physical integrity of the area is verified. Physical validation performs the following tasks:

- validates that the CIs are of the proper type and that the CIs are in the proper sequence
■ validates the IMS overhead bytes in the CI prefix, and validates the VSAM overhead bytes in the CI suffix

■ validates the free space element (FSE) chains

■ validates that each RAP, DOVF, IOVF, and SDEP CI is filled completely with segments, FSEs, and scraps

■ validates each allocation table entry and the free block chain within each Space Map CI

■ validates DOVF free block chains, and cross-validates each IOVF CI with the corresponding entry in the Space Map CIs

■ checks that each pointer to a root or direct dependent segment references a location within the direct portion of the area

■ checks that PCL and SSPTTR pointers are null if the corresponding PCF pointer is null

■ checks that the PCL pointer is not null if the corresponding PCF pointer is not null

■ checks that each pointer to a sequential dependent segment references a location within the sequential portion of the area

■ checks that the complete 8-byte pointer references a position within the valid logical bounds of the sequential dependent portion of the area

**Pointer validation**

The analysis function validates the logical integrity of pointers to the root and direct dependent segments. You can specify an implicit value for the POINTER_VALIDATION keyword in the GLOBAL command. This keyword value is used by all analysis functions within the job step. To override the GLOBAL command value, specify the POINTER_VALIDATION keyword explicitly on the ANALYZE command.

■ POINTER_VALIDATION=FULL constructs a complete cross-reference of every pointer and segment occurrence. When POINTER_VALIDATION=FULL, the analysis process performs the following tasks:

— checks every pointer to ensure that a segment of the proper type actually appears at the referenced relative byte address (RBA)
— checks that every segment is referenced by one and only one physical child first (PCF) or physical twin forward (PTF) pointer

— identifies the complete chain of dependent segment occurrences associated with their parent segment (the PCF/PTF pointers)

— checks physical child last (PCL) and subset (SSPTR) pointers to verify that the segments which are referenced appear in the chain that is defined by the PCF/PTF pointers

— checks all keyed segments for correct key sequencing to ensure that the sequence keys of logically adjacent segments are ascending and unique

**POINTER_VALIDATION=QUICK** (the default) generates a checksum for each segment type within each unit of work (UOW). The checksum consists of the sum of the RBAs for each segment occurrence minus the RBAs of all PCF/PTF pointers to the segment type. Because there should be exactly one PCF/PTF pointer to each segment, a non-zero value indicates a pointer problem. A second checksum consists of the sum of the RBAs for each segment with a null PTF pointer to each segment minus the RBAs of all PCL pointers to the segment type. A non-zero value indicates a problem. SSPTR pointers are not validated in this mode.

**POINTER_VALIDATION=OFF** bypasses logical integrity verification of pointers to the root and direct dependent segments. The analysis function still reads the area to perform physical validation and to gather statistics, and provides the same reporting as **POINTER_VALIDATION=QUICK**.

**NOTE**
Because many statistics that are required for report generation can be collected only when **POINTER_VALIDATION=FULL**, certain reports are not produced if other values are selected. The following reports will not be generated if **POINTER_VALIDATION=QUICK** or **POINTER_VALIDATION=OFF** is selected:

- Segment I/O Analysis Report
- Record Length Analysis Report
- Record Placement Analysis Report

**POINTER_VALIDATION=NONE** specifies that the analysis function is not to be executed. This value is valid only on **GLOBAL**, **REORGANIZE**, **RELOAD**, **CHANGE**, and **IMAGECOPY** commands; it is not valid when used with the **ANALYZE** command. This value is used to completely cancel the analysis process as an override of the **POINTER_VALIDATION=QUICK** default, which otherwise occurs during these processes.
**RAP validation**

The method that is used to verify the logical integrity of root anchor point (RAP) and root segment pointers can be controlled independently, using the RAP_VALIDATION keyword. You can specify an implicit value for the RAP_VALIDATION keyword on the GLOBAL command. This keyword value is used by all analysis functions within the job step. To override the GLOBAL command value, explicitly specify the RAP_VALIDATION keyword on the ANALYZE command:

- RAP_VALIDATION=(NOXREF,) (the default) validates RAP and root segment pointers in the same manner as that specified by the POINTER_VALIDATION keyword.

- RAP_VALIDATION=(XREF,) constructs a complete cross-reference of every RAP and root segment occurrence. The analysis function checks every pointer to ensure that a root segment is displayed at the referenced location. The complete chain of root segments that are associated with the RAP is identified. The analysis function checks that every root segment occurrence is referenced by one (and only one) pointer.

- With RAP_VALIDATION=(NOPLACEMENT) (the default), the placement of root segments is not validated.

- With RAP_VALIDATION=(PLACEMENT), the placement of every root segment is validated by calling the randomizer (if available) to confirm the following placement:
  - The root segment is located in the proper area.
  - The root segment is located in the proper RAP.

**SDEP validation**

The logical integrity of SDEP pointers can be verified when you execute the analysis function. You can specify an implicit value for the SDEP_VALIDATION keyword on the GLOBAL command. This keyword value is used by all analysis functions within the job step. To override the GLOBAL command value, explicitly specify the SDEP_VALIDATION keyword on the ANALYZE command. Any of the following modes of pointer validation can be selected by using the SDEP_VALIDATION keyword:

82  *Fast Path Online Suite User Guide*
SDEP validation

- **SDEP_VALIDATION=FULL** constructs a complete cross-reference of every pointer and segment occurrence. The analysis function checks every pointer to ensure that a segment occurrence appears at the referenced location. The complete chain of sequential dependents that are associated with the root segment are identified. The analysis function verifies that each segment occurrence is referenced by one (and only one) pointer.

- **SDEP_VALIDATION=QUICK** (the default) generates a checksum for all sequential dependents within the area. The checksum consists of the sum of the RBAs for each segment occurrence minus the RBAs of all SDEP pointers. Because there should be exactly one pointer to each segment, a non-zero value indicates a pointer problem.

- **SDEP_VALIDATION=OFF** bypasses logical verification of sequential dependent pointers. The product still reads the SDEP control intervals to perform physical validation and to gather statistics.

- **SDEP_VALIDATION=NONE** bypasses all processing of the SDEP control intervals. The product does not perform physical validation and does not gather statistics.

### Active SDEP boundaries

The boundaries of the active sequential dependent portion of the area are defined by the logical begin (LB) and logical end (LE) values that are stored within the DMAC. The interpretation of these values depends on the type of analysis that is being executed.

During online analysis, other applications might be updating the database. When analysis starts, the LB value is obtained from the in-core copy of the DMAC that is used by the IMS control region. Because new SDEP segments can be added during analysis, Fast Path Online Analyzer/EP ignores the LE in the DMAC and computes its own value for the LE as the LB plus one full cycle. This value is the largest that the LE could possibly become during the analysis. As the root segments are scanned, the 8-byte SDEP pointers are evaluated against this range. If the SDEP pointer is less than LB, it is treated as a null pointer (ignored). If the SDEP pointer is greater than the computed LE, it is reported as an error. Fast Path Online Analyzer/EP obtains the updated values of the LB and LE from the in-core copy of the DMAC used by the IMS control region just before it scans the SDEP blocks.

During concurrent analysis, other applications might be updating the database. Because of this discrepancy window, which is characteristic of concurrent mode, the LB and LE values stored in the DMAC might be completely out of date by the time of the analysis. The analysis function scans SDEP segments only if they fall within the recorded boundaries. All 8-byte SDEP pointers outside of these boundaries are treated as null pointers (ignored). SDEP segments that are added after the DMAC has been obtained will cause SDEP pointer problems and/or orphaned segments to be reported.
Correcting pointer errors

**NOTE**

SDEP_VALIDATION=QUICK should *not* be used in concurrent mode.

---

Orphaned sequential dependents

If a root segment is deleted, its sequential dependent segments become “orphans.” The segments still exist within the active sequential dependent portion of the area, but they are not referenced by a valid SDEP pointer. The SDEP validation modes handle orphaned sequential dependents as follows:

- **SDEP_VALIDATION=None** and SDEP_VALIDATION=OFF do not detect orphaned segments and will not report them as errors.

- **SDEP_VALIDATION=QUICK** reports a pointer error if an orphaned segment exists. This mode of pointer validation cannot distinguish between orphaned segments and other problems, and it is not recommended if orphaned segments are known to exist in the area.

- **SDEP_VALIDATION=FULL** identifies the individual segments as orphans.

You can use the ORPHANED_SDEP_MSG keyword to control the severity that is associated with this condition and the generation of messages.

**NOTE**

If you are running IMS version 6.1 or later, any SDEP with a time stamp less than the logical begin time stamp is treated as deleted. SDEP_VALIDATION=QUICK can report a pointer problem after SDEP scan/delete has been run. This condition occurs if a root segment is deleted and that root segment was previously the owner of one of the logically deleted SDEP segments.

---

Correcting pointer errors

If pointer errors occur during the analysis process, you can use the optional CORRECTIONS_FILECTL subcommand with the ANALYZE command to correct pointer errors. If the CORRECTIONS_FILECTL subcommand is specified, pointer corrections control cards are created and then written to a pointer corrections output data set that can be used to zap pointer errors to zero. If the CORRECTIONS_FILECTL subcommand is not specified, no pointer corrections control cards are created, and invalid pointers are not removed.
Restrictions for correcting pointer errors

Pointer corrections control cards can only be generated for direct pointers or sequential pointers. As a result, POINTER_VALIDATION=FULL or SDEP_VALIDATION=FULL must also be specified with the ANALYZE command and the CORRECTIONS_FILECTL subcommand to correct pointer errors.

The following restrictions apply to correcting errors for direct pointers or sequential pointers:

- To generate pointer corrections records for direct pointers, POINTER_VALIDATION =FULL must be specified with the Analyze function.

- To generate pointer corrections records for sequential pointers, SDEP_VALIDATION=FULL must be specified with the Analyze function.

**NOTE**

If POINTER_VALIDATION=FULL is specified, SDEP_VALIDATION defaults to FULL.

- Pointer corrections control cards can only be generated for the following errors:

  BMC121190E  dbdname, areaname, rba: Segment=segname1 segtype to segment=segname2 - pointer=value out of bounds

  BMC121191E  dbdname, areaname, rba: Segment=segname1 segtype to segment=segname2 - pointer=(cycle:rba) out of logical bounds

  BMC121192E  dbdname, areaname, rba: Segment=segname1 segtype to segment=segname2 - SDEP pointer=(cycle:rba) out of physical bounds

  BMC121193E  dbdname, areaname, rba: Segment=segname1 ptrtype to segment=segname2 - segment points to itself

  BMC121194E  dbdname, areaname, rba: Segment=segname1 ptrtype to segment=segname2 PCF pointer=00000000, ptrtype pointer=value

  BMC121200E  dbdname, areaname, rba: Segment=segname1 found - segment=segname2 expected Segment=segname3 (at rba2) contains a ptrtype pointer to segment=segname2 (at rba1)
Allocating the pointer error corrections data set

The pointer error corrections output data set can be allocated by using JCL DD statements or by using dynamic allocation. When the CORRECTIONS_FILECTL subcommand is specified, along with POINTER_VALIDATION=FULL or SDEP_VALIDATION=FULL on the ANAYZE command, a pointer error corrections output data set is always written. However, if no pointer errors are encountered, the pointer error corrections output data set will be empty. By default, only one output pointer error corrections data set is produced for each input area; you cannot combine multiple areas into a single data set.

Using CORRECTIONS_FILECTL with standard JCL

Use the DDNAME keyword (not the DSNAME keyword) to indicate that the pointer error corrections data set is to be allocated by using JCL DD statements.

**NOTE**

When using JCL to allocate the pointer error corrections data set, ensure that only one area is written to any DD statement. If more than one area is written to the same DD statement, the pointer error corrections data set of the second area will overwrite the pointer error corrections data set of the first area.

The example in Figure 46 uses JCL to allocate a single pointer error corrections data set for an area.

**Figure 46** JCL to allocate a pointer error corrections data set by using the DDNAME keyword (part 1 of 2)

```plaintext
//PFP EXEC PGM=DFSRRC00,REGION=0M,
//       PARM=(IFP,dbdname, DBF#FPU0)
//STEPLIB DD DISP=SHR, DSN=BMC.PFP.LOAD
//       DD DISP=SHR, DSN=IMS.RESLIB
//ddname DD UNIT=TAPE, DISP=(NEW,CATLG),
//       DSN=your.dataset.name
//PFPSYIN DD *
```
Using CORRECTIONS_FILECTL with dynamic allocation

Use the DSNAME keyword (not the DDNAME keyword) to indicate that the pointer error corrections data is to be dynamically allocated and to specify the name of the data set where the error corrections control cards are to be written. Use the DSNAME keyword to supply a mask to construct the data set name. Substitution variables within the mask can be used to generate unique names for each output data set.

Additional related keywords, such as UNIT, DISP, and DATACLAS, can be used to control the allocation and disposition of the data set, similar to the corresponding JCL keywords.

The example in Figure 47 shows how to generate a pointer error corrections data set by using dynamic allocation.

**Figure 47  Dynamic allocation of pointer error corrections data set by using DSNAME keyword**

```plaintext
ANALYZE DBDD=dbdname, AREAA=areaname,
POINTER_VALIDATION=FULL,
CORRECTIONS_FILECTL DDNAME=ddname
/*
```

Processing the generated output

The output that is written to the pointer error corrections data set can be used to correct pointer errors. The PROCESS_AREA command and the PERFORM subcommand are automatically generated by the analysis process (and placed in the pointer error corrections data set) for each area that contains one or more pointer errors. In addition, a set of VER, REP, and COMMIT parameters is also generated for each pointer error.

The example in Figure 48 shows how to process the output that is written to the pointer error corrections data set.

**Figure 46  JCL to allocate a pointer error corrections data set by using the DDNAME keyword (part 2 of 2)**

```plaintext
ANALYZE DBD=dbdname, AREA=areaname,
POINTER_VALIDATION=FULL,
CORRECTIONS_FILECTL DDNAME=ddname
```
Generating analysis reports

The REPORT subcommand controls the generation of analysis function reports. The REPORT subcommand lets you specify characteristics of output reports. This subcommand works whether analysis is invoked directly using the ANALYZE command or by another BMC utility that uses the appropriate command or keyword.

The REPORT subcommand defines a report set which consists of analysis selections, UOW range selection, report formatting options, and routing options. Multiple REPORT subcommands can be used to request the generation of multiple report sets. Each report set is independent of every other report set.

You can specify reporting controls by placing your REPORT subcommands within the GLOBAL command. These reporting specifications are used by all analysis functions within the job step. To override these reporting specifications, place one or more REPORT subcommands within the ANALYZE command.

Report selection

The analysis function can produce any or all of 11 reports that fall into three categories. These reports contain detailed data to help you understand every aspect of database performance, space usage, segment characteristics, and record characteristics. As shown in Table 6, a keyword is available to request each report individually.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Report title</th>
<th>Report keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>space usage</td>
<td>Free Space Analysis</td>
<td>FREESPACE_ANALYSIS</td>
</tr>
<tr>
<td></td>
<td>IOVF Space Analysis</td>
<td>IOVF_SPACE_ANALYSIS</td>
</tr>
<tr>
<td></td>
<td>UOW Detailed Analysis</td>
<td>UOW_DETAILED_ANALYSIS</td>
</tr>
<tr>
<td>segment</td>
<td>Pointer Analysis</td>
<td>POINTER_ANALYSIS</td>
</tr>
<tr>
<td></td>
<td>Segment I/O Analysis</td>
<td>SEGMENT_IO_ANALYSIS</td>
</tr>
</tbody>
</table>

Table 6  Fast Path Online Analyzer/EP characteristics, report titles, and keywords (part 1 of 2)
Two of these reports use additional customization keywords:

- The SYNONYM_CHAIN_INCREMENT keyword specifies the reporting interval for the Synonym Chain Analysis Report. This keyword is specified on the REPORT subcommand.

- The RECORD_LENGTH_INCREMENT keyword specifies the record length interval for the Record Length Analysis Report. This keyword is specified on the REPORT subcommand.

- The LARGEST_DATABASE_RECORDS keyword specifies the number of largest database records to be reported on the Record Length Analysis Report. This keyword can be specified under any primary command that enables you to request analysis with the POINTER_VALIDATION keyword.

**NOTE**
You must specify POINTER_VALIDATION=FULL on the primary command to generate the Record Length Analysis Report. The LARGEST_DATABASE_RECORDS keyword implements the tracking of these records only; it does not generate the report. For more information about the function of the LARGEST_DATABASE_RECORDS keyword, see page 92.

For summary descriptions of these reports, see “Analysis report descriptions” on page 101. For more detailed descriptions of these reports, see the Fast Path/EP Series Reference Manual. For examples of the reports, see the PFPANLYZ member in the REPORTS data set.

### Requesting all reports

You can use the REPORT_DEFAULT keyword to specify whether all reports are produced by default. If you specify REPORT_DEFAULT=YES, all reports are produced unless you suppress individual reports by using the associated keywords.
Excluding selected reports

The sample command set in Figure 49 specifies that analysis will issue all reports (except for the UOW Detailed Analysis Report).

Figure 49  Sample JCL specifying special reporting to exclude one report

```
ANALYZE DBD=dbdname,AREA=areaname
REPORT
  REPORT_DEFAULT=YES,
  UOW_DETAILED_ANALYSIS=NO
```

Producing selected reports

If you specify REPORT_DEFAULT=NO, no reports are produced unless you select individual reports by using the associated keywords. The sample command set in Figure 50 specifies that only the Free Space Analysis Report be produced.

Figure 50  Sample JCL specifying special reporting to include one report

```
ANALYZE DBD=dbdname,AREA=areaname
REPORT
  REPORT_DEFAULT=NO,
  FREESPACE_ANALYSIS=YES
```

Unless POINTER_VALIDATION= FULL is specified, the information that is required for generation of certain reports is not collected. The following reports are generated only in FULL validation mode:

- Segment I/O Analysis
- Record Length Analysis
- Record Placement Analysis

The information that is required for the Synonym Chain Analysis Report is collected if POINTER_VALIDATION=FULL or RAP_VALIDATION=XREF is specified. If neither of these keyword values is specified, the report is not generated.

UOW range report specification

Keywords that specify the UOW range on which to report include the following:

- STARTUOW specifies the UOW on which to start reporting.
- STOPUOW specifies the UOW on which to stop reporting.
You can use the STARTUOW and STOPUOW keywords to specify a range of UOWs to be included in the report set.

To limit the range of UOWs to be analyzed, use the command set in Figure 51.

**Figure 51  Sample JCL limiting the range of UOWs to be analyzed**

```
ANALYZE DBD=dbdname,AREA=areaname
REPORT
  STARTUOW=startuow,
  STOPUOW=stopuow
```

The report set includes database dbdname, area areaname, starting with UOW startuow and ending with UOW stopuow. These keywords have no effect on the verification processing; the complete area is always verified. You can use the keywords to limit reporting to certain UOWs. Some reports are always area-wide, regardless of UOW range selection.

If the HISTORY_DDNAME keyword has been specified, a separate history record is produced for each UOW range that is specified.

**Report formatting**

You can use the REPORT_HEADING keyword to specify a value to be generated in the page heading of each output page in the report set.

You can use the REPORT_LINE_COUNT keyword to specify the maximum number of printed lines to be placed on each output page within the report set.

**Report routing**

The destination of reports can be designated by using the REPORT_DDNAME keyword. You can use the JCL parameters on the referenced DD statements to place reports on tape, disk, or other storage medium.

You can generate multiple copies of a report by listing more than one ddname. You can also route more than one report set to the same destination by using the same ddname on multiple REPORT subcommands.

**WARNING**

Because the DD statement is opened and closed for each report set, you must use DISP=MOD in these cases.
Specifying the number of largest database records

The example in Figure 52 uses two REPORT subcommands. The first set of reports is written to a tape data set; the second REPORT subcommand requests the Free Space Analysis Report to be routed to the default destination.

**Figure 52** Sample JCL for special use of the REPORT subcommand

```
//PFP EXEC PGM=DFSRRC00,REGION=0M,
//      PARM=(IFP,dbname,DFU#FPU0)
//STEPLIB  DD DISP=SHR,DSN=BMC.PFP.LOAD
//         DD DISP=SHR,DSN=IMS.RESLIB
//REPORTS DD UNIT=TAPE,DSN=your.dataset.name,
//         DISP=(NEW,CATLG)
//PFPSYSIN DD *
ANALYZE DBD=dbname,IAREA=areaname
   REPORT
      REPORT_DDNAME=REPORTS
   REPORT
      REPORT_DEFAULT=NO,
      FREESPACE_ANALYSIS=YES
/*
```

Specifying the number of largest database records

Fast Path Online Analyzer/EP incorporates a method that allows you to specify the number of the largest database records that are tracked by the analysis process. By default, Fast Path Analyzer/EP retains information on the 10 largest database records when you specify full pointer analysis by using the POINTER_VALIDATION keyword.

You can request a different number of largest database records by specifying a desired value on the LARGEST_DATABASE_RECORDS keyword. This keyword can be specified under any primary command that enables you to request analysis with the POINTER_VALIDATION keyword. When you specify the LARGEST_DATABASE_RECORDS keyword, the process will reserve adequate storage to track the number of largest database records that you request.

If you have activated the Fast Path Analyzer/EP statistics repository facility, the information that is gathered by the analysis process is stored in the repository for future retrieval. The information on the largest database records can also be obtained at a later time by requesting the Record Length Analysis Report from the repository using any of the available methods. For more information, see the *Fast Path Offline Suite User Guide*.

The example shown in Figure 53 will track information on the 30 largest database records contained in the specified area.
The example shown in Figure 54 will also track information on the 30 largest database records contained in the area. The REPORT subcommand is specified on the command set to generate the Record Length Analysis Report. This report will make the information available on these largest database records.

Figure 54   Requesting the record length analysis report simultaneously with specification of number of largest database records

```
ANALYZE DBD=dbdbname, IAREA=areaname,
    POINTER_VALIDATION=FULL,
    LARGESTDATABASE_RECORDS=30
    REPORT REPORT_DEFAULT=NO, RECORD_LENGTH_ANALYSIS=YES
```

### Detecting exception conditions

To facilitate the monitoring of particular database characteristics during analysis, the analysis function provides the THRESHOLD subcommand and its associated keywords to indicate database threshold settings. Using these keywords, you can specify that warning messages be sent to the processing log when certain database conditions are detected. Threshold tests are performed only when you specify the corresponding threshold keywords. Threshold exception conditions that are detected are shown also on the Area Summary Report. The threshold keywords are categorized into the general groups types that are listed in Table 7.

#### Table 7   Threshold types and keywords (part 1 of 2)

<table>
<thead>
<tr>
<th>Threshold Type</th>
<th>Threshold Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>free space</td>
<td>DOVF_FREESPACEMAX_PERCENT</td>
</tr>
<tr>
<td></td>
<td>FREESPACEMAX_DOVF</td>
</tr>
<tr>
<td></td>
<td>FREESPACEMAX_RAA_DOVF</td>
</tr>
<tr>
<td></td>
<td>FREESPACEMAX_RAA_IOVF</td>
</tr>
<tr>
<td></td>
<td>IOVF_FREESPACEMAX_PERCENT</td>
</tr>
<tr>
<td></td>
<td>IOVF_USED_PERCENT</td>
</tr>
<tr>
<td></td>
<td>RAA_FREESPACEMAX_PERCENT</td>
</tr>
<tr>
<td></td>
<td>RAP_OVERFLOW_PERCENT</td>
</tr>
<tr>
<td>I/O</td>
<td>RECORD_IO_AVERAGE</td>
</tr>
<tr>
<td></td>
<td>RECORD_IO_MAXIMUM</td>
</tr>
</tbody>
</table>
Detecting exception conditions

You can specify thresholds by placing the THRESHOLD subcommand within the GLOBAL command. These implicit threshold specifications are used by all analysis functions within the job step. To override the GLOBAL command, you can explicitly specify a THRESHOLD subcommand within the ANALYZE command. In the example shown in Figure 55, thresholds that are specified on the GLOBAL command are tested for all areas except area2 of database1. Specifying THRESHOLD without any keywords under the last ANALYZE command overrides the GLOBAL command, so no threshold checking is performed for area2.

### Table 7  Threshold types and keywords (part 2 of 2)

<table>
<thead>
<tr>
<th>Threshold Type</th>
<th>Threshold Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O (continued)</td>
<td>ROOT_IO_AVERAGE</td>
</tr>
<tr>
<td></td>
<td>ROOT_IO_MAXIMUM</td>
</tr>
<tr>
<td>RAP usage</td>
<td>SYNONYM_CHAIN_LENGTH</td>
</tr>
<tr>
<td></td>
<td>SYNONYM_CHAIN_MAXIMUM</td>
</tr>
<tr>
<td></td>
<td>SYNONYM_CHAIN_PERCENT</td>
</tr>
<tr>
<td>overflow characteristic</td>
<td>RECORD_IOVF_PERCENT</td>
</tr>
<tr>
<td></td>
<td>UOW_DOVF_PERCENT</td>
</tr>
<tr>
<td></td>
<td>UOW_IOVF_AVERAGE</td>
</tr>
<tr>
<td></td>
<td>UOW_IOVF_MAXIMUM</td>
</tr>
<tr>
<td></td>
<td>UOW_IOVF_PERCENT</td>
</tr>
</tbody>
</table>

**NOTE**

Record I/O thresholds and the RECORD_IOVF_PERCENT overflow threshold can be tested only if POINTER_VALIDATION=FULL has been requested. Root I/O thresholds and RAP usage thresholds can be tested if POINTER_VALIDATION=FULL or RAP_VALIDATION=XREF has been specified.

You can specify thresholds by placing the THRESHOLD subcommand within the GLOBAL command. These implicit threshold specifications are used by all analysis functions within the job step. To override the GLOBAL command, you can explicitly specify a THRESHOLD subcommand within the ANALYZE command. In the example shown in Figure 55, thresholds that are specified on the GLOBAL command are tested for all areas except area2 of database1. Specifying THRESHOLD without any keywords under the last ANALYZE command overrides the GLOBAL command, so no threshold checking is performed for area2.

### Figure 55  Using the THRESHOLD subcommand with the GLOBAL command (part 1 of 2)

```plaintext
//PFP EXEC PGM=DFSRC00,REGION=0M,
// PARM=(IFP, database1, DBF#FPU0)
//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
//         DD DISP=SHR,DSN=IMS.RESLIB
//PFPSYSIN DD *
GLOBAL
  REPORT
  RECORD_PROFILE_ANALYSIS=NO
  THRESHOLD
  FREESPACE_RAA_DOVF=(25,25),
  FREESPACE_RAA_IOVF=(50,50)
ANALYZE DBD=database1
ANALYZE DBD=database2,IAREA=(area1,area3,area4)
ANALYZE DBD=database2,IAREA=area2
```
Generating input to the Area Change Modeling Utility

To generate input to the Area Change Modeling Utility, add the following statements to the JCL you use for the analysis function:

- Add a DD statements identifying the segment information data set to be created.
- Specify the MODEL_DDNAME keyword on the ANALYZE command to identify the names of the DD statements in the JCL.

You can generate multiple copies of a report by listing more than one ddname.

The segment information data set is used as input to the Area Change Modeling Utility. Sample JCL is shown in Figure 56. For more information about the Area Change Modeling Utility, see the Fast Path/EP Series Reference Manual.

### Figure 56  Sample JCL for generating input to the Area Change Modeling Utility

```plaintext
//PFP EXEC PGM=DFSRC00,REGION=OM,
// PARM=(IFP,dbname,DFB#FPUO)
//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
//      DD DISP=SHR,DSN=IMS.RESLIB
//ddname1 DD DISP=OLD,DSN=your.dataset.name
//PFPSYSIN DD *
ANALYZE DBD=dbname,
      MODEL_DDNAME=ddname1
/*
```

Suppressing repetitious messages

For each anomaly that is encountered in an area, the analysis function generates a warning or error message. When the number of places that a particular condition exists is large, a large number of messages is produced.
Using a history file

You can reduce the number of repetitious messages by using the \texttt{MESSAGE\_SUPPRESSION} keyword. Use this keyword to specify the maximum number of times that a warning or error message is to be produced. In the example in Figure 57, the \texttt{MESSAGE\_SUPPRESSION} keyword specifies that a warning message be issued no more than 10 times and that an error message be produced no more than 15 times. When messages are suppressed in this way, Fast Path Online Analyzer/EP produces a summary of the number of times that each message has been suppressed.

Figure 57 Using the \texttt{MESSAGE\_SUPPRESSION} keyword

\begin{verbatim}
ANALYZE DBD=dbdname, IAREA=areaname,
MESSAGE\_SUPPRESSION=(10,15)
\end{verbatim}

To suppress the generation of orphaned SDEP error messages, you can specify \texttt{ORPHANED\_SDEP\_MSG=NONE} as shown in Figure 58.

Figure 58 Using the \texttt{ORPHANED\_SDEP\_MSG} keyword

\begin{verbatim}
ANALYZE DBD=dbdname,
ORPHANED\_SDEP\_MSG=NONE
\end{verbatim}

Using a history file

Fast Path Online Analyzer/EP provides a facility to generate a file containing historical summary records. BMC recommends the use of the Repository Facility for storing historical data. This data can be collected based on the analysis of multiple areas, and can be generated in a single report. For more information, see the \textit{Fast Path Offline Suite User Guide}.

When an area is analyzed, a statistics record is \textit{appended} to a user-specified file. The file can be used as input to a user-supplied reporting facility.

Use the \texttt{HISTORY\_DDNAME} keyword on the \texttt{ANALYZE} command to specify the ddnames of DD statements to which the records are to be written. You can generate multiple copies of a report by listing more than one ddname.

\textbf{WARNING}

To prevent destructive concurrent access to the data set, the DD statement that is used for the output data should specify \texttt{DISP=OLD} as shown in Figure 59.

Use the sample JCL command set in Figure 59 to write statistics to the history file.
UOW range specification

An additional record is written to the history file for each UOW range that is requested on any REPORT subcommand for the area being processed. The record contains statistics that are specific to the range of UOWs that are requested.

The sample JCL command set in Figure 60 will write two records to the history file. One record will contain statistics for the entire area; the other record will contain statistics only for the UOWs that are specified.

Figure 59 Using the HISTORY_DDNAME keyword

```plaintext
//PFP EXEC PGM=DFSRC00,REGION=OM,
// PARM=(IFP,dbname,DFB#FPU0)
//STEPLIB DD DSN=BMC.PFP LOAD,DISP=SHR
// DD DSN=IMS.RESLIB,DISP=SHR
//ddname1 DD DSN=your.dataset.name,DISP=OLD
//PFPSYSIN DD *
  ANALYZE DBD=dbname,IAREA=areaname,
  HISTORY_DDNAME=ddname1
/*
```

For the complete history file record layout, see the Fast Path/EP Series Reference Manual.

Figure 60 Limiting range of UOWs in record written to history file

```plaintext
//PFP EXEC PGM=DFSRC00,REGION=OM,
// PARM=(IFP,dbname,DFB#FPU0)
//STEPLIB DD DSN=BMC.PFP LOAD,DISP=SHR
// DD DSN=IMS.RESLIB,DISP=SHR
//ddname1 DD DSN=your.dataset.name,DISP=OLD
//PFPSYSIN DD *
  ANALYZE DBD=dbname,IAREA=areaname,
  HISTORY_DDNAME=ddname1
  REPORT STARTUOW=5,STOPUOW=10
/*
```
Enhancing performance

Caching is not used during an online analysis.

Using the input caching feature with concurrent analysis can speed the process in many circumstances. Using the ICACHE keyword to specify IOVF input caching can improve performance by eliminating the random I/O to the IOVF control intervals. Do not specify input caching of SDEP CIs. SDEP CIs are processed sequentially; caching provides no benefit and can actually slow the process.

Whether input caching is useful and significant to the concurrent analysis function depends on the internal operating mode of the function.

The analysis function has two internal modes of operation: sequential mode and direct mode. Fast Path Online Analyzer/EP decides internally which mode is required, or which is more efficient, and executes accordingly. If Fast Path Analyzer/EP performs the analysis function in sequential mode, any caching that is specified by using the ICACHE keyword is ignored.

Producing an image copy during analysis

If your site has a license for Fast Path Online Image Copy/EP, you can request that one or more output image copy data sets be produced during online analysis. Specify the IC subcommand for each image copy to be produced. The example in Figure 61 requests the creation of a single output image copy data set for the area being analyzed. For details about the IC subcommand, see Chapter 6, “Making an online image copy.”

Figure 61  Requesting an image copy during analysis

```
ANALYZE DBD=dbdname,IAREA=areaname
   IC DSNNAME='dataset-name-mask',UNIT=TAPE,DISP=(NEW,CATLG)
```

Image copy output is not supported during analysis in concurrent mode or when the input to the analysis function is an image copy data set.
Fast Path Online Analyzer/EP will perform pointer validation *automatically* (by default) when you make an online image copy by using the Fast Path Online Image Copy/EP IMAGECOPY command.

Automatic pointer validation provides confirmation of the area’s pointer integrity, while providing statistics that show space usage.

---

**NOTE**

You can override automatic quick pointer validation and request more extensive validation by specifying POINTER_VALIDATION=FULL on the IMAGECOPY or GLOBAL command. Direct pointers will be validated by using the cross-reference technique. Although not recommended, you can also specify any other valid value for the POINTER_VALIDATION keyword.

If an SDEP segment is defined for the database, SDEP pointers are also validated by default by using the same technique as specified for direct pointers. You can also specify any valid value for the SDEP_VALIDATION keyword to control how SDEP pointers are to be validated.

You can also specify additional functions that are associated with the online analysis function which is provided by Fast Path Online Analyzer/EP. Use the RAP_VALIDATION keyword to control how RAPs are processed. Use the REPORT and THRESHOLD subcommands to control the generation of analysis reports and exception testing.

The example in Figure 62 requests the creation of a single output image copy data set for each area within the database. By omission of a value for POINTER_VALIDATION, quick analysis is also performed for each area. SDEP pointers (if any exist) are validated by using the same technique. For more information, see “Pointer validation” on page 80. Only the Free Space Analysis Report is produced.

---

**Figure 62** Specifying the analysis function with the IMAGECOPY command

```plaintext
IMAGECOPY DBD=ddname
  IC DSNAME='dataset-name-mask',
  DISP=(NEW,CATLG),UNIT=TAPE
  REPORT DEFAULT=NO,FREESPACE_ANALYSIS=YES
```
Requesting analysis during reorganization

Fast Path Online Analyzer/EP will perform pointer validation automatically (by default) when you execute an online reorganization by using the Fast Path Online Reorg/EP REORGANIZE command.

This automatic process applies the default value QUICK for the POINTER_VALIDATION keyword. The analysis function performs a checksum validation of pointers for each segment type within each UOW of the database that is specified on the REORGANIZE command.

Automatic pointer validation provides assurance of the area’s pointer integrity, while providing statistics that show how the reorganization process affected space usage.

**NOTE**

You can override automatic quick pointer validation and request more detailed validation by specifying POINTER_VALIDATION=FULL on the REORGANIZE or GLOBAL command. Direct pointers are validated by using the cross-reference technique. You can also specify any other valid value for the POINTER_VALIDATION keyword (such as POINTER_VALIDATION=NONE to completely disable the analysis process).

You can manually specify another valid value for the SDEP_VALIDATION keyword to control how SDEP pointers are to be validated.

You can also specify additional functions that are associated with the analysis function:

- Use the RAP_VALIDATION keyword to control how RAPs are processed.

- Use the LARGEST_DATABASE_RECORDS keyword to specify the number of largest database records to be tracked by the analysis process.

- Use the REPORT and THRESHOLD subcommands to control the generation of analysis reports and exception testing.

Because no values are specified for the POINTER_VALIDATION and SDEP_VALIDATION keywords, the example in Figure 63 requests that pointer validation be performed during reorganization of the area by using the QUICK (checksum) technique and that no validation of SDEP pointers is performed. Full validation for RAP pointers is requested explicitly. The example produces only the Free Space Analysis report and does not check threshold conditions.

**Figure 63** Control statements for requesting analysis during reorganization (part 1 of 2)

```
REORGANIZE  DBD=dbdname.
  RAP_VALIDATION=XREF.
```
Analysis report descriptions

Reports can be produced during online analysis processes. This section provides brief descriptions of these reports. For detailed descriptions of each report, see the Fast Path/EP Series Reference Manual.

Area Summary Report

The Area Summary Report provides basic information about each area that is analyzed. It is generated automatically with each report set and precedes all other reports. Threshold exception conditions are shown on the report.

Free Space Analysis Report

The Free Space Analysis Report provides the following information:

- summary and detailed information about the total and usable amount of free space in each section of the database (RAA base, DOVF, and IOVF)
- statistics about the number and size of the free space elements in each section of the database
- overflow usage analysis (DOVF and IOVF)

For an example of the report, refer to the PFPANLYZ member in the REPORTS data set.
The Free Space Analysis Report provides valuable information that can be used to determine the following items:

- when to reorganize or expand an area
- optimal values for UOW and ROOT parameters
- DOVF and IOVF usage

Free space calculations have the following characteristics:

- The entire area is included in the report. When an optional UOW range report is requested, the report also contains a summary of the free space within the selected UOWs.

- The total usable space in a CI is the block size minus 21 (overhead bytes).

- The length of the free space element (FSE) is included in usable free space only if it is large enough to hold the smallest maximum segment defined in the area.

- Scraps (free space less than 4 bytes) are included in free space calculations, but they are not included in FSE calculations.

**IOVF Space Analysis Report**

The IOVF Space Analysis Report provides IOVF block usage information that is obtained from each IOVF space map block to show how IOVF is being allocated. All space map blocks in the area are included in the report. The range scale indicates the amount of full CIs in the IOVF section.

**UOW Detailed Analysis Report**

The UOW Detailed Analysis Report provides detailed information about each UOW. You can use the report to perform the following tasks:

- identify which UOWs require reorganization
- manage space usage when the randomizer has been modified to group related data into contiguous UOWs
**Pointer Analysis Report**

For each segment type within a database, the Pointer Analysis Report provides detailed information for each of the following prefix pointer types:

- physical twin forward (PTF)
- physical child first (PCF) and physical child last (PCL)
- subset (SSPTR) and sequential dependent (SDEP) pointers

Additionally, each root anchor point (RAP) within the database is treated as a special segment type so that RAP chain pointers are included in this report. The following statistics are provided for each pointer type:

- count and percentage of null pointers
- count and percentage of pointers that point into the same CI
- count and percentage of pointers that point into a different CI

**Segment I/O Analysis Report**

The Segment I/O Analysis Report provides valuable information for database performance tuning and for evaluation of I/O requirements for specific user transactions. Analysis is used also to determine the optimal reorganization point and to validate the effects of UOW and ROOT value parameter changes.

The Segment I/O Analysis Report can be generated only when POINTER_VALIDATION=FULL is specified.

The Segment I/O Analysis Report provides the following statistics for a typical database record:

- direct dependent segments in each part of an area (SDEPs segments are not included)
- physical I/O operations required to retrieve dependent segments
- physical I/O required to retrieve an average database record
- physical I/O required to retrieve an average root segment
Segment Length Analysis Report

The Segment Length Analysis Report shows the data lengths of segments in 21 reporting intervals (or less). In each interval, statistics are provided on the number of segments and their lengths. This report lets you see the level of compression for compressed segments.

Segment Placement Analysis Report

The Segment Placement Analysis Report provides an overview of where segments reside in the database. This report can provide information when to use the load control function in Fast Path Reorg/EP.

NOTE
SDEP segments are not included in this report.

Record Length Analysis Report

The Record Length Analysis Report shows the variability of database record lengths in each area. By default, it also provides information about the 10 largest database records in each area that are tracked by the analysis process.

You can override the default number of largest database records to be tracked by the analysis process. To request a number of largest database records that is different from the default, specify the desired value on the LARGEST_DATABASE_RECORDS keyword. This keyword can be specified under any primary command that enables you to request analysis with the POINTER_VALIDATION keyword.

The Record Length Analysis Report can be generated only when POINTER_VALIDATION=FULL is specified on the primary command.

The Record Length Analysis Report shows database record lengths in 21 reporting intervals. Each reporting interval provides statistics about the number and percentage of database records and the average, maximum, and minimum record lengths.
Record Placement Analysis Report

The Record Placement Analysis Report shows the number of database records in each of seven placement categories and the statistics about the number of DOVF and IOVF blocks used by a record. The report indicates where the data will be placed and overall segment placement distribution. These statistics are helpful in determining sources of high I/O.

The Record Placement Analysis Report can be generated only when POINTER_VALIDATION=FULL is specified.

The Record Placement Analysis Report provides information to help you determine

- when to reorganize a DEDB area
- when to expand control interval (CI) size
- the optimal value for the UOW parameter in the DBD

Record Profile Analysis Report

The Record Profile Analysis Report shows statistics about the number and length of database records, dependent segment frequencies, and length statistics. SDEP segments are not included in this report.

The Record Profile Analysis Report provides valuable information about the characteristics of the database records. In addition, the information is used to determine

- when to reorganize an area
- optimal values for the ROOT and UOW parameters in the DBD

Synonym Chain Analysis Report

The Synonym Chain Analysis Report gives detailed information about the frequency and length of synonym chains. The report provides statistics about root segment placement and physical I/O requirements.

The Synonym Chain Analysis Report can be generated only when POINTER_VALIDATION=FULL or RAP_VALIDATION=XREF is specified.
Retaining analysis statistics in the repository

The Synonym Chain Analysis Report provides valuable information to determine

- optimal values for ROOT and UOW parameters
- performance tuning for an area
- effectiveness of a randomizing module (In theory, root segments should be distributed evenly across all RAPs.)

Retaining analysis statistics in the repository

Fast Path Online Analyzer/EP incorporates an optional statistics repository facility. The statistics repository creates and maintains a set of statistics that are collected during analysis processes. This information can be used for historical reporting and trend analysis. The purpose of the repository facility is to provide a mechanism for the storage and retrieval of statistics that are collected by execution of the analysis process. If the repository facility is not activated, no statistical information is retained after an analysis of a DEDB or area. The repository facility is available to online and offline analysis processes.

The repository facility lets you

- store the statistics that are obtained by the analysis function
- retrieve the statistics for reporting purposes
- use the statistics to perform trend analysis and forecasting
- store, view, or delete customizations to the severity level or issuance threshold (the number of times it is issued before it is suppressed) for selected product messages

What the repository contains

The information stored in the repository is the same statistical information that Fast Path Online Analyzer/EP uses to generate reports through the REPORT subcommand. The repository consists of a catalog data set and any number of statistics data sets. The catalog data set contains a record of all statistics data sets and a set of allocation rules that are used in the creation of new statistics data sets. If the repository facility is active when you execute an online analysis, it creates a new statistics data set according to the allocation rules and adds a new entry into the repository catalog.
Using the repository

For complete information on creating, managing, and maintaining the repository for statistics storage, retrieval, and usage, see the Statistics Repository Facility chapter in the Fast Path Offline Suite User Guide. For information about using the repository for message customization, see Appendix B, “Dynamic message modification.”

Sample ANALYZE command scenarios

For scenarios that show how to use the online ANALYZE command with key related keywords and subcommands, see Appendix C, “Sample online command scenarios.”
Extracting data from an online DEDB

This chapter provides information on the capabilities, benefits, and use of the online DEDB data extract function that is provided by Fast Path Online Analyzer/EP. This function lets you extract full or partial segment data from DEDBs while they are online to IMS.

This chapter discusses how to specify the appropriate command syntax for all available extract functions. It also looks at the extract process from the user perspective as a series of chronological steps and explains the various results that can be obtained, depending on how you define the process in the job control language.

This chapter discusses the following topics:

- Online data extract function overview
  - Format flexibility
  - Benefits
- How online extract works
  - Control region requirement
  - Online data extract function inputs and outputs
  - Process overview
- EXTRACT command keywords and subcommands
- Selecting the extract file format
- Selecting the database and areas
- Defining the extract output data set
- Selecting database segments for input and output
- Including and excluding segments for input
- Narrowing the selection
- Outputting segment records
- Samples of INCLUDE, EXCLUDE, and OUTPUT subcommands
- Controlling segment expansion
- Controlling content of segment records
Online data extract function overview

The Fast Path Online Analyzer/EP online data extract function provides an excellent alternative to user-written batch message processing programs (BMPs) for extracting segment data (full or partial segments) from DEDBs while they are online to IMS. The online data extract function offers flexible, generalized segment selection and data extraction facilities. Selected segments and/or data items can be written to a sequential data set for subsequent processing by a user-written program.

Because Fast Path Online Analyzer/EP incorporates all batch functionality that is provided by the Fast Path Analyzer/EP product, you can also perform an offline data extract by using Fast Path Online Analyzer/EP. For information about performing offline data extracts, see the Fast Path Offline Suite User Guide.

Format flexibility

The EXTRACT command is versatile and powerful. While the database is online, you can extract as much or as little data as you want from a DEDB. The format and content of the outcome (the extract output file) can be controlled by specifying a user-customized combination of command syntax.

Input flexibility

The online data extract function lets you select the database and areas within the database to be used as input to the extract. The online data extract function also lets you include or exclude certain segments for processing.

Output flexibility

Extracted data is written to an output file by using one of three user-selectable output formats. The format that you select dictates the extent to which you can exercise control over the database-level and area-level header and trailer records, and the content of the segment data records (including segment prefixes and suffixes).
Benefits

If the application reads the extract data does not have specific formatting requirements, or if those requirements have not yet been determined, the default format (EXTRACT) should be used. A module is provided with Fast Path Analyzer/EP to facilitate reading the extracted data from a user-written program. For instructions about how to use this module, see the Fast Path/EP Series Reference Manual. For a sample COBOL program to retrieve extracted data, see the member PFUGXDR in the Fast Path/EP sample library.

The data might be written in the same format (HDUNLOAD) as the Fast Path Reorg/EP unload function. This format should be used when the extracted data is used as input to the reload process.

The data might be written in a custom-tailored format (USER). This format should be used when an existing application program requires extracted data in a specific format.

Benefits

The online data extract function provides the following benefits:

- It increases productivity of application programmers by extracting data, using a control statement-driven, generalized, high-speed online utility.
- It does not affect database availability. The EXTRACT command accesses a DEDB with logical database record integrity while the database is being updated by IMS.
- It provides a convenient mechanism for creating test databases. Specified database records can be unloaded easily from a production database and reloaded to a test database.
- It eliminates IMS overhead because BMPs are not required.
- It provides full multiple area data set (MADS) usage.

How online extract works

The online data extract function runs in an IMS Fast Path (IFP) dependent region. Fast Path Online Analyzer/EP completes an online data extract that is based on how you combine the EXTRACT command with its associated keywords and subcommands. This command syntax defines the inputs to (and the output from) the extract process.
Control region requirement

When performing an online data extract, you must define the database to the IMS Control Region with ACCESS=UP.

Online data extract function inputs and outputs

As shown in Figure 64, the online data extract function extracts data from one or more area data sets of an online DEDB and writes the data to a sequential file.

Process overview

Because the EXTRACT command makes only one pass through the data in the area, it is important to understand the chronological process that it executes. Subcommands and keywords are executed by the online data extract function to control data content and record formatting at each step in the process.

Figure 65 shows how each step in the process is executed as specified by command language elements. However, because the availability of some options are dependent on other options, the subcommands and keywords are discussed in this chapter in the order that you should consider when becoming familiar with how to code a customized extract.
Figure 65  Online extract data control process

Table 8 lists the command language that is used for each of the steps in the online data extract process. Some keywords are directly related to the EXTRACT command; others require an interim subcommand. To facilitate your understanding and correct coding of JCL to implement an extract, BMC recommends that you consider these steps in the order presented in Table 8.
Selecting the extract file format

The general format of the extract output file can be controlled by specifying one of three values that are available for the EXTRACT_FORMAT keyword. As summarized in Table 9, each available choice enables a different set of more detailed options for tailoring extract output. Certain functions that affect record layout as well as the output data are available only for particular extract file formats.

### Table 8 EXTRACT command subcommands and keywords

<table>
<thead>
<tr>
<th>Function</th>
<th>Command or subcommand</th>
<th>Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>selecting the extract file format</td>
<td>EXTRACT</td>
<td>EXTRACT_FORMAT</td>
</tr>
<tr>
<td>selecting the database and areas</td>
<td>EXTRACT</td>
<td>DBD IAREA</td>
</tr>
<tr>
<td>defining the extract output data set</td>
<td>EXTRACT OFILECTL</td>
<td>OUTPUT_DSN_MASK</td>
</tr>
<tr>
<td>selecting database segments for input</td>
<td>INCLUDE EXCLUDE</td>
<td>SAMPLE_INTERVAL SAMPLE_LIMIT SEGMENT WHERE</td>
</tr>
<tr>
<td>selecting database segments for output</td>
<td>OUTPUT</td>
<td>SEGMENT WHERE</td>
</tr>
<tr>
<td>controlling segment expansion</td>
<td>EXTRACT</td>
<td>EXPAND</td>
</tr>
<tr>
<td>controlling the content of segment records</td>
<td>OUTPUT</td>
<td>SEGMENT FIELDS</td>
</tr>
<tr>
<td>controlling the content of segment record prefixes and suffixes</td>
<td>EXTRACT</td>
<td>SEGMENT_RECORD_PREFIX SEGMENT_RECORD SUFFIX</td>
</tr>
<tr>
<td>controlling header and trailer records</td>
<td>USER_RECORD</td>
<td>BREAK FIELDS</td>
</tr>
<tr>
<td>sorting the extract file</td>
<td>EXTRACT</td>
<td>SORT SORT_OPTION</td>
</tr>
</tbody>
</table>

For more information about the following topics, see the *Fast Path/EP Series Reference Manual*:

- syntax of commands, subcommands, and keywords that are discussed in this book
- diagrams that show the syntax and available parameters and values for Fast Path Online Suite commands and subcommands

**Selecting the extract file format**

The general format of the extract output file can be controlled by specifying one of three values that are available for the EXTRACT_FORMAT keyword. As summarized in Table 9, each available choice enables a different set of more detailed options for tailoring extract output. Certain functions that affect record layout as well as the output data are available only for particular extract file formats.
Table 9  Data extract tailoring options according to value specified for EXTRACT_FORMAT keyword

<table>
<thead>
<tr>
<th>Function</th>
<th>EXTRACT</th>
<th>HDUNLOAD</th>
<th>USER</th>
</tr>
</thead>
<tbody>
<tr>
<td>selecting segments for input</td>
<td>INCLUDE SEGMENT WHERE</td>
<td>INCLUDE SEGMENT WHERE</td>
<td>INCLUDE SEGMENT WHERE</td>
</tr>
<tr>
<td></td>
<td>INCLUDE SAMPLE_LIMIT SAMPLE_INTERVAL</td>
<td>INCLUDE SAMPLE_LIMIT SAMPLE_INTERVAL</td>
<td>INCLUDE SAMPLE_LIMIT SAMPLE_INTERVAL</td>
</tr>
<tr>
<td>outputting segment records</td>
<td>OUTPUT SEGMENT WHERE</td>
<td>OUTPUT SEGMENT * WHERE</td>
<td>OUTPUT SEGMENT WHERE</td>
</tr>
<tr>
<td>(controlling the generation of segment records</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>to the output file)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>controlling the content of output</td>
<td>OUTPUT FIELDS WHERE</td>
<td>OUTPUT FIELDS WHERE</td>
<td>OUTPUT FIELDS WHERE</td>
</tr>
<tr>
<td>segment records</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>controlling content of the segment record</td>
<td>no user control; standard EXTRACT format</td>
<td>no user control; standard HDUNLOAD format</td>
<td>SEGMENT_RECORD_PREFIX</td>
</tr>
<tr>
<td>prefix</td>
<td>is used.</td>
<td>is used.</td>
<td></td>
</tr>
<tr>
<td>controlling content of the segment record</td>
<td>no user control; no suffix is generated.</td>
<td>no user control; no suffix is generated.</td>
<td>SEGMENT_RECORD_SUFFIX</td>
</tr>
<tr>
<td>suffix</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>controlling header records</td>
<td>no user control; standard EXTRACT header</td>
<td>no user control; standard HDUNLOAD header</td>
<td>USER_RECORD BREAK</td>
</tr>
<tr>
<td>records</td>
<td>records are generated.</td>
<td>records are generated.</td>
<td>FIELDS</td>
</tr>
<tr>
<td>controlling trailer records</td>
<td>no user control; no trailer records are</td>
<td>no user control; standard HDUNLOAD trailer</td>
<td>USER_RECORD BREAK</td>
</tr>
<tr>
<td></td>
<td>generated.</td>
<td>records are generated.</td>
<td>FIELDS</td>
</tr>
<tr>
<td>sorting the extract file</td>
<td>SORT SORT_OPTION</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

**EXTRACT_FORMAT=EXTRACT**

EXTRACT_FORMAT=EXTRACT (the default) specifies that the file be formatted in the standard Fast Path Online Analyzer/EP extract file format. This format should be used when the application that will read the extract data does not have specific formatting requirements or when those requirements have not yet been determined. The record layout that is used in this format is shown in the Fast Path/EP Series Reference Manual.
EXTRACT_FORMAT=EXTRACT lets you control the following processes:

- data expansion (EXPAND keyword)
- selection of segments for input (INCLUDE/EXCLUDE subcommands and associated keywords)
- exclusion of data records for selected segments from the output file (OUTPUT subcommand with optional WHERE keyword)
- segment data content (OUTPUT subcommand with FIELDS keyword)
- sorting of the extract file (SORT and SORT_OPTION keywords)

EXTRACT_FORMAT=EXTRACT does not let you control the following processes:

- segment data record prefix and suffix
- header and trailer records

Figure 66 shows the record layout when EXTRACT_FORMAT=EXTRACT is selected.
**EXTRACT_FORMAT=HDUNLOAD**

EXTRACT_FORMAT=HDUNLOAD specifies that the file be written in a format that is compatible with the IBM HD Reorganization Unload utility. This format is the same format that is used by the Fast Path Reorg/EP unload and reload functions and should be used when the extracted data is to be used as input to the reload process. EXTRACT_FORMAT=HDUNLOAD lets you control the following processes:

- data expansion (EXPAND keyword)
- selection of segments for input (INCLUDE/EXCLUDE subcommands and associated keywords)
- segment data content (OUTPUT subcommand with FIELDS keyword)

**EXTRACT_FORMAT=HDUNLOAD does not** let you control the following processes:

- exclusion of data records for selected segments within a hierarchical path
- segment data record prefix and suffix
- header and trailer records
- sorting of the extract file

*Figure 67* shows the record layout when EXTRACT_FORMAT=HDUNLOAD is selected.

*Figure 67  Map of the HDUNLOAD format extract output file*
EXTRACT_FORMAT=USER

EXTRACT_FORMAT=USER allows the most detailed level of control of the output file. This format should be used when an existing application program requires extracted data in a specific format. EXTRACT_FORMAT=USER lets you control the following processes:

- data expansion (EXPAND keyword)
- selection of segments for input (INCLUDE/EXCLUDE subcommands and associated keywords)
- exclusion of segment data records for selected segments (OUTPUT subcommand with optional WHERE keyword)
- segment data content (OUTPUT subcommand with FIELDS keyword)
- segment data record prefix and suffix (SEGMENT_RECORD_PREFIX and SEGMENT_RECORD_SUFFIX keywords)
- header and trailer records (USER_RECORD subcommand and keywords)

EXTRACT_FORMAT=USER does not let you support sorting of the extract file.

Figure 68 shows the available user-selectable options for record layout when EXTRACT_FORMAT=USER is selected. Keywords used to customize record layouts are shown in boldface type, and are discussed in detail later in this chapter. All areas in the extract output file will have the same layout as specified by the various keywords.

Figure 68  Map of the USER format extract output file
Selecting the database and areas

When specified with the EXTRACT command, the DBD keyword identifies the name of the online DEDB (DBD name) from which data is to be extracted. Because the DBD name is supplied as an execution parameter, the DBD keyword can be omitted. If the keyword is coded, it must specify the same DBD name as supplied in the execution parameter.

**WARNING**

If the DBD keyword does not specify the same DBD name as supplied in the execution parameter, an error message is issued and the online extract function terminates.

The IAREA keyword can be used to select specific areas from which the data is to be extracted. If you omit the IAREA keyword, data is extracted from all areas that are defined in the DEDB.

Areas can be specified on the IAREA keyword by using any combination of area names, area numbers, or a area ranges. The following parameters are available for the IAREA keyword:

- **IAREA=ALL** (default) or **IAREA=*** specifies all areas of the DEDB.
- **IAREA=areaname** specifies one or more areas by using the one-character to eight-character area name for each area specified. Multiple area names must be enclosed in parentheses and separated by commas.
- **IAREA=areanumber** specifies one or more areas by using the one-character to five-character area number for each area specified. Multiple area numbers must be enclosed in parentheses and separated by commas.
- **IAREA=(RANGE=(startarea,endarea))** specifies a consecutive range of areas using either areaname or areanumber parameters. The area number associated with startarea must be less than the area number associated with endarea.

An asterisk (*) can be used to specify all areas of the DEDB. When the * character is used with the RANGE keyword, it can be used to specify the beginning or ending range for specific areas of the DEDB.

To extract data from an entire DEDB, use a command set like the example shown in Figure 69.

**Figure 69  Sample control statement for extracting data from all areas**

```plaintext
EXTRACT DBD=dbdname, IAREA=ALL
```
To extract data only from specific areas, use a command set like the example shown in Figure 70.

**Figure 70  Sample control statement for extracting data from specific areas**

```
EXTRACT DBD=dbname, AREA=(AREANAM1, AREANAM3, RANGE=(5,8))
```

This control statement uses a combination of area names and area numbers to request that data is to be extracted from specific areas. Data will be extracted from the areas named AREANAM1 and AREANAM3. Also, data will be extracted from all consecutive areas from area number 5 to area number 8 (area5, area6, area7 and area8).

### Defining the extract output data set

Extract data sets can be allocated by using DD statements that are supplied in the JCL or by using dynamic allocation. BMC recommends dynamic allocation because of the flexibility and improved control that it provides.

By default, one output extract data set is produced for each input area. You can specify the optional OFILECTL subcommand to combine data that is obtained from one or more input areas into a single output data set. One output extract data set is created for each OFILECTL subcommand that is specified.

Regardless of which format you select for the extract output file, the DCB for each output file will always have the following attributes:

- **RECFM=VB** (variable block)
- **LRECL=** minimum record length for the file calculated automatically by Fast Path Analyzer/EP. You can, however, specify a larger value.
- **BLKSIZE=** calculated automatically by Fast Path Online Analyzer/EP

### Creating a separate data set for each area

If no OFILECTL subcommand is specified, a separate output extract data set is produced for each input area. The DD statement for each output extract data set can be named OAREAxxx, which uses a three-digit area number, or OARxxxxx, which uses a five-digit area number. The OAREAxxx DD statement can only be used for area numbers 1 through 999. The OARxxxxx DD statement can be used for numbers 1 through 2048, and must be used for areas numbers greater than 999.
If the OAREAxxx DD or OARxxxxx DD statement is omitted from the JCL, Fast Path Online Analyzer/EP attempts to dynamically allocate it by using the OUTPUT_DSN_MASK keyword. The data set name that is generated by using this mask must be a preexisting cataloged data set.

**Using standard JCL**

Use JCL DD statements named OAREAxxx, where xxx is the three-digit area number that is listed in the DBD, to indicate the output extract data set produced for each input area. When accessing area numbers greater than 999, use the OARxxxxx DD statement (where xxxxx is the five-digit area number).

The example in Figure 71 requests the creation of one output extract data set for each input area. The EXTRACT command determines the DBD name from the PARM field on the EXEC statement. The data that is extracted from input area areanam1 is written to the OAREA001 DD statement, the data that is extracted from input area areanam2 is written to OAREA002, and the data that is extracted from input area area1050 is written to OAR01050.

**Figure 71 JCL to allocate extract data sets**

```
//PFP EXEC PGM=DFSRRC00,REGION=0M,
//       PARM=(IFP,dbname,DFBFPUO)
//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
//       DD DISP=SHR,DSN=IMS.RESLIB
//       DD DISP=SHR,DSN=BMC.DPK.LOAD
//$$DPICDS DD DISP=SHR,DSN=PFP.DPK.DPICDS
//$$DPITBL DD DISP=SHR,DSN=PFP.DPK.DPITBL
//OAREA001 DD UNIT=TAPE,DISP=(NEW,CATLG),
//         DSN=your.dataset.name1
//OAREA002 DD UNIT=TAPE,DISP=(NEW,CATLG),
//         DSN=your.dataset.name2
//OAR01050 DD UNIT=TAPE,DISP=(NEW,CATLG),
//         DSN=your.dataset.name1050
//PFPSYSIN DD *
//EXTRACT IAREA=(areanam1,areanam2,area1050)
/*
```

**Using dynamic allocation**

To use dynamic allocation, omit the OAREAxxx and OARxxxxx DD statements from the JCL. Use the OUTPUT_DSN_MASK keyword to supply a mask to construct the preexisting cataloged data set name. Substitution variables within the mask can be used to generate unique names for each output data set.
Combining multiple areas into a single data set

You can refer to a relative generation of a generation data group (GDG) by including it in the mask, such as in `gdg-name(-1)`. The relative generation number that you specify must be zero or less. If you refer to a GDG name without specifying a relative generation number, the product automatically appends a relative generation of zero.

The example in Figure 72 requests that a separate preallocated output extract data set be written for each input area.

**Figure 72** Dynamic allocation of extract data sets by using `OUTPUT_DSN_MASK` keyword

```plaintext
EXTRACT
OUTPUT_DSN_MASK='dataset-name-mask'
```

Combining multiple areas into a single data set

Under certain circumstances, consolidation of extracted data from multiple inputs into one extract file might be convenient. You can specify the optional OFILECTL subcommand to combine the data that is obtained from one or more input areas into a single output file. One output extract data set is produced for each OFILECTL subcommand that is specified. Use the OAREA keyword on the OFILECTL subcommand to specify the areas that will be placed into the output extract data set.

**NOTE**

Each input area that is specified by using the IAREA keyword on the EXTRACT command must also be specified in the OAREA keyword of one (and only one) OFILECTL subcommand.

Using OFILECTL with standard JCL

Use the DDNAME keyword (*not* the DSNAME keyword) to indicate that the extract data set is to be allocated by using JCL DD statements.

The example in Figure 73 uses JCL to allocate an extract data set. Extracted data obtained from all areas that are defined in the database are combined and written into a single extract data set.

**Figure 73** JCL to allocate an extract data set by using the DDNAME keyword (part 1 of 2)

```plaintext
//PFP EXEC PGM=DFSRRC00,REGION=0M,
//       PARM=(IFP,dbdname,DBF#FPU0)
//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
//         DD DISP=SHR,DSN=IMS.RESLIB
//         DD DISP=SHR,DSN=BMC.DPK.LOAD
//$$DPICDS DD DISP=SHR,DSN=PFP.DPK.DPICDS
```
Combining multiple areas into a single data set

Chapter 5 Extracting data from an online DEDB

Using OFILECTL with dynamic allocation

Use the DSNAME keyword (not the DDNAME keyword) to indicate that the extract data set is to be dynamically allocated. Use the DSNAME keyword to supply a mask to construct the data set name. Substitution variables within the mask can be used to generate unique names for each output data set.

Additional related keywords, such as UNIT, DISP, and DATACLAS, can be used to control the allocation and disposition of the data set, similar to the corresponding JCL keywords.

The example in Figure 74 requests the creation of two extract data sets. The data that is extracted from input areas areanam1 and areanam2 is written into one extract data set, and the data from areanam3 is written into a separate data set. Both output files are written to tape devices and are recorded in the system catalog on completion.

**NOTE**

The ddname that is used for dynamic allocation is OAREAxxx, where xxx is the three-digit area number of the first area that is listed in the OAREA keyword, relating to the sequential order of the areas that are listed in the DBD. If an OAREAxxx DD statement is supplied in the JCL, it is used instead of the dynamic allocation parameters. If the first area that is listed on the OAREA keyword has an area number greater than 999, the OARxxxxx DD statement is used.

The data set name specified with the DSNAME keyword can be used to construct a GDG by specifying the base name for the GDG. As an option, you can include a relative generation number as part of the DSNAME keyword.
If the data set name matches an existing GDG base name, the product automatically appends the appropriate generation information to the base name that is equivalent to relative generation (+1), if not specified explicitly. DISP=NEW is required when a new generation data set is being created.

The example in Figure 75 requests that all extract data be written to a single data set. The output data set, specified by using the DSNAME keyword, is allocated on disk and is a new generation within a GDG.

### Figure 75 Using a generation data set group

```plaintext
EXTRACT DBD=dbdname
    OFILECTL DSNAME='gdg-dataset-name-mask(+1)',
        DISP=(NEW,CATLG),UNIT=3390,VOLSER=123456,
        SPACE=(CYL,100,10,RLSE)
```

---

## Selecting database segments for input and output

By default, all root segments and their dependent segments (including SDEP segments) are selected. You can modify the selection of segments by using the optional INCLUDE and EXCLUDE subcommands.

The segments that you include (or do not exclude) by using these subcommands are selected and forwarded to the output processing for record formatting. You can also use several keywords with the INCLUDE and EXCLUDE subcommands to further refine the input selection process.

You can use the OUTPUT subcommand to specify only segments that you want written to the output extract file. You can also use selected keywords on the OUTPUT subcommand to apply conditional selection criteria. You cannot, however, specify segments on an OUTPUT subcommand that were excluded from your initial selection with an EXCLUDE subcommand or a conditional INCLUDE subcommand.

### Including and excluding segments for input

The INCLUDE subcommand lets you specify criteria for selecting segments of interest for the extract. If the INCLUDE subcommand selection criteria are met, the selected segments and their dependents are forwarded for output processing.

The EXCLUDE subcommand lets you specify criteria for excluding segments from the extract. If the EXCLUDE subcommand selection criteria are met, the selected segments and their dependents are not forwarded for output processing.
The SEGMENT keyword specifies the name of the segment to which the selection criteria are applied. Additional selection criteria can be specified by using the WHERE keyword, which uses field names or positional identifiers with relational or Boolean logical expressions.

A simple INCLUDE or EXCLUDE subcommand uses the required SEGMENT keyword without additional keywords. Understanding the following rules that apply to the basic subcommands is prerequisite to using additional keywords with INCLUDE and EXCLUDE:

- By default, if no INCLUDE or EXCLUDE statement is used with the EXTRACT command, all segments are selected.
- The INCLUDE subcommand selects the segment that is identified by the SEGMENT keyword and all dependents of the segment. Including a segment in this manner also selects the parent hierarchy chain above the segment.
- The EXCLUDE subcommand excludes the segment that is identified by the SEGMENT keyword and all dependents of the segment. You cannot include a segment and exclude a root or a parent of the segment. Excluding a parent excludes the entire hierarchy beneath the parent.

A basic EXTRACT command with an INCLUDE and EXCLUDE subcommand is shown in Figure 76.

| EXTRACT IAREA=(areanam1,areanam3) |
| INCLUDE SEGMENT=segname1 |
| EXCLUDE SEGMENT=segname2 |

While identifying the segments of interest for the extract by using the INCLUDE and EXCLUDE subcommands, you can apply additional criteria to narrow down the selection by using the following keywords:

- **WHERE** keyword selects segments by applying conditional criteria that is specified in a Boolean expression.
- **SAMPLE_INTERVAL** keyword selects segments by intervals of occurrence.
The SAMPLE_LIMIT keyword sets a limit on the number of segments meeting the criteria that will be selected.

### Applying conditional criteria

You can also narrow segment selection by using the WHERE keyword. This combination applies relational and Boolean operators to apply conditional criteria when selecting segments. For details about the operations and syntax of the WHERE keyword and detailed syntax rules for coding these expressions, see the *Fast Path/EP Series Reference Manual*.

In the example in Figure 77, four segments exist in the database but only two segments (PFTROOT and PFPADDR) are written to the two extracted files. PFPACCT and its dependents are explicitly excluded by the EXCLUDE subcommand. Only the PFPROOT segments that have field FLDNAME equal BMCPFP and a value greater than 10,000 in position 46 are written to the output file. Only the PFPADDR segments (of the selected PFROOTS), where position 23 is equal to 22, are written.

**Figure 77  EXTRACT command using INCLUDE, EXCLUDE, and WHERE**

```
EXTRACT IAREA=(areanam1, areanam3)
  INCLUDE SEGMENT=PFPROOT,
      WHERE=(FLDNAME EQ C’BMCPFP’ AND 46:5P GT P’10000’)
  INCLUDE SEGMENT=PFPADDR,
      WHERE=(23:1X EQ X’22’)
  EXCLUDE SEGMENT=PFPACCT
```

### Selecting specific database records by root key

When the selection criteria for the extract identifies a list of root key values, the online extract process will invoke the randomizer to retrieve these root segments directly. Performance is improved significantly because a sequential read of all root addressable pointers (RAPs) within the area is not necessary to select segments that meet the criteria.

The EXTRACT command will examine the selection criteria specified for the root segment to determine if it identifies root key values for all segments that match the selection criteria. If this condition is met, the randomizer will be used to read the root segments directly.

Root key values are specified by using the WHERE keyword on the INCLUDE subcommand for the root segment. The WHERE keyword must specify a special class of expressions called a root key expression that identifies every root key for the segments requested for inclusion.
The online extract process invokes the randomizer to select root segments under the following conditions:

- The WHERE keyword must include a root key expression that identifies the root key with a logical EQ and a literal.
- The root key can be specified by using its segment search key field as defined in the DBD or by precisely specifying its position and length (with a column:length designation) and a data type.
- The root key expression on the WHERE keyword can be followed by other selection criteria by using a logical AND.
- The root key expression on the WHERE keyword can include multiple root key expression values by using a logical OR.
- Multiple INCLUDE subcommands can be specified under a single EXTRACT command. However, a root key expression must be specified on the WHERE keyword under each INCLUDE subcommand.

The example in Figure 78 requests an online extract for a database containing six segments. Because the EXTRACT command includes three separate INCLUDE subcommands, PFPROOT segments (and their associated dependent segments) that meet any of the following criteria are written to the output file:

- field named ROOTKEY that is equal to BMCPFP
- 6-byte root key field beginning in column 3 that is equal to PFPX4Z
- field named ROOTKEY that is equal to PFP500 and 3-byte field beginning in column 40 that is equal to 123

Because each WHERE keyword in Figure 78 specifies a valid root key expression, the randomizer will be invoked to improve performance of the online extract.

**Figure 78** EXTRACT command using INCLUDE statements that identify root keys and invoke the randomizer

```
EXTRACT AREA=(areanam1,areanam3)
  INCLUDE SEGMENT=PFPROOT,
    WHERE=(ROOTKEY EQ C BMCPFP')
  INCLUDE SEGMENT=PFPROOT,
    WHERE=(3:6 EQ C'PFPX4Z')
  INCLUDE SEGMENT=PFPROOT,
    WHERE=(ROOTKEY EQ C'PFP500' AND 40:3 EQ '123')
```
Using criteria that are applied to alternate segments

You can specify segment selection, based on criteria that are applied to alternate segments, by using a segment-qualified field reference within the WHERE keyword. Only the parentage segments within the hierarchical path from the root can be used for qualification.

In the example in Figure 79, SEGC is selected for extract only if FIELD1 in SEGB is equal to the character string BMC.

**Figure 79 Qualifying field references using WHERE keyword**

```
EXTRACT IAREA=(areanam1,areanam3)
   INCLUDE SEGMENT=SEGC,
     WHERE=(SEGB.FIELD1 EQ 'BMC')
```

Segment sampling

The SAMPLE_INTERVAL keyword can be used to specify segment selection based on the frequency of the segment occurrence. For example, if you have specified SAMPLE_INTERVAL=5, every fifth segment will be selected. For a root segment, every fifth occurrence in the database and all dependents will be selected. For a dependent segment, every fifth occurrence under its parent is selected.

The SAMPLE_LIMIT keyword can be used to specify the maximum number of segment occurrences to be selected. For example, if you have specified SAMPLE_LIMIT=500, the first 500 segments will be selected. For a root segment, the first 500 occurrences and all dependents are selected. For a dependent segment, the first 500 occurrences within its parent are selected.

In the example in Figure 80, only one area is input, so only one output extract file is created. The output extract file is sorted by root key. The result of the INCLUDE subcommand is that every ninth occurrence of segment PFPROOT, and all dependents, are selected, with a maximum of 5000 occurrences of PFPROOT.

**Figure 80 EXTRACT command with SAMPLE_INTERVAL and SAMPLE_LIMIT keywords**

```
EXTRACT IAREA=areanam2,
   EXTRACT_FORMAT=EXTRACT,
   SORT=YES
   INCLUDE SEGMENT=PFPROOT,
     SAMPLE_INTERVAL=9,
     SAMPLE_LIMIT=5000
   EXCLUDE SEGMENT=PFPADDR,
     SAMPLE_LIMIT=5
```
The EXCLUDE subcommand changes the result of the previous INCLUDE for dependent segment PFPADDR. For each PFPROOT segment, the first five segment occurrences for segment PFPADDR are omitted. If, for example, PFPADDR has eight occurrences, the last three segment occurrences would be selected. If under another PFPROOT, segment PFPADDR had only four occurrences, no PFPADDR segments (or their dependents) would be selected.

### Outputting segment records

The optional OUTPUT subcommand can be used to limit the writing of segment records to the output file. By default, if no OUTPUT subcommands are specified, one segment record is written for each selected segment.

When one or more OUTPUT subcommands are specified, segment records are written only for segments that are specified. No output record will be written for segment types that are not specified on any OUTPUT subcommand.

**NOTE**

When EXTRACT_FORMAT=HDUNLOAD is used, a segment record is written for each selected segment, regardless of any OUTPUT subcommands specified.

You can use the OUTPUT subcommand to specify only segments that you want written to the output extract file. You must include a separate OUTPUT subcommand for each segment type that you want included in the output, as shown in Figure 81.

**Figure 81  Using the OUTPUT subcommand to control generation of segment records**

```
EXTRACT IAREA=ALL
   OUTPUT SEGMENT=PFPROOT
```

The writing of the output segment record can be controlled by using conditional qualification that is specified by the optional WHERE keyword on the OUTPUT subcommand. If multiple OUTPUT subcommands are specified for the same segment, the first subcommand in which the selection criteria is satisfied will be used. If the selection criteria is not satisfied for any OUTPUT subcommand, no output record is written for the segment.

**Figure 82** shows an example where conditional qualification is used for a segment record during an online data extract. If the conditional qualification that is specified by the WHERE keyword is met, the segment record is output. If the conditional qualification is not met, the segment record is not output.
Samples of INCLUDE, EXCLUDE, and OUTPUT subcommands

Figure 83 shows the root segment SEGA and its hierarchy. Using this hierarchy as a model, the tables in this section show examples of the EXTRACT command combined with the INCLUDE, EXCLUDE, and OUTPUT subcommands and the resulting segments that are selected (by INCLUDE/EXCLUDE subcommands) and output (by OUTPUT subcommand).

Figure 83  SEGA root segment and its dependents

Two separate tables are presented to show the different segment output results with the HDUNLOAD format because of its segment hierarchical path restrictions:

- Table 10 shows selected segments and segment output when EXTRACT_FORMAT=EXTRACT or EXTRACT_FORMAT=USER has been specified.

- Table 11 shows selected segments and segment output when EXTRACT_FORMAT=HDUNLOAD has been specified.

Table 10  Using INCLUDE, EXCLUDE and OUTPUT subcommands with the EXTRACT command and EXTRACT_FORMAT=EXTRACT or EXTRACT_FORMAT=USER (part 1 of 2)

<table>
<thead>
<tr>
<th>Command set</th>
<th>Selected segments</th>
<th>Output segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTRACT IAREA=ALL</td>
<td>All roots and their dependent segments are selected.</td>
<td>A segment record is written for each selected segment.</td>
</tr>
<tr>
<td>No INCLUDE, EXCLUDE, or OUTPUT subcommands are specified with the EXTRACT command.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXCLUDE SEGMENT=SEGA</td>
<td>No segments are selected.</td>
<td>No segment records are output.</td>
</tr>
</tbody>
</table>
Table 10 Using INCLUDE, EXCLUDE and OUTPUT subcommands with the EXTRACT command and EXTRACT_FORMAT=EXTRACT or EXTRACT_FORMAT=USER (part 2 of 2)

<table>
<thead>
<tr>
<th>Command set</th>
<th>Selected segments</th>
<th>Output segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXCLUDE SEGMENT=SEGA INCLUDE SEGMENT=SEGB</td>
<td>No segments are selected. (Excluding a parent segment excludes all of its dependent segments.)</td>
<td>No segment records are output.</td>
</tr>
<tr>
<td>EXCLUDE SEGMENT=SEGB</td>
<td>SEGA, SDEP, SEGE, SEGF, and SEGG are selected.</td>
<td>A segment record is written for each selected segment.</td>
</tr>
<tr>
<td>INCLUDE SEGMENT=SEGA, WHERE=(selection criteria)</td>
<td>All SEGA segments that meet criteria and their dependents are selected.</td>
<td>A segment record is written for each selected segment.</td>
</tr>
<tr>
<td>EXCLUDE SEGMENT=SDEP EXCLUDE SEGMENT=SEGB EXCLUDE SEGMENT=SEGE OUTPUT SEGMENT=SEGG</td>
<td>SEGA and SEGG are selected.</td>
<td>A segment record is written for each selected SEGG that is selected. (This command set is equivalent to the command set for SEGG.)</td>
</tr>
<tr>
<td>OUTPUT SEGMENT=SEGG</td>
<td>All roots and their dependent segments are selected.</td>
<td>A segment record is written for each SEGG selected.</td>
</tr>
<tr>
<td>INCLUDE SEGMENT=SEGB, WHERE=(selection criteria) EXCLUDE SEGMENT=SEGC OUTPUT SEGMENT=SEGD, WHERE=(selection criteria)</td>
<td>All SEGA segments are selected. SEGB segments that meet the selection criteria in the WHERE keyword are selected if they are children of a selected SEGB segment.</td>
<td>A segment record is written for each selected SEGD segment that meets the criteria on the OUTPUT subcommand.</td>
</tr>
</tbody>
</table>

Table 11 Using INCLUDE, EXCLUDE and OUTPUT Subcommands with the EXTRACT Command and EXTRACT_FORMAT=HDUNLOAD (part 2 of 2)

<table>
<thead>
<tr>
<th>Command set</th>
<th>Segments selected</th>
<th>Segments output</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTRACT IAREA=ALL No INCLUDE, EXCLUDE, or OUTPUT subcommands are specified with the EXTRACT command.</td>
<td>All roots and their dependent segments are selected.</td>
<td>A segment record is written for each selected segment.</td>
</tr>
<tr>
<td>EXCLUDE SEGMENT=SEGA</td>
<td>No segments are selected.</td>
<td>No segment records are output.</td>
</tr>
<tr>
<td>EXCLUDE SEGMENT=SEGA INCLUDE SEGMENT=SEGB</td>
<td>No segments are selected. (Excluding a parent segment excludes all of its dependent segments.)</td>
<td>No segment records are output.</td>
</tr>
<tr>
<td>EXCLUDE SEGMENT=SEGB</td>
<td>SEGA, SDEP, SEGE, SEGF, and SEGG are selected.</td>
<td>A segment record is written for each selected segment.</td>
</tr>
<tr>
<td>INCLUDE SEGMENT=SEGA, WHERE=(selection criteria)</td>
<td>All SEGA segments that meet criteria and their dependents are selected.</td>
<td>A segment record is written for each selected segment.</td>
</tr>
<tr>
<td>EXCLUDE SEGMENT=SDEP EXCLUDE SEGMENT=SEGB EXCLUDE SEGMENT=SEGE OUTPUT SEGMENT=SEGG</td>
<td>SEGA and SEGG are selected.</td>
<td>All SEGA and SEGG segments selected are written.</td>
</tr>
</tbody>
</table>
Controlling segment expansion

The output file can contain data from compressed segments. As an option, the extracted data can be expanded by using the EXPAND keyword. This keyword can be used, regardless of the value that you specified for the EXTRACT_FORMAT keyword:

- **EXPAND=NO** (the default) specifies that the segment data be written to the extract output file unchanged.
- **EXPAND=YES** specifies that compressed segment data be expanded when written to the extract output file.
- **EXPAND=(segment1, segment2, ..., segmentn)** causes only the compressed segments that are listed by name to be expanded when written to the extract output file.

**NOTE**

If the FIELDS keyword is used with the OUTPUT subcommand to refer to the content of a compressed segment, you must expand the segment by using the EXPAND=YES value or the EXPAND=(segment1, segment2, ..., segmentn) syntax on the EXTRACT command. This expansion will ensure that the operands which are specified on the FIELDS keyword correspond to the proper segment columns.

**Figure 84** shows an example of using the EXPAND keyword to selectively specify segment expansion. Only segment1 and segment3 in area1 are expanded; all other segments are written to the extract output file unchanged.

**Figure 84 Selectively specifying segments for expansion with extract**

```
EXTRACT DBD=dbdname, IAREA=area1,
   EXTRACT_FORMAT=EXTRACT, EXPAND=(segment1, segment3)
```
Controlling content of segment records

The OUTPUT subcommand and its associated keywords can be used to control content of the segment record. By default, the output record will contain the full, unmodified contents of the selected segment occurrence.

If you do not want to write the full segment contents, you can identify individual fields to be written to the extract output file by using the FIELDS keyword. The FIELDS keyword specifies a list of expressions; the value of each expression is evaluated and placed into the output record in the order that the expressions are specified. For details about the operations and syntax of the FIELDS keyword and detailed syntax rules for coding expressions, see the Fast Path/EP Series Reference Manual.

**WARNING**

When EXTRACT_FORMAT=HDUNLOAD is specified and the FIELDS keyword is used with the OUTPUT subcommand to change the key of a dependent segment, the segments must remain in ascending key sequence under the root key. Fast Path Online Analyzer/EP does not have a facility to resequence the dependent segments in the extract file. If the change to a dependent segment key is not specified in ascending key sequence, a failure occurs if the extracted data is used as input to a reload process.

**NOTE**

When the FIELDS keyword is used with the OUTPUT subcommand to refer to the content of a compressed segment, you must expand the segment by using the EXPAND=YES value or the EXPAND=(segment1,segment2,...segmentn) syntax on the EXTRACT command. This expansion will ensure that the operands which are specified on the FIELDS keyword correspond to the proper segment columns.

The example in **Figure 85** shows the FIELDS keyword coded on the OUTPUT subcommand.

**Figure 85  EXTRACT command with OUTPUT subcommand and FIELDS keyword**

```
EXTRACT IAREA=ALL
  OUTPUT SEGMENT=PFPROOT,
    FIELDS=(3:5,40:9,ROOTFLD6)
```

In this example, only the root segments in the database are extracted to the output files. Output records for segment PFPROOT are modified, and only three fields are written to the output file. The first field starts in position 3 of PFPROOT and extends five bytes. The second field starts in position 40 of PFPROOT and extends nine bytes. The third field is defined as ROOTFLD6 in the DBD definition.
The content of the output segment record can be controlled by using conditional qualification that is specified by the optional WHERE keyword on the OUTPUT subcommand. By using the optional WHERE keyword on the OUTPUT subcommand, generation of an output record for a segment can be controlled by using conditional qualification. If multiple OUTPUT subcommands are specified for the same segment, the first subcommand in which the selection criteria is satisfied is used. If the selection criteria is not satisfied for any OUTPUT subcommand, no output record is written for the segment.

Figure 86 shows an example where an alternative format for the segment record is selected by using conditional qualification during an online data extract. If the conditional qualification is met, the first OUTPUT subcommand is used and the FIELDS keyword generates a modified segment image. If the conditional qualification is not met, the second OUTPUT subcommand is selected and the unmodified segment is written.

**Figure 86 Using the WHERE keyword to conditionally specify segment content**

```
EXTRACT IAREA=ALL
  OUTPUT SEGMENT=PFPROOT,
    FIELDS=(3:20,C'ABC',23:*)
   WHERE=(23:3 EQ C' ')
  OUTPUT SEGMENT=PFPROOT
```

### Controlling content of segment record prefixes and suffixes

When you select EXTRACT_FORMAT=USER as the general extract formatting option, you can also control the content of the segment prefix and suffix by using the SEGMENT_RECORD_PREFIX and SEGMENT_RECORD_SUFFIX keywords.

The SEGMENT_RECORD_PREFIX keyword specifies a list of expressions. The value of each expression is evaluated and placed into the prefix portion of the segment output record in the order that the expressions are specified. If the SEGMENT_RECORD_PREFIX keyword is omitted, the output record contains no data (zero bytes) in the prefix when EXTRACT_FORMAT=USER has been specified.

The SEGMENT_RECORD_SUFFIX keyword specifies a list of expressions. The value of each expression is evaluated and placed into the suffix portion of the segment output record in the order that the expressions are specified. If the SEGMENT_RECORD_SUFFIX keyword is omitted, the output record contains no data (zero bytes) in the suffix when EXTRACT_FORMAT=USER has been specified.

The example in Figure 87 uses the SEGMENT_RECORD_PREFIX keyword to generate a simple prefix for the output segment records. The prefix will contain the segment name and its concatenated key value.

**Figure 87    Using SEGMENT_RECORD_PREFIX keyword to control content of segment record prefix**

```plaintext
EXTRACT IAREA=ALL, EXTRACT_FORMAT=USER,
    SEGMENT_RECORD_PREFIX=(SEGMENT_NAME, SEGMENT_CKEY)
```

The example in Figure 88 uses the SEGMENT_RECORD_SUFFIX keyword to generate a simple suffix for the output segment records. The suffix will contain the segment code and its hierarchical level.

**Figure 88    Using SEGMENT_RECORD_SUFFIX keyword to control content of segment record suffix**

```plaintext
EXTRACT IAREA=ALL, EXTRACT_FORMAT=USER,
    SEGMENT_RECORD_SUFFIX=(SEGMENT_CODE, SEGMENT_LEVEL)
```

## Controlling header and trailer records

When you select EXTRACT_FORMAT=USER as the general extract formatting option, you can also control the content of the segment header and trailer records by using the optional USER_RECORD subcommand and its associated BREAK and FIELDS keywords.

### Specifying the generation of header and trailer records

The value that is specified for the BREAK keyword defines a trigger event that causes the record to be written. Table 12 shows four options that are available for the BREAK keyword, which correspond to the types of header and trailer records that might be generated.

<table>
<thead>
<tr>
<th>Value for BREAK keyword</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>BREAK=(DATABASE, BEFORE)</td>
<td>database-wide header record</td>
</tr>
<tr>
<td>BREAK=(DATABASE, AFTER)</td>
<td>database-wide trailer record</td>
</tr>
</tbody>
</table>
Defining content of header and trailer records

One output record is generated for each occurrence of the USER_RECORD subcommand. When multiple subcommands with the same BREAK trigger are specified, one record is generated for each subcommand in the same order that the subcommands are specified.

Defining content of header and trailer records

The values that are specified for the FIELDS keyword define the content of the record. The value of each expression is placed into the output record in the order specified. For details about the operations and syntax of the FIELDS keyword and detailed syntax rules for coding expressions, see the Fast Path/EP Series Reference Manual.

The example in Figure 89 uses BREAK and FIELDS keywords with the USER_RECORD subcommand to create a database-wide header record and an area trailer record.

Figure 89 Using BREAK and FIELDS keywords to create database header and trailer records

<table>
<thead>
<tr>
<th>Value for BREAK keyword</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>BREAK=(AREA, BEFORE)</td>
<td>area header record</td>
</tr>
<tr>
<td>BREAK=(AREA, AFTER)</td>
<td>area trailer record</td>
</tr>
</tbody>
</table>

Sorting the extract file

When you select EXTRACT_FORMAT=EXTRACT as the general extract formatting option, you can also sort the extract file. The following options are available for sorting the extract file:

- using the SORT keyword
- using the SORT_OPTION keyword
Using the SORT keyword

The output file that is created by the data extract function can be sorted into root key sequence. The sequence is specified by using the SORT keyword. A sort key is appended to the beginning of the record.

- SORT=NO (the default) specifies that the output file not be sorted. The sequence of the records corresponds to the order in which the data appears in the input area (RAP sequence).

- SORT=YES specifies that the output file be sorted. The sequence of the records corresponds to the symbolic key of the root segments.

When the output file is sorted, the layout of the data records is changed. When sorting extract output files, Fast Path Online Analyzer/EP interacts with the sort product that is used at your site to pass appropriate sort work space and sort message information. The sorted and unsorted record layouts are shown in the Fast Path/EP Series Reference Manual.

Using the SORT_OPTION keyword

The SORT_OPTION keyword provides a convenient method for supplying optional sort tuning parameters to your site’s Sort utility.

SORT_OPTION=DYNALLOC (the default) specifies that sort work space will be dynamically allocated according to your Sort utility’s installation defaults.

For more information, see the reference manual for the sort product that is used at your site.

Sample EXTRACT command scenarios

For scenarios that show how to use the online EXTRACT command with key related keywords and subcommands, see Appendix C, “Sample online command scenarios.”
Making an online image copy

This chapter provides information on the online image copy function that is provided by Fast Path Online Image Copy/EP. This chapter also discusses how offline image copies of DEDBs can be created simultaneously during execution of several Fast Path Online Suite product functions.

This chapter discusses the following topics:

Introduction to the image copy function .................................................. 139
   Control region requirement ................................................................. 140
   Image copy function inputs and outputs ............................................. 140
IMAGECOPY command keywords and subcommands .......................... 140
Selecting the database and areas .......................................................... 141
Allocating the image copy data set ....................................................... 143
   Using JCL to allocate the image copy data set .................................. 143
   Using dynamic allocation for the image copy data set ..................... 143
Stacking image copy data sets .............................................................. 144
Generating multiple output image copies ............................................ 145
Recording the image copy within DBRC ............................................... 146
Compressing the output image copy ................................................... 146
Analyzing the area during image copy ............................................... 147
Making an image copy during reorganization ..................................... 149
Sample IMAGECOPY command scenarios .......................................... 149

Introduction to the image copy function

Use the image copy function to generate one or more image copies of a DEDB. The IMAGECOPY command executes the Fast Path Online Image Copy/EP product.

Many Fast Path Online Suite product functions provide the facility to create an image copy of the DEDB simultaneously with the execution of the function. An image copy is requested by using the IC subcommand with the function.
Control region requirement

Specifying the IC subcommand with an online function creates an image copy by using Fast Path Online Image Copy/EP, and a license for the product is required.

Control region requirement

To create an online image copy, you must define the database to the IMS Control Region with ACCESS=UP.

Image copy function inputs and outputs

The image copy function has one input and one or more outputs as shown in Figure 90.

Figure 90  Image copy function inputs and outputs

IMAGECOPY command keywords and subcommands

Table 13 lists the keywords and subcommands that are available for the IMAGECOPY command.
Table 13  IMAGECOPY command keywords and subcommands

<table>
<thead>
<tr>
<th>Function</th>
<th>Command or subcommand</th>
<th>Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>selecting the database and areas</td>
<td>IMAGECOPY</td>
<td>DBD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IAREA</td>
</tr>
<tr>
<td>allocating the image copy data set</td>
<td>IC</td>
<td>DDNAME</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSNAME</td>
</tr>
<tr>
<td>stacking image copy data sets</td>
<td>IC</td>
<td>STACK_NAME</td>
</tr>
<tr>
<td>generating multiple output image copies</td>
<td>IC</td>
<td>all associated keywords</td>
</tr>
<tr>
<td>recording the image copy with DBRC</td>
<td>IC</td>
<td>NOTIFY</td>
</tr>
<tr>
<td>compressing the output image copy</td>
<td>IC</td>
<td>COMPRESSION</td>
</tr>
<tr>
<td>analyzing the area during an image copy</td>
<td>IMAGECOPY</td>
<td>POINTER_VALIDATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RAP_VALIDATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SDEP_VALIDATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LARGEST_DATABASE_RECORDS</td>
</tr>
<tr>
<td></td>
<td>REPORT</td>
<td>all associated keywords</td>
</tr>
<tr>
<td></td>
<td>THRESHOLD</td>
<td>all associated keywords</td>
</tr>
</tbody>
</table>

For more information about the following topics, see the *Fast Path/EP Series Reference Manual*:

- syntax of commands, subcommands, and keywords that are discussed in this book
- diagrams that show the syntax and available parameters and values for Fast Path Online Suite commands and subcommands

**Selecting the database and areas**

The DBD keyword identifies the name of the DEDB (DBD name) for which an image copy will be generated. For an online image copy, the DBD name is supplied as an execution parameter, and the DBD keyword can be omitted. If the DBD keyword is coded, it must specify the same DBD name as supplied in the execution parameter.

**WARNING**

If the DBD keyword does not specify the same DBD name as supplied in the execution parameter, an error message is issued and the online image copy function terminates.
The IAREA keyword can be used to request that an image copy be generated for specific areas in the DEDB. If you omit the IAREA keyword, an image copy will be generated for all areas that are defined in the DEDB.

Areas can be specified on the IAREA keyword by using any combination of area names, area numbers, or area ranges. The following parameters are available for the IAREA keyword:

- **IAREA=ALL** (default) or IAREA=* specifies all areas of the DEDB.
- **IAREA=areaname** specifies one or more areas by using the one-character to eight-character area name for each area specified. Multiple area names must be enclosed in parentheses and separated by commas.
- **IAREA=areanumber** specifies one or more areas by using the one-character to five-character area number for each area specified. Multiple area numbers must be enclosed in parentheses and separated by commas.
- **IAREA=(RANGE=(startarea,endarea))** specifies a consecutive range of areas using either areaname or areanumber parameters. The area number associated with startarea must be less than the area number associated with endarea.

An asterisk (*) can be used to specify all areas of the DEDB. When the * character is used with the RANGE keyword, it can be used to specify the beginning or ending range for specific areas of the DEDB.

To generate an image copy of an entire DEDB, use a command set like the example shown in Figure 91.

**Figure 91  Sample control statement for generating image copy of all areas**

```
IMAGECOPY DBD=dbdname,IAREA=ALL
```

To generate an image copy of specific areas within a DEDB, use a command set like the example shown in Figure 92.

**Figure 92  Sample control statement for generating image copy of specific areas**

```
IMAGECOPY DBD=dbdname,
  IAREA=(AREANAM1,AREANAM3,RANGE=(5,8))
```

This control statement uses a combination of area names and area numbers to request that an image copy be generated for specific areas. The areas named AREANAM1 and AREANAM3 will have an image copy generated. Also, all consecutive areas from area number 5 to area number 8 (area5, area6, area7 and area8) will have an image copy generated.
Allocating the image copy data set

The output image copy data set can be allocated by using DD statements supplied in the JCL or by using dynamic allocation. BMC recommends dynamic allocation because of the flexibility and improved control that it provides.

Using JCL to allocate the image copy data set

Use the DDNAME keyword to indicate that the image copy data set is allocated by using JCL DD statements.

The example in Figure 93 shows the request to make an image copy of an area. Since the IMAGECOPY command runs in the online environment, the database name is supplied in the JCL using the PARM keyword on the EXEC statement. The output image copy is written by the DD statement named icddname that is supplied in the JCL.

**Figure 93  JCL to allocate an image copy data set**

```
//PFP EXEC PGM=DFSRRC00,REGION=0M,
  // PARM=(IFP,dbdname,DBF#FPUO)
  //STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
  // DD DISP=SHR,DSN=IMS.RESLIB
  //icddname DD UNIT=TAPE,DISP=(NEW,CATLG),
  // DSN=your.dataset.name
  //PFPSYSIN DD *
  IMAGECOPY IAREA=areaname
  IC DDNAME=icddname */
```

**NOTE**

When using JCL to allocate the output image copy, ensure that only one area is written to a DD statement. If more than one area is written to the same DD statement, the image copy of the second area overwrites the image copy of the first area.

Using dynamic allocation for the image copy data set

Use the DSNAME keyword to indicate that the image copy data set is to be dynamically allocated, and to supply a mask to be used for constructing the data set name. Substitution variables within the mask can be used to generate unique names for each output data set.
Additional keywords, such as UNIT, DISP and DATACLAS, can be used to control the allocation and disposition of the data set, similar to the corresponding JCL keywords.

The example in Figure 94 requests the creation of a single output image copy data set for each area (using the default IAREA=ALL) within the database. Each image copy data set is written to a tape device and is recorded in the system catalog.

**Figure 94  Dynamic allocation of an image copy data set**

```
IMAGECOPY DBD=dbname
  IC  DSNAME='dataset-name-mask',
      DISP=(NEW,CATLG),UNIT=TAPE
```

The data set name specified with the DSNAME keyword can refer to a generation data set group (GDG). Code the DSNAME keyword to specify the base name for the GDG. You can include the (+1) relative generation number as part of the DSNAME keyword.

The example in Figure 95 uses a DSNAME keyword to refer to a GDG.

**Figure 95  Using a GDG**

```
IMAGECOPY
  IC  DSNAME='dataset-name-mask(+1)',
      DISP=(NEW,CATLG),UNIT=TAPE
```

If you do not specify the (+1) relative generation number, the product automatically appends the incremented (+1) generation number to the data set name if it is recorded in the system catalog as a GDG base name.

**Stacking image copy data sets**

Use the STACK_NAME keyword to specify the name of a stack group to which the image copy data set belongs. The same name is specified for each image copy that is to be included in the group. All image copy data sets that are part of the same stack group are written onto the same tape volume (or volumes) as file number 1, 2, 3, and so on.

**Figure 96** specifies that an image copy data set be created for each area in the database. The output data sets are all to be written to the same tape volume.
Generating multiple output image copies

Dynamic allocation is always used for stacked data sets. When an IC subcommand containing a unique value for the STACK_NAME keyword is encountered, values that are specified for the UNIT, VOLCNT, VOLSER, EXPDT and RETPD keywords (if any) on the IC subcommand are saved; these values will apply to all image copy data sets that are written to the stack. The UNIT keyword must be specified and must refer to a tape device type. All other dynamic allocation keywords, such as DSNAME, pertain to the individual output image copy data set that is being defined.

DISP=NEW is required for stacked data sets; there are no restrictions on the remaining parameters for the DISP keyword.

In Figure 97, two image copy data sets are requested for each area in the database. You cannot request that both copies be written to the same stack. You can, however, specify a different STACK_NAME keyword on each IC subcommand. You can also mix stacked and non-stacked image copy processing within the same primary command.

Generating multiple output image copies

One output image copy data set is produced for each IC subcommand that is specified. The product imposes no limit on the number of duplicate image copies that can be produced simultaneously.

The example in Figure 98 shows a request to make three image copy data sets for each area within the database. The first and second image copy data sets are written to tape devices and are recorded in the system catalog. The third image copy is written to an existing cataloged data set and overwrites the prior contents of the data set.

Figure 96  Using the STACK_NAME keyword

```
IMAGECOPY DBD=dbname,IAREA=ALL
  IC STACK_NAME=NAME1,UNIT=TAPE,DISP=(NEW,CATLG),
    DSNNAME='dataset-name-mask'
```

Figure 97  Unacceptable use of the same stack name within a single command

```
IMAGECOPY DBD=dbname,IAREA=ALL, ...
  IC STACK_NAME=NAME1, ...
  IC STACK_NAME=NAME1, ...
```

Figure 98  Specifying multiple image copy data sets for each area

```
IMAGECOPY
  IC DSNNAME='dataset-name1-mask',
    DISP=(NEW,CATLG),UNIT=TAPE
  IC DSNNAME='dataset-name2-mask',
    DISP=(NEW,CATLG),UNIT=TAPE
  IC DSNNAME='dataset-name3-mask',DISP=SHR
```
Recording the image copy within DBRC

If the area being processed is registered within DBRC, the image copy data set or data sets that are produced will be recorded by using the NOTIFY.IC process. There is no limit to the number of image copy data sets you can create, but no more than two image copy data sets can be recorded within DBRC. You can control which image copy data set or data sets are recorded by using the NOTIFY keyword.

NOTIFY=YES (the default) indicates that the image copy data set is eligible to be recorded in DBRC unless an error occurs during the creation of the image copy data set.

NOTIFY=NO indicates that the image copy data set is not eligible to be recorded in DBRC even if no errors occur during the creation of the image copy data set. Specifying this value can be useful when producing an image copy that is intended for offsite recovery purposes.

If more than two image copy data sets are created successfully, the first two eligible image copies (in the order that the IC subcommands are specified) are recorded within DBRC.

The example in Figure 99 requests the creation of three output image copies. The first two image copies are eligible to be recorded within DBRC, but the third image copy is not eligible. If an error occurs during creation of the first image copy, it is not recorded; the second image copy is the only image copy recorded, because the third image copy is not eligible (NOTIFY=NO is specified). If NOTIFY=YES was specified on the third image copy, it would have been recorded in DBRC.

Figure 99 Recording image copies with DBRC

<table>
<thead>
<tr>
<th>IMAGECOPY IAREA=areaname</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC DDNAME=ddname1, NOTIFY=YES</td>
</tr>
<tr>
<td>IC DDNAME=ddname2, NOTIFY=YES</td>
</tr>
<tr>
<td>IC DDNAME=ddname3, NOTIFY=NO</td>
</tr>
</tbody>
</table>

Compressing the output image copy

The image copy data sets produced by the Fast Path Online Image Copy/EP functions conform to the standard format used by IMS. You can apply a BMC image copy compression algorithm to reduce the size of the output data set. Compressed image copies are fully compatible with all BMC IMS products (such as RECOVERY PLUS for IMS) but must be expanded before they can be used by other products.

COMPRESSION=None (the default) specifies that the output image copy data set is not compressed.
A BMC compression algorithm can be selected by using the other values for the COMPRESSION keyword. Valid values are FSE, CCC, and DPE.

- **COMPRESSION=FSE** (free space element elimination) does not write free space elements or free blocks to the output. No other compression is performed.

- **COMPRESSION=CCC** (common character compression) compresses repeating groups of low-values, high-values, zeros, blanks, or combinations of these characters. Free blocks are not written to the output. This option compresses the prefix and key portions of the database records, if possible.

- **COMPRESSION=DPE** (full character compression) compresses all character types by using a subset of the BMC DATA PACKER/IMS product. Free blocks are compressed and written to the output. This option compresses the prefix and key portions of the database records, if possible.

The example in Figure 100 requests the creation of three output image copies. The first image copy is not compressed. The second image copy data set is compressed by using the FSE algorithm. The third image copy data set is compressed by using the CCC algorithm.

**Figure 100  Compressing the image copy data set**

```
IMAGECOPY  IAREA=areaname
  IC DDNAME=ddname1,COMPRESSION=NONE
  IC DDNAME=ddname2,COMPRESSION=FSE
  IC DDNAME=ddname3,COMPRESSION=CCC
```

### Analyzing the area during image copy

If your site has a license for Fast Path Online Analyzer/EP, pointer validation will occur automatically (by default) when you make an online image copy of an area by using the IMAGECOPY command. This automatic process applies the default value of QUICK for the POINTER_VALIDATION keyword to perform a checksum validation of pointers for each segment type within each UOW of the area that is specified on the IMAGECOPY command.

Automatic pointer validation provides an assurance of the area’s pointer integrity, while providing statistics that show how the segment data affected space usage.

**NOTE**

You can override automatic quick pointer validation and request more extensive validation by specifying POINTER_VALIDATION=FULL on the IMAGECOPY or GLOBAL command. Direct pointers are validated by using the cross-reference technique. You can also specify any other valid value for the POINTER_VALIDATION keyword (such as POINTER_VALIDATION=NONE to completely disable the analysis function).
If an SDEP segment is defined for the database, SDEP pointers are also validated by default by using the same technique as specified for direct pointers. You can manually specify another valid value for the SDEP_VALIDATION keyword to control how SDEP pointers are to be validated.

You can also specify additional functions that are associated with the analysis function:

- Use the RAP_VALIDATION keyword to control how RAPs are processed.
- Use the LARGEST_DATABASE_RECORDS keyword to specify the number of largest database records to be tracked by the analysis process.
- Use the REPORT and THRESHOLD subcommands to control the generation of analysis reports and exception testing.

The example in Figure 101 requests the creation of a single output image copy data set for each area within the database. By omission of a value for POINTER_VALIDATION, quick analysis is also performed for each area. SDEP pointers (if any exist) are validated using the same technique. (See “Pointer validation” on page 80 for more detailed information.) Only the Free Space Analysis report is produced.

Making an online image copy during an online analysis process is essentially the same as performing an online analysis while making an online image copy. The example in Figure 102 is functionally equivalent to the example in Figure 101. The main difference is that you are actively requesting the analysis (instead of it occurring automatically with the IMAGECOPY command).

For more information about analyzing areas, see Chapter 4, “Analyzing a DEDB online.”
Making an image copy during reorganization

You can request that one or more image copy data sets be created for an area during the online reorganization function. Use the IC subcommand to request creation of an image copy.

The example in Figure 103 requests the creation of two output image copy data sets for each area within the database as the areas are being reorganized.

**Figure 103  Specifying the image copy function with the REORGANIZE command**

```
REORGANIZE  DBD=dbdname, IAREA=ALL
  IC  DSNAME='dataset-name1-mask',
      DISP=(NEW,CATLG),UNIT=TAPE
  IC  DSNAME='dataset-name2-mask',
      DISP=(NEW,CATLG),UNIT=TAPE
```

For more information about reorganizing areas, see Chapter 3, “Reorganizing a DEDB online.”

**Sample IMAGECOPY command scenarios**

For scenarios that show how to use the online IMAGECOPY command with key related keywords and subcommands, see Appendix C, “Sample online command scenarios.”
Extending a DEDB online

This chapter provides information on the capabilities, setup, and use of the online Extend function that is provided by Fast Path Online Reorg/EP. This function lets you increase the size of IOVF or SDEP storage portions of a DEDB while they are online to IMS.

This chapter discusses the following topics:

- Extend function overview ................................................................. 152
  - User-specified control of the extend ........................................ 152
    - Automatic pointer validation during an extend .................. 152

- Extend function requirements ...................................................... 152
  - Control region requirement ............................................... 152
  - DEDB maintenance requirements ...................................... 153

- Extend function inputs and outputs ............................................. 153

- Selecting the type of extend ....................................................... 154
  - Online extend ................................................................. 154
  - Control statement ......................................................... 154
  - Performance considerations ............................................. 155
  - MADS considerations ...................................................... 156

- EXTEND command keywords and subcommands .............................. 156

- Selecting the database and areas ................................................ 157
- Allocating the area data set ..................................................... 158
- Specifying the execution mode ................................................... 158

- Extending IOVF and SDEP storage portions ................................ 158
  - Extending IOVF .............................................................. 159
  - Extending SDEP .............................................................. 161

- Analyzing the DEDB during an extend ....................................... 163
- Making an image copy during an extend .................................... 164
- Sample EXTEND command scenarios .......................................... 164
Extend function overview

The patented Extend function of Fast Path Online Reorg/EP offers a high-performance alternative for extending an area data set. It allows you to extend space without performing a database reorganization.

If an out-of-space condition occurs in either the IOVF or SDEP portion of a DEDB area, you can use the Extend function to increase the size of the IOVF or SDEP portions without performing an online or an offline reorganization. The ability to use this function to extend the IOVF or SDEP portions of the area can save time and resources when compared to executing the traditional unload and reload processes to accommodate the space extend.

The Extend function, which is executed by the EXTEND command is available for both offline and online processing.

User-specified control of the extend

User-specified keywords are available to let you control the DEDB extend process manually. You can extend all areas or selected areas of a DEDB.

Automatic pointer validation during an extend

If you have a license for the Fast Path Online Analyzer/EP product, the product performs quick (checksum) pointer validation by default when you execute the online Extend function.

Extend function requirements

The following control region and DEDB maintenance requirements apply to the online Extend function.

Control region requirement

When performing an online extend, you must define the database to the IMS Control Region with ACCESS=UP.
DEDB maintenance requirements

The following maintenance requirements apply to the online Extend function when you increase the size of IOVF or SDEP storage portions of a DEDB:

- After extending IOVF, you must update the DBD and the ACB with the increased ROOT size because the ACB is not updated with the increased ROOT parameter when the extend for IOVF has been committed. However, the in-core DMAC is updated with the increased ROOT parameter.

If the ACB is not updated and the area is ever recovered, reloaded, or reinitialized to a smaller-sized IOVF, IMS will not bring the area data set online.

- After extending IOVF OR SDEP, you might need to change your area data set VSAM cluster definition to allow for the extended size of the IOVF or SDEP portion the next time that you perform maintenance on the DEDB.

Extend function inputs and outputs

The Extend function will dynamically increase IOVF or SDEP space in the existing area data set. The same area data set is used as input to (and output from) the Extend function.

Possible inputs and outputs for an online Extend function are shown in Figure 104.

Figure 104  Extend function inputs and outputs
The Extend function always issues a summary report to the report log when extend activities are complete. For an example of the Extend Report, see the PFPXCTND member in the REPORTS data set.

## Selecting the type of extend

Fast Path Online Reorg/EP extends a DEDB without taking it offline from IMS.

Because Fast Path Online Reorg/EP incorporates all batch functionality that is provided by the Fast Path Reorg/EP product, you can also perform offline extends by using Fast Path Online Reorg/EP. For more information, see the *Fast Path Offline Suite User Guide*.

### Online extend

The online extend process extends a DEDB without taking it offline. Performing an online extend requires a license for Fast Path Online Reorg/EP.

### Control statement

The PFPSYSIN control statements include the command set necessary to run the online Extend function. A sample control statement for an online extend of IOVF is shown in Figure 105. Extending is limited to only the area that is specified with the IAREA keyword.

**Figure 105  Sample control statement for an online extend of IOVF**

```plaintext
//PFP EXEC PGM=DFSRRCOO,REGION=0M,
//      PARM=(IFP,dbdname,DBF#FPUO)
//STEPLIB DD DSN=BMC.PFP.LOAD,DISP=SHR
// DD DSN=IMS.RESLIB,DISP=SHR
//PFPSYSIN DD *
  EXTEND DBD=bdname,IAREA=areaname,
  EXTEND_IOVF=(UNITS_OF_WORK,200)
/*
```
Performance considerations

This section discusses performance considerations for the online Extend function.

Analyzing the area while extending

If you have a license for the Fast Path Online Analyzer/EP product, the product performs quick (checksum) pointer validation by default during the online Extend function. The I/O that is required to read the UOW and IOVF control intervals is shared between the two functions.

For more information, see “Analyzing the DEDB during an extend” on page 163. For example, you can control the level of pointer validation or turn pointer validation off, depending on your performance considerations.

Creating an output image copy while extending

If you create one or more output image copies while performing an online extend, the I/O that is required to read the UOW control intervals is shared between the two functions. A license for the Fast Path Online Image Copy/EP product is required.

The image copy process requires the IOVF control intervals to be processed sequentially. The I/O that is required to read the IOVF control intervals for the image copy is not shared. The I/O that is required for processing of the SDEP control intervals is shared between image copy and analysis functions.

You can use the IC subcommand to request creation of an image copy for each area during the extend. DBRC is informed (NOTIFY.IC) that a concurrent image copy has been created.

Message suppression

For each anomaly that is encountered in an area, the Extend function generates a message with a specific suffix (severity) level. When the number of places that a particular condition exists is large, a large number of messages is produced. You can reduce the number of repetitious messages produced by using the MESSAGE_SUPPRESSION keyword to specify the maximum number of times that an informational, warning, error, or critical message is to be produced. For the online extend process, this keyword functions in the same manner as it does for the Fast Path/EP online analysis process. For more information, see “Suppressing repetitious messages” on page 95.
MADS considerations

Fast Path Online Reorg/EP fully supports multiple area data sets (MADS) for online extends. All online MADS that are registered for each area are extended simultaneously.

EXTEND command keywords and subcommands

Table 14 lists the keywords and subcommands that are available for the EXTEND command.

<table>
<thead>
<tr>
<th>Function</th>
<th>Command or subcommand</th>
<th>Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>selecting the database and area</td>
<td>EXTEND</td>
<td>DBD IAREA</td>
</tr>
<tr>
<td>specifying the execution mode</td>
<td>EXTEND</td>
<td>TYPE_RUN</td>
</tr>
<tr>
<td>extending IOVF or SDEP storage portions</td>
<td>EXTEND</td>
<td>EXTEND_IOVF EXTEND_SDEP</td>
</tr>
<tr>
<td>analyzing the DEDB while extending</td>
<td>EXTEND</td>
<td>POINTER_VALIDATION RAP_VALIDATION SDEP_VALIDATION LARGEST_DATABASE_RECORDS all associated keywords all associated keywords</td>
</tr>
<tr>
<td>suppressng repetitious messages</td>
<td>EXTEND</td>
<td>MESSAGE_SUPPRESSION ORPHANED_SDEP_MSG</td>
</tr>
<tr>
<td>making an image copy of a DEDB while extending</td>
<td>IC</td>
<td>all associated keywords</td>
</tr>
</tbody>
</table>

For more information about the following topics, see the Fast Path/EP Series Reference Manual:

- syntax of commands, subcommands, and keywords that are discussed in this book
- diagrams that show the syntax and available parameters and values for Fast Path Online Suite commands and subcommands
Selecting the database and areas

The DBD keyword identifies the name of the DEDB (DBD name) to be extended. For an online extend, the DBD name is supplied as an execution parameter, and the DBD keyword can be omitted. If the DBD keyword is coded, it must specify the same DBD name as supplied in the execution parameter.

**WARNING**

If the DBD keyword does not specify the same DBD name as supplied in the execution parameter, an error message is issued and the online Extend function terminates.

The IAREA keyword can be used to select specific areas to be extended. If you omit the IAREA keyword, *all* areas that are defined in the DEDB are extended.

Areas can be specified on the IAREA keyword by using any combination of area names, area numbers, or area ranges. The following parameters are available for the IAREA keyword:

- **IAREA=ALL** (default) or **IAREA=** specifies all areas of the DEDB.
- **IAREA=areaname** specifies one or more areas by using the one-character to eight-character area name for each area specified. Multiple area names must be enclosed in parentheses and separated by commas.
- **IAREA=areanumber** specifies one or more areas by using the one-character to five-character area number for each area specified. Multiple area numbers must be enclosed in parentheses and separated by commas.
- **IAREA=(RANGE=(startarea,endarea))** specifies a consecutive range of areas using either areaname or areanumber parameters. The area number associated with startarea must be less than the area number associated with endarea.

An asterisk (*) can be used to specify all areas of the DEDB. When the * character is used with the RANGE keyword, it can be used to specify the beginning or ending range for specific areas of the DEDB.

To extend IOVF for all areas in an entire DEDB, use a command set like the example shown in Figure 106.

**Figure 106 Sample control statement for extending IOVF for all areas**

```
EXTEND DBD=dbdname,IAREA=ALL,
    EXTEND_IOVF=(UNITS_OF_WORK,100)
```

To extend SDEPs for specific areas, use a command set like the example shown in Figure 107.
Allocating the area data set

For an online extend, the area data set is accessed by using IMS control region services. No JCL is needed for the area data set; if present, it is ignored.

Specifying the execution mode

When using the EXTEND command, you can use the TYPE_RUN keyword to specify the execution mode for an area extension. The following values are available:

- EXECUTE (default) performs an area extension.
- SIMULATE imitates (models) an area extension without actually executing the extension.

For more information about the TYPE_RUN keyword, see the Fast Path/EP Series Reference Manual.

Extending IOVF and SDEP storage portions

If an out-of-space condition occurs in IOVF or SDEP, you can use the Fast Path Online Reorg/EP patented Extend feature to extend the IOVF or SDEP portions of a DEDB. The ability to extend the IOVF or SDEP portions of the area can save time and resources in comparison to using offline CHANGE or UNLOAD/RELOAD commands. For more information about using the extend feature during reorganization, see “Extending IOVF and SDEP storage portions” on page 57 in Chapter 3, “Reorganizing a DEDB online.”

If you use the Extend feature with an online reorganization and the area is registered with Virtual Storage Option (VSO), the IMS VUNLOAD command must be issued before executing the reorganization.
Depending on how many IOVF or SDEP portions that you request, the number of control intervals (CIs) that are required to accommodate your request is rounded up to the next control area (CA) boundary. If additional CIs must be added as a result of the rounding, the additional CIs are added to the SDEP portion of the database.

IOVF can be extended online when there are SDEPs defined. However, the minimum number of blocks that are being added must be as large as the number of SDEP blocks that are currently in use. The number of SDEP blocks that are in use can be reduced by executing the SDEP Scan/Delete utilities before executing the Extend function.

If you are extending IOVF, and the SDEP portion of the area is in a wrapped condition, the SDEP blocks will be rearranged to unwrap the SDEP portion.

If you are extending SDEP only, and the SDEP portion of the area is in a wrapped condition, the SDEP blocks will not be unwrapped. The existing SDEP blocks are rearranged so that the newly created SDEP blocks fall between the SDEP Logical End (LE) and SDEP Logical Begin (LB) points. As a result, the new blocks are immediately available for use; there is no need to run the SDEP Scan/Delete utilities. This action occurs only if the number of new SDEP blocks exceeds the number of used SDEP blocks between the LB point and the original end of file.

The following sections present step-by-step procedures for using the EXTEND command to extend IOVF or SDEP.

## Extending IOVF

The IOVF portion can be extended by specifying units of work, control intervals, or cylinders. The EXTEND_IOVF keyword allows the EXTEND command to extend the IOVF portion.

### To extend the IOVF

1. Ensure that adequate space is present on the primary volume of the target area data set to accommodate the request before trying to increase the IOVF portion during the Extend function. If adequate space is not present, allocate additional volumes to the area data set by using IDCAMS ALTER ADDVOLUME.

   **NOTE**
   
   If an IDCAMS ALTER ADDVOLUME is used while the area data set is opened and allocated (online to IMS), the data set must be closed and reopened so that VSAM can recognize the added volume. Use the /DBR command to close the data set, and the /STA command to reopen the data set.

2. Run the Extend function by using the EXTEND_IOVF keyword to specify the amount of space to add.
When the Extend function is complete, the EXTEND Area Summary Report from the PFPRPTS DD statement shows the new ROOT parameter for the DBD source. Use the values that are indicated to update the DBD definition to include the IOVF extension. Message BMC110000I is also issued, indicating the extended blocks and that the DMAC update has completed.

### Sample control statements for extending IOVF

Sample control statements for extending IOVF by using units of work are shown in Figure 108.

**Figure 108** Sample control statements for extending IOVF by 50 units of work

```sql
EXTEND  DBD=dbdname, IAREA=areaname, 
       EXTEND_IOVF=(UNITS_OF_WORK,50)
```

Sample control statements for extending IOVF by using cylinders are shown in Figure 109.

**Figure 109** Sample control statements for extending IOVF by 50 cylinders

```sql
EXTEND  DBD=dbdname, IAREA=areaname, 
       EXTEND_IOVF=(CYLINDERS,50)
```

### Sample output

The example in Figure 110 shows messages for BMC110000I and PFPRPTS information that are generated when adding 50 units of work to the IOVF portion.

**Figure 110** Sample BMC110000I messages and PFPRPTS information (part 1 of 2)
The following message is issued by IMS in the control log for the IMSPILEX data sharing IMS Partners:

BMC110000I  dbdName, areaName: AREA extension committed date, time

**Subsequent maintenance on the DEDB**

When the extend for IOVF has been committed, the in-core DMAC is updated with the increased ROOT parameter. However, because the ACB is *not* updated with the increased ROOT parameter, you must update the DBD and the ACB with the increased ROOT size.

If the ACB is not updated and the area is ever recovered, reloaded, or reinitialized to a smaller-sized IOVF, IMS will not bring the area data set online.

**To ensure that the extended IOVF is applied to the DBD definition and that the ACB definition is revised**

1. Update the ROOT parameter of the AREA statement in the DBD source with the values that you recorded from the EXTEND Summary Report from the PFPRPTS DD statement.
2. Run DBDGEN.
3. Run ACBGEN.

**NOTE**

You might need to change your area data set VSAM cluster definition to reflect the extended size.

---

**Extending SDEP**

The SDEP storage portion can be extended in control intervals, units of work, or cylinders. To extend the SDEP storage area, use the EXTEND_SDEP keyword with the EXTEND command to specify the amount of space to be added.

---

### Figure 110 Sample BMC110000I messages and PFPRPTS information (part 2 of 2)

<table>
<thead>
<tr>
<th>Description</th>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Independent Overflow CI's</td>
<td>3,500</td>
<td>4,000</td>
</tr>
<tr>
<td>IOVF Space Map CI's</td>
<td>30</td>
<td>34</td>
</tr>
<tr>
<td>IOVF Data CI's</td>
<td>3,470</td>
<td>3,966</td>
</tr>
<tr>
<td>Used IOVF CI's</td>
<td>71 (2%)</td>
<td>71 (2%)</td>
</tr>
<tr>
<td>Free IOVF CI's</td>
<td>3,399 (98%)</td>
<td>3,895 (98%)</td>
</tr>
</tbody>
</table>
To extend SDEP storage without reorganization

1 Ensure that adequate space is present on the primary volume of the target area data set to accommodate the request before trying to increase the SDEP portion during the Extend function. If adequate space is not present, allocate additional volumes to the area data set by using IDCAMS ALTER ADDVOLUME.

**NOTE**
If an IDCAMS ALTER ADDVOLUME is used while the area data set is opened and allocated (online to IMS), the data set must be closed and reopened so that VSAM can recognize the added volume. Use the /DBR command to close the data set, and the /STA command to reopen the data set.

2 Before running the Extend function with the EXTEND_SDEP, BMC recommends that you execute the SDEP Scan/Delete utilities.

**NOTE**
If the SDEP storage area is in a logically wrapped condition when the extend is performed, the new CIs will appear to be in use (even though they are empty) until you execute the SDEP Scan/Delete utilities.

3 Run the Extend function with the EXTEND_SDEP keyword to specify the number of control intervals to add.

**Sample Control Statements for Extending SDEP**

Sample control statements for extending SDEP by using control intervals are shown in Figure 111.

**Figure 111  Sample control statements for extending SDEP by 100 control intervals**

```
EXTEND DBD=dbdname, IAREA=areaname,
    EXTEND_SDEP=(CONTROL_INTERVALS,100)
```

Sample control statements for extending SDEP by using cylinders are shown in Figure 112.

**Figure 112  Sample control statements for extending SDEP by 50 cylinders**

```
EXTEND DBD=dbdname, IAREA=areaname,
    EXTEND_SDEP=(CYLINDERS,50)
```
Sample output

The following example shows message BMC110000I that is generated when adding control intervals to the SDEP portion:

BMC110000I dbdname, areaname: AREA extension committed date, time

Subsequent maintenance on the DEDB

The next time that you perform maintenance on the DEDB, you might need to change your area data set VSAM cluster definition to allow for the extended size of the SDEP portion.

Analyzing the DEDB during an extend

If your site has a license for the Fast Path Online Analyzer/EP product, pointer validation will occur automatically (by default) when you execute an online extend.

This automatic process applies the default value QUICK for the POINTER_VALIDATION keyword. The analysis function performs a checksum validation of pointers for each area of the database that is specified on the EXTEND command. Automatic pointer validation provides assurance of the area’s pointer integrity, while providing statistics that show how the extend process affected space usage.

**NOTE**

You can override automatic quick pointer validation and request more detailed validation by specifying POINTER_VALIDATION=FULL on the EXTEND or GLOBAL command. Direct pointers are validated by using the cross-reference technique. Although not recommended, you can also specify another valid value for the POINTER_VALIDATION keyword (such as POINTER_VALIDATION=NONE) to completely disable the analysis function.

You can manually specify any valid value for the SDEP_VALIDATION keyword to control how SDEP pointers are to be validated.

You can also specify additional functions that are associated with the analysis function:

- Use the RAP_VALIDATION keyword to control how RAPs are processed.
- Use the LARGEST_DATABASE_RECORDS keyword to specify the number of largest database records to be tracked by the analysis process.
Making an image copy during an extend

You can request that one or more output image copy data sets be produced during an extend. Specify the IC subcommand for each image copy to be produced. For details about the IC subcommand, see Chapter 6, “Making an online image copy.”

When an image copy is requested during an online extend, a license for Fast Path Online Image Copy/EP is required.

The example in Figure 114 requests the creation of a single output image copy data set for the area being reorganized.

Sample EXTEND command scenarios

For scenarios that show how to use the online EXTEND command with key related keywords and subcommands, see Appendix C, “Sample online command scenarios.”
Diagnostic procedures and tools

This appendix contains information on preparing and gathering information before contacting BMC Customer Support. This appendix discusses the following topics:

Overview .......................................................................................................................... 167
  Note software product levels ......................................................................................... 168
  Locate generated SVC dump .......................................................................................... 168
  Define the general problem .......................................................................................... 170
  Provide duplication and resolution information .............................................................. 171
Using diagnostic commands ............................................................................................ 176
  Using the DUMP facility ............................................................................................... 176
Diagnosing problems with expressions ........................................................................... 177

Overview

To assist BMC Customer Support in providing a faster response to any problem, you should have certain information available when you call. Information about software product levels and problem descriptions are required to resolve the problem.

The procedures, questions, and information requests in this appendix were prepared by BMC Customer Support to help you receive timely and quality support. Your initial effort will help resolve issues involving BMC products. You might want to copy and fill out the forms in this appendix and send them to BMC for further discussion.
Note software product levels

Gather the following basic product level information:

- Fast Path Online Suite product version, release, and maintenance level
- IBM IMS version, release, and PUT level
- IBM z/OS version

Locate generated SVC dump

Each product in the Fast Path Online Suite will generate an SVC dump automatically when a product abend occurs. MVS will generate a notification message that the SVC dump has been created. You should locate the SVC dump that is produced by the product abend. The data set name for the SVC dump can be found in the JES messages in the JOBLOG. BMC might request that you supply the SVC dump with other requested documentation.

If the Fast Path product is unable to generate an SVC dump automatically following a product abend, message BMC110980E might be generated with return code 8. Table 15 lists explanations for reason codes that might be issued with the RC=8.

<table>
<thead>
<tr>
<th>Reason code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No SVC dump was requested.</td>
</tr>
<tr>
<td>1</td>
<td>An SVC dump was successfully started.</td>
</tr>
<tr>
<td>2</td>
<td>An SVC dump was suppressed because the capture phase of another SVC dump was in progress.</td>
</tr>
<tr>
<td>3</td>
<td>AN SVC dump was suppressed by a request by the installation (example: DUMP=NO at IPL or CHNGDUMP SET, NODUMP).</td>
</tr>
<tr>
<td>4</td>
<td>An SVC dump was suppressed by a SLIP NODUMP command.</td>
</tr>
<tr>
<td>5</td>
<td>An SVC dump was suppressed because a SYS1.DUMP data set was not available.</td>
</tr>
<tr>
<td>6</td>
<td>An SVC dump was suppressed because an I/O error occurred during the initialization of the SYS1.DUMP data set.</td>
</tr>
<tr>
<td>8</td>
<td>An SVC dump was suppressed because an SRB could not be scheduled to activate the dump tasks in the requested address spaces.</td>
</tr>
<tr>
<td>9</td>
<td>An SVC dump was suppressed because a terminating error occurred in SVC dump before the first dump record was written.</td>
</tr>
<tr>
<td>A</td>
<td>An SVC dump was suppressed because a status stop SRB condition was detected.</td>
</tr>
<tr>
<td>B</td>
<td>An SVC dump was suppressed by Dump Analysis and Elimination (DAE).</td>
</tr>
<tr>
<td>15</td>
<td>The parameter list address is zero.</td>
</tr>
<tr>
<td>16</td>
<td>The parameter list is not a valid SVC or SNAP parameter list.</td>
</tr>
<tr>
<td>Reason code</td>
<td>Explanation</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>17</td>
<td>The caller-supplied data set is not supported.</td>
</tr>
<tr>
<td>18</td>
<td>The start address is greater than or equal to the end address in a storage list.</td>
</tr>
<tr>
<td>19</td>
<td>The caller-supplied header is longer than 100 characters.</td>
</tr>
<tr>
<td>1A</td>
<td>The caller requested a 4K buffer, but did not reserve it.</td>
</tr>
<tr>
<td>1B</td>
<td>A storage list overlaps the 4K buffer.</td>
</tr>
<tr>
<td>1C</td>
<td>The caller-supplied DCB is not valid.</td>
</tr>
<tr>
<td>1E</td>
<td>An ASID in the ASID list is syntactically not valid.</td>
</tr>
<tr>
<td>22</td>
<td>The 4K buffer was requested with an SVC dump already in progress.</td>
</tr>
<tr>
<td>25</td>
<td>A subpool ID that was not valid was specified in the subpool list.</td>
</tr>
<tr>
<td>28</td>
<td>Part of the parameter list is inaccessible.</td>
</tr>
<tr>
<td>29</td>
<td>The caller-supplied DCB is inaccessible.</td>
</tr>
<tr>
<td>2A</td>
<td>The caller-supplied storage list is inaccessible.</td>
</tr>
<tr>
<td>2B</td>
<td>The caller-supplied header data is inaccessible.</td>
</tr>
<tr>
<td>2C</td>
<td>The caller-supplied ECB is inaccessible.</td>
</tr>
<tr>
<td>2D</td>
<td>The caller's ASID list is inaccessible.</td>
</tr>
<tr>
<td>2E</td>
<td>The caller's SUMLIST/SUMLSTA is inaccessible.</td>
</tr>
<tr>
<td>2F</td>
<td>The caller's SUBPLST is inaccessible.</td>
</tr>
<tr>
<td>30</td>
<td>The caller's KEYLIST is inaccessible.</td>
</tr>
<tr>
<td>31</td>
<td>Copies of the SLIP register and PSW are inaccessible.</td>
</tr>
<tr>
<td>32</td>
<td>The caller-supplied SRB is inaccessible.</td>
</tr>
<tr>
<td>33</td>
<td>The version number in the parameter list is not valid.</td>
</tr>
<tr>
<td>34</td>
<td>The caller's LISTD is inaccessible.</td>
</tr>
<tr>
<td>35</td>
<td>The caller's SUMLISTL is inaccessible.</td>
</tr>
<tr>
<td>36</td>
<td>The parameter list contains conflicting parameters.</td>
</tr>
<tr>
<td>37</td>
<td>The ID is longer than 50 characters.</td>
</tr>
<tr>
<td>38</td>
<td>The ID is not addressable.</td>
</tr>
<tr>
<td>39</td>
<td>The PSWREGS area is an incorrect length.</td>
</tr>
<tr>
<td>3A</td>
<td>The PSWREGS area is not addressable.</td>
</tr>
<tr>
<td>3B</td>
<td>The symptom record is not valid.</td>
</tr>
<tr>
<td>3C</td>
<td>The symptom record is not addressable.</td>
</tr>
<tr>
<td>3D</td>
<td>The DEB for the caller-supplied DCB is inaccessible.</td>
</tr>
<tr>
<td>3E</td>
<td>SVC dump is already using the maximum amount of virtual storage (as determined by the installation, using the MAXSPACE parameter on the CHNGDUMP command) to process other dumps.</td>
</tr>
<tr>
<td>3F</td>
<td>The caller-supplied STRLIST area is inaccessible.</td>
</tr>
<tr>
<td>40</td>
<td>The caller-supplied INTOKEN area is inaccessible.</td>
</tr>
<tr>
<td>41</td>
<td>The caller-supplied REMOTE area is inaccessible.</td>
</tr>
<tr>
<td>42</td>
<td>The caller-supplied PROBDESC area is inaccessible.</td>
</tr>
</tbody>
</table>
Define the general problem

By answering the following questions, you will begin to refine or isolate the problem you are having:

- What product messages or IMS messages were issued before and after the problem occurred (for example, Job Log, Control Region, and Dependent Region)?

- Are any system completion codes (ABEND) issued that explain why the system abnormally terminated the product?

- Are there circumstances that caused the problem? Circumstances which might be useful information needed to diagnose the problem include
  - starting and stopping a message region
  - peak system stress
  - other system component failure

- Has software maintenance been applied to any software components?

- Is the problem affecting one or more of the following entities?
  - particular system
  - DBD
  - area
  - PSB/transaction

- How often does the problem occur? Is the problem intermittent or continuous?

- If the problem is in another vendor’s product, and Fast Path Online Suite products are involved, are any other software vendors such as IBM working on the problem? If so, what is the ETR number or who is the other vendor contact?
BMC Customer Support might ask you to send documentation to reproduce a problem. You might be asked to provide the following items:

- job logs and SYSOUT
- SVC dump (generated automatically by Fast Path product)
- DBDLIB and DBD source members
- ACBLIB members
- DUMP command output
- image copies of databases
- monitor reports and job history statistics (CPU, elapsed time, EXCPs)

If a problem occurs, have the symptom dump available when you call BMC Customer Support. Save the SVC dump until the problem has been resolved to your satisfaction. Customer Support will generally ask for the PSW, PSW data, and general purpose registers at time of abend. Figure 115 shows an example of a symptom dump.

**Figure 115  Sample symptom dump**

```
************************************************************************
*                                                                      *
*                      BMC SOFTWARE ABEND SUMMARY                        *
*                                                                      *
*               FOR ASSISTANCE CALL BMC CUSTOMER SUPPORT                 *
*                                                                      *
*                 PLEASE HAVE THE FOLLOWING AVAILABLE                   *
*                   1. COMPLETE JES LOG FOR THIS JOB                    *
*                   2. PFPPRINT FOR ABENDING STEP                        *
*                   3. OUTPUT FROM DDNAME SYS00016                     *
*                                                                      *
*      JOB NAME: PFOXX070 JOB08392  STEP NAME: EXTRA01.PFPMIFP         *
*                                                                      *
*      S0C4 U0000    PSW 07841000 80000000  00000000 7F291F4E          *
*      AMODE31 ASC=P KEY=8 SUP CC=8 MASK=0 ILC=6 INTC=0004             *
*      MOD=PFPMAIN  CSECT=PFC00000  EP=PFC00000 OFFS=00000590           *
*      DATA 7F291F48  D5038008BC4DCA77400E05080                        *
*      VIRTUAL PAGE ADDRESS CAUSING EXCEPTION 00000000                 *
*      ASSEMBLED ON  06/29/12  14.43                                   *
*      LAST FIX APPLIED: ZPFP31                                       *
*      REGISTERS AT ENTRY TO ABEND:                                    *
*              0   00000000  7F4D7830  2E4042F0  A7F4FF66              *
*              4   7F4C0088  00000000  7F4D782C  FF28E9FE              *
*              8   A7F4FF66  7F59A650  2E4042A8  2E4041C0              *
*             C  7F29198B  7F4078C0  7F4078C0  7F29198B              *
*                                                                      *
************************************************************************
```
The ABEND Summary (dump shown in Figure 115) supports AMODE64 and other environments. In addition to the summary displaying in the JES log, a separate output file is dynamically allocated; the ABEND summary displays first in the file.

After the ABEND Summary, the Save Area Trace (Figure 116) displays. The trace provides the following information for each save area (starting with TCBFSA):

- module being entered
- module that made the call
- registry contents at entry
- contents of any allocated $AUTO memory

The dynamic output file (SYS00016) in Figure 116 provides the following items:

- duplicate of the ABEND Summary box
- snap dump of data areas
- a listing, in chronological order, of the registered save areas used in the task that abended

Figure 116  Sample Save Area Trace (part 1 of 4)
Figure 116 Sample Save Area Trace (part 2 of 4)

WEB (7F560470):

Save area trace:
CSECT ???????? (7F4D7B30) was entered at EP 7E4D7B30+00000000 (0001D92E) from module *SYSTEM*+00000000 (00D0BEE0)
Save area at 0000E0F0: Type=S0, Next=0000E0F1, Prev=00000000, Frame=00000000, StkX=00000000, StkP=00000000

CSECT ???????? (8001D92E) was entered at EP 7F63B658+00000000 (0001D92E) from module ????????+00000000 (00FDDBE0)
Save area at 0000E0F0: Type=S0, Next=0000E0F1, Prev=00000000, Frame=00000000, StkX=00000000, StkP=00000000

CSECT ???????? was entered at EP 7F63B658+00000000 (0001D92E) from module ????????+00000000 (00FDDBE0)

Appendix A Diagnostic procedures and tools
Figure 116  Sample Save Area Trace (part 3 of 4)
If BMC Customer Support requests that you supply a dump, provide the appropriate accompanying documentation. Use the BMC web site to submit the dump via FTP. Contact BMC Customer Support for instructions.

**NOTE**

Abbreviated dumps (like an Abend-Aid dump) do not provide the type of required information for problem resolution. Please ensure that a complete dump is taken.

The problem summary and documentation forms shown in Figure 117 and Figure 118 should be copied, filled out, and sent to BMC Customer Support for their understanding and resolution of your problem.

**Figure 117 Problem summary form (part 1 of 2)**

<p>| Fast Path Online Suite Product |</p>
<table>
<thead>
<tr>
<th>PROBLEM SUMMARY</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Issue Number:</strong> __________ <strong>Date:</strong> __________</td>
</tr>
<tr>
<td><strong>Company:</strong> ____________________</td>
</tr>
<tr>
<td><strong>Contact:</strong> ____________________</td>
</tr>
<tr>
<td><strong>Phone:</strong> ____________________</td>
</tr>
<tr>
<td><strong>Email:</strong> ____________________</td>
</tr>
</tbody>
</table>

--------------------------Software Product Levels --------------------------

<table>
<thead>
<tr>
<th>Product</th>
<th>Version.Release</th>
<th>Maintenance Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>_________</td>
<td>_________</td>
<td>_________</td>
</tr>
</tbody>
</table>

| IMS Version.Release.PUT Level: | _________ |
| MVS Version.Release: | _________ |

--------------------------Problem Description --------------------------

<table>
<thead>
<tr>
<th>Product Messages:</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC</td>
</tr>
<tr>
<td>BMC</td>
</tr>
<tr>
<td>BMC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other Messages (IMS, MVS, or JES):</th>
</tr>
</thead>
<tbody>
<tr>
<td>____________________</td>
</tr>
</tbody>
</table>

| System Completion Code= _____ | Reason Code= _____ |
Using diagnostic commands

You can use Fast Path Online Suite commands to help with diagnostics by using the DUMP facility.

Using the DUMP facility

The DUMP command requests a dump for diagnostic information when BMC Customer Support instructs you to provide a dump. You will generate a dump, based on the following DUMP_TYPE keyword values:

- ABEND – terminate the job, and generate a dump to the SYSUDUMP or SYSDUMP data set
Diagnosing problems with expressions

- SNAP – generate a snap dump of storage areas to the data set indicated in the PFPSNAP DD

- SVC – generate a system dump of the address space to the system dump data set

Table 16 provides information on the DUMP command set, its keywords, and parameters.

<table>
<thead>
<tr>
<th>Command</th>
<th>Keyword</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUMP</td>
<td>DUMP_TYPE</td>
<td>ABEND</td>
<td>specifies the type of dump to perform</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNAP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SVC</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
<td></td>
<td>ends command set</td>
</tr>
</tbody>
</table>

You can generate a SVC dump by command set like the following example:

```
DUMP DUMP_TYPE=SVC END
```

Diagnosing problems with expressions

The information in this section can be used to diagnose problems with the following product functions:

- expressions that are available in the Fast Path Online Analyzer/EP online data extract function

- expressions that are available in Fast Path Reorg/EP DEDB change, unload, and reload functions

The operators and operands that are used within a script or an expression are converted by the command language parser into a sequence of small operational steps. These operational steps are encoded into binary codes (called pseudo-code). This process is similar to the process that might be used by a programming language compiler to convert the source language into machine language. The generated pseudo-instructions are conceptually similar to actual machine instructions that might be executed by a CPU.

During execution of the product, the pseudo-code that is generated by the parser is executed (interpreted) as required. As the pseudo-code is executed, various conditions are tested to detect problems to ensure that invalid results are not generated. When one of these tests fails, an exception report is generated, and the command function terminates. The exception report is preceded by the following message:
Diagnosing problems with expressions

BMC110300C <dbdname> <areaname>: Exception code <code (name)> occurred during processing of <keyword> expression coded for the <command name> command on row <row number>, column <column number>.

The message contains the specific reason for the failure, identified by the exception code and its corresponding name. Table 17 lists the exceptions that might be produced. The most common exceptions encountered pertain to the problems with data values (Data, Overflow, Divide).

Table 17  Expression exception codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>Operation</td>
<td>invalid pseudo-instruction encountered</td>
</tr>
<tr>
<td>0005</td>
<td>Addressing</td>
<td>invalid data address encountered</td>
</tr>
<tr>
<td>0006</td>
<td>Specification</td>
<td>invalid data specification encountered</td>
</tr>
<tr>
<td>0007</td>
<td>Data</td>
<td>invalid data value encountered</td>
</tr>
<tr>
<td>0008</td>
<td>Overflow</td>
<td>value exceeds capacity of operand</td>
</tr>
<tr>
<td>0009</td>
<td>Fixed Divide</td>
<td>divide by zero (data-type F or X)</td>
</tr>
<tr>
<td>000B</td>
<td>Packed Divide</td>
<td>divide by zero (data-type P)</td>
</tr>
<tr>
<td>0010</td>
<td>Segment Trans</td>
<td>segment address not available</td>
</tr>
<tr>
<td>0012</td>
<td>Trans Specification</td>
<td>segment index invalid</td>
</tr>
<tr>
<td>0013</td>
<td>Structure Specification</td>
<td>structure index invalid</td>
</tr>
<tr>
<td>0015</td>
<td>Operand</td>
<td>run-time environment invalid</td>
</tr>
<tr>
<td>0019</td>
<td>Subscript</td>
<td>subscript out of range</td>
</tr>
<tr>
<td>0020</td>
<td>RBA Specification</td>
<td>invalid RBA value</td>
</tr>
<tr>
<td>0021</td>
<td>RBA Read</td>
<td>control interval read error</td>
</tr>
<tr>
<td>0022</td>
<td>RBA Write</td>
<td>control interval write error</td>
</tr>
<tr>
<td>0024</td>
<td>DMAC Lock</td>
<td>DMAC lock error</td>
</tr>
<tr>
<td>0028</td>
<td>Verification</td>
<td>VER function mismatch</td>
</tr>
<tr>
<td>0030</td>
<td>Stack full</td>
<td>register stack overflow</td>
</tr>
<tr>
<td>0031</td>
<td>Stack empty</td>
<td>register stack underflow</td>
</tr>
</tbody>
</table>

Message BMC110300C also identifies the failing script or expression by the keyword that is used and the position of the command containing that keyword. The remainder of the exception report provides detailed information needed for diagnosing the exact problem.

The contents of the exception report will vary, depending on the nature of the error. When possible, the product provides a specific diagnosis of the problem in a short summary report, such as the example shown in Figure 119.
This exception report identifies the pseudo-instruction that was being processed when the error was detected. In the preceding example, an error occurred at offset 006C while executing a LOAD instruction for the data field coded as CUSTROOT.26:4P (a 4-byte packed field starting in column 26 of the root segment). Apparently, a root segment has been read that does not contain a valid packed value in this field. This is followed by a snap dump showing the contents of the segment being processed and the concatenated key of the target segment. In this example, the target segment for the expression is CURRSEGME, even though the error occurred in a field from the root segment.

For other exceptions, the product produces a complete exception analysis, such as the example shown in Figure 120. In this example, a decimal divide exception has occurred during processing of the FIELDS keyword from some OUTPUT subcommand (not shown). This exception is caused when an attempt is made to divide by zero.

**Figure 119** Example exception report for common data exceptions

<table>
<thead>
<tr>
<th>BMC110300C DBFSAMD3, CUSDB: Exception Code 007 (data) occurred during processing of FIELDS expression coded for the OUTPUT command on row 5, column 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>The error occurred at offset 006C while executing pseudo-instruction:</td>
</tr>
<tr>
<td>006C FF00 FF080019 05000004 00000000 00000000 LOAD CUSTROOT.26:4P</td>
</tr>
</tbody>
</table>

********** Segment=CUSTROOT (at 7F5DD358) **********

| 000000 0030C2D9 F0F260E3 F0F1D1C1 D4C5E240 *..BR02.T01JAMES * |
| 000010 E3C1EBB3 D6D94040 40F0F040 40404040 *TAYLOR 00 * |
| 000020 40404040 40404040 40404040 40404040 * |

********** Concatenated key for target segment (CURRSEGME) **********

| 000000 C2D9F0F2 60E3F0F1 E4F1 *BR02.T01U1 * |

**Figure 120** Example of a complete exception report (part 1 of 2)

<table>
<thead>
<tr>
<th>BMC110300C DBFSAMD3, CUSDB: Exception Code 00B (decimal divide) occurred during processing of FIELDS expression coded for the OUTPUT command on row 5, column 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>The error occurred at offset 00A2; pseudo-code generated for expression:</td>
</tr>
<tr>
<td>0000 FF00 FF030002 02800002 00000000 00000000 LOAD 3:2C</td>
</tr>
<tr>
<td>0012 FFF8 FF050000 02800002 00000000 00000000 STORE (OUTPUT).1:2C</td>
</tr>
<tr>
<td>0024 FF00 FF030006 05000000 00000000 00000000 LOAD 7:11P</td>
</tr>
<tr>
<td>0036 FFF8 FF050000 02800002 00000000 00000000 STORE (OUTPUT).3:11P</td>
</tr>
<tr>
<td>0048 FF00 FF030011 05000000 00000000 00000000 LOAD 18:8P</td>
</tr>
<tr>
<td>005A FFF8 FF050000 05000000 00000000 00000000 STORE (OUTPUT).14:8P</td>
</tr>
<tr>
<td>006C FF00 FF030006 05000000 00000000 00000000 LOAD 7:11P</td>
</tr>
<tr>
<td>007E FFF8 FF030011 05000000 00000000 00000000 LOAD 18:8P</td>
</tr>
<tr>
<td>0090 FFD8 FF040000 05000000 00000000 00000000 CONV(-1) (WORK).1:16P</td>
</tr>
<tr>
<td>00A2 FF5D DIV</td>
</tr>
<tr>
<td>00A4 FFF8 FF050015 05000000 00000000 00000000 STORE (OUTPUT).22:8P</td>
</tr>
<tr>
<td>00B6 FFFF 0000 EXIT 0</td>
</tr>
</tbody>
</table>
Diagnosing problems with expressions

This exception report identifies the offset of the pseudo-instruction that was being processed when the error was detected, followed by a detailed listing of all pseudo-code that was generated for the script or expression in error. While it is not necessary to understand the specific details of the pseudo-code to diagnose the problem, being able to identify the operation involved with the error might provide all required information. The example indicates that the error occurred at offset 00A2. Looking at the pseudo-code listing, you find that a DIV instruction was being processed. This is preceded by LOAD instructions for the dividend (006C), the divisor (007E), and a CONV(-1) instruction that modifies the size of the dividend in preparation for the division. The divisor is the data field coded as 18:8P (an 8-byte packed field starting in column 18 of the default segment). Apparently, in one of the input segments selected this field contains a zero value.

The exception report lists the concatenated key of the current segment (CURRSEGM), with the I/O area for that segment. The report indicates that the 8-byte field that is used as the divisor (at offset X’0011’) in this segment has a zero value.

The exception report shows the current contents of the working storage used in script or expression evaluation (if any). It also shows the current contents of the output area being generated (if any); this would not be present for a WHERE expression because it does not produce an output area.
The exception report shows the current I/O area for all other database segments used in the expression. In the example, no database segments other than the target segment are referenced.

The exception report snaps the contents of the run-time expression environment block(#EXE). The content of this storage might be useful in diagnosing certain types of exceptions but is not needed in this example.
Appendix B Dynamic message modification

This appendix contains procedures on how to execute the PFPEPR00 utility program to dynamically modify eligible product messages. This appendix discusses the following topics:

Overview ........................................................... 183
Benefits ............................................................... 184
Customization capabilities ....................................... 184
Activation requirement ............................................ 185
Maintenance considerations ....................................... 185
Message modification subcommands ............................ 185
Dynamically modifying messages ................................. 186
Message severity levels .......................................... 187
Messages eligible for dynamic modification .................. 187
Changing severity level for a message ......................... 189
Suppressing repetitious messages .............................. 189
Listing message customizations ............................... 190
Restoring customized messages to product defaults ........ 190
Activating message customizations ............................ 191
Activating customizations by using JCL ...................... 191
Activating customizations by using dynamic allocation of the catalog on OPTIONS command ..................... 192
Activating dynamic allocation of customizations by using DFSMDA member 192

Overview

In addition to the analysis statistics storage and retrieval functions that are discussed in the *Fast Path Offline Suite User Guide*, the Fast Path/EP statistics repository facility also lets you define and retain customizations to *eligible* messages that might be issued by Fast Path Online Suite primary command processes. *Eligible* refers to messages that are available for customization as defined within the product. Customizations that you specify are stored in the same repository catalog that is associated with the statistics repository facility.
Benefits

You might find message customization to be useful for the following situations:

- to reduce the “nuisance factor” of a message, such as the number of times a message about segment length errors is issued for a very large area
- to reduce the severity level of a message from *E* (error) to *W* (warning) so that issuance of the message will not terminate processing (such as a pointer error message that would otherwise terminate an unload process)
- to increase the severity level of a condition-specific message to force an automatic snap dump to be taken in response to a detected condition within the DEDB or area

Customization capabilities

Using the PFPEPR00 utility, you can implement the following customization tasks for eligible messages that are issued by the Fast Path Online Suite products:

- override (change) the default suffix level (informational, warning, error or critical)
- suppress issuance of the specified message by all subsequent primary command processes after reaching a specified occurrence threshold
- restore any (or all) customizations to product defaults
- list all message customizations you already specified by a previous execution of PFPEPR00 (and that are stored in the repository catalog)

Any customizations you specify for messages by using the PFPEPR00 utility will apply to all Fast Path Online Suite primary command processes until you execute PFPEPR00 again to accomplish any of the following tasks:

- change previously specified message customizations
- add customizations for additional messages
- restore any (or all) customizations to product defaults

You can also specify message customizations on a temporary, per-job basis. For more information, see “Dynamically modifying messages” on page 38.
Activation requirement

When you have defined message customizations and have stored them in the repository catalog, you must define the repository catalog data set name in your job input to activate message customizations. For more information, see “Activating message customizations” on page 191.

Maintenance considerations

When you store message customizations in the repository and subsequently activate the repository by executing a Fast Path Online Suite primary command set, a statistics catalog record that contains area summary statistics is written to the repository catalog. At the minimum, a statistics catalog record is written to the repository catalog for each job that activates the message customizations which are stored in the repository.

Depending on the allocation rules you have set up in your repository, these statistics catalog records and more detailed statistics data sets will accrue in the repository over a period of time when you repeatedly use the message customization feature. You might need to perform periodic maintenance to delete these statistics records.

For information about deleting the statistics catalog entries that accrue in the repository catalog when you use message customizations, see the Statistics Repository Facility chapter in the *Fast Path Offline Suite User Guide*.

Message modification subcommands

The PFPEPR00 utility program offers the following subcommands that can be specified on the PROCESS_EPR command to dynamically modify eligible product messages:

- OVERRIDE
- RESET
- LIST

The functions of available message modification commands and subcommands are listed in Table 18.
Dynamically modifying messages

The OVERRIDE subcommand is useful when you want to change the severity (suffix) level for a selected message or messages, or override default suppression levels for selected messages. Table 19 lists the functions of the keywords that can be used with the OVERRIDE subcommand.

Table 19 Available keywords for the OVERRIDE subcommand

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>MESSAGE_LEVEL</td>
<td>change default severity level for message</td>
</tr>
<tr>
<td>MESSAGE_NUMBER</td>
<td>specify product ID number of message to be customized</td>
</tr>
<tr>
<td>MESSAGE_LIMIT</td>
<td>set threshold limit for suppression of message</td>
</tr>
</tbody>
</table>

The value PFP.PFPEPR is specified on the REPOSITORY_DSNAME keyword for all sample JCL shown in this appendix. PFP.PFPEPR is the name of the default repository catalog that is created automatically during installation of a Fast Path Online Suite product.

You must specify a separate OVERRIDE subcommand with appropriate keywords for each message that you want to customize. Multiple OVERRIDE subcommands might be specified under a single PROCESS_EPR command. In this manner, you can customize multiple messages with a single execution of the PFPEPR00 utility.
Message severity levels

Fast Path Online Suite products issue messages with a message number followed by a letter suffix to indicate the type or severity of the message. Table 20 lists the suffix types that can be specified (changed) for a dynamically modifiable message.

<table>
<thead>
<tr>
<th>Suffix</th>
<th>Condition code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>none</td>
<td>indicates an informational message</td>
</tr>
<tr>
<td>W</td>
<td>4</td>
<td>indicates a warning message</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A low-severity problem exists, and processing continues.</td>
</tr>
<tr>
<td>E</td>
<td>8</td>
<td>indicates an error message</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A problem exists, but processing continues.</td>
</tr>
<tr>
<td>C</td>
<td>12</td>
<td>indicates a critical problem exists that must be corrected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If possible, processing continues.</td>
</tr>
</tbody>
</table>

Messages eligible for dynamic modification

Table 21 lists all Fast Path Online Suite messages which can be customized by the user. In addition to setting an issuance threshold for these messages, you can also change the default suffix (severity level) code to any of the values that are listed in Table 20.

<table>
<thead>
<tr>
<th>Message number/ default suffix</th>
<th>Message text</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC110260E</td>
<td>Row rowNumber, column columnNumber: Randomizer module moduleName for database databaseName cannot be found in libraryName; root key retrieval will not be performed</td>
</tr>
<tr>
<td>BMC110262W</td>
<td>Index indexName: Specific Partition is not supported. PARTITION=ALL substituted</td>
</tr>
<tr>
<td>BMC111162C</td>
<td>dbdName, areaName, segmentRBA Segment length error - segment=segment length = value DMB specifies LENGTH=(max,min)</td>
</tr>
<tr>
<td>BMC111178W</td>
<td>dbdName: Change affecting all areas without IAREA=ALL</td>
</tr>
<tr>
<td>BMC111600I</td>
<td>dbdName: Count of segments selected (number) does not match count of segments written (number)</td>
</tr>
<tr>
<td>BMC111601I</td>
<td>dbdName: Count of Segments read (number) does not match count of segments written (number)</td>
</tr>
<tr>
<td>BMC111702W</td>
<td>dbdName,areaName,ddName: An error has occurred during the image copy function function; action received error code errorcode</td>
</tr>
</tbody>
</table>
### Table 21  Fast Path Online Suite messages available for dynamic modification (part 2 of 3)

<table>
<thead>
<tr>
<th>Message number/default suffix</th>
<th>Message text</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC112001C</td>
<td>database,areaName: Command commandName terminated by operator</td>
</tr>
<tr>
<td>BMC115501I</td>
<td>databaseName: Segment name is being added to the output database. No occurrences of this segment will be loaded during the restructure process.</td>
</tr>
<tr>
<td>BMC115503I</td>
<td>databaseName: Area name is being added to the output database. This area will be loaded during the restructure process.</td>
</tr>
<tr>
<td>BMC115540W</td>
<td>databaseName: The DELTA IMS interface was not detected. Carefully follow the steps defined in the manual before issuing the /STA DB command.</td>
</tr>
<tr>
<td>BMC115548W</td>
<td>databaseName: EARLY_TERMINATION was requested</td>
</tr>
<tr>
<td>BMC121104W</td>
<td>dbdName, areaName, rba: Block contains an FSE at offset=offset that is not cleared to zero</td>
</tr>
<tr>
<td>BMC121110W</td>
<td>dbdName, areaName, rba: Block contains a SCRAP at offset=offset that is not cleared to zero</td>
</tr>
<tr>
<td>BMC121116W</td>
<td>dbdName, areaName, rba: Block contains an improperly formatted SCRAP at offset=offset</td>
</tr>
<tr>
<td>BMC121122W</td>
<td>dbdName, areaName, rba: Unexpected end of file encountered</td>
</tr>
<tr>
<td>BMC121123W</td>
<td>dbdName, areaName, rba: RBA mismatch - DENDRRBA=address</td>
</tr>
<tr>
<td>BMC121124W</td>
<td>dbdName, areaName, rba: RDF mismatch - DENDRDF=value</td>
</tr>
<tr>
<td>BMC121125W</td>
<td>dbdName, areaName, rba: CIDF MISMATCH - DENDCIDF=value</td>
</tr>
<tr>
<td>BMC121126W</td>
<td>dbdName, areaName, rba: Block flag byte invalid - DBLKFLG1=value</td>
</tr>
<tr>
<td>BMC121127W</td>
<td>dbdName,areaName: Data block(s) exist beyond the logical end of file - DMACFBAD=block</td>
</tr>
<tr>
<td>BMC121128W</td>
<td>dbdName,areaName, rba: Invalid CI-type encountered in Reorg UOW.</td>
</tr>
<tr>
<td>BMC121135W</td>
<td>dbdName,areaName, rba: Unused bytes beyond space map are not zero</td>
</tr>
<tr>
<td>BMC121136W</td>
<td>dbdName,areaName, rba: Unused space map block entry at offset=offset should be zero</td>
</tr>
<tr>
<td>BMC121143W</td>
<td>dbdName, areaName, rba: IOVF block contains an invalid value for DBLKBTDW=value</td>
</tr>
<tr>
<td>BMC121145W</td>
<td>dbdName,areaName, rba: IOVF block is marked allocated but it is empty</td>
</tr>
<tr>
<td>BMC121150W</td>
<td>dbdName, areaName, rba: Unused SDEP block bytes beyond offset offset are not zero</td>
</tr>
<tr>
<td>BMC121151W</td>
<td>dbdName, areaName, rba: The contents of DBLKFSSOF in this SDEP block are inappropriate (X'xxx')</td>
</tr>
<tr>
<td>BMC121152W</td>
<td>dbdName, areaName, rba: Field DBLKFSSOF in this SDEP should be X'xxx'; it is incorrectly set to X'xxx'</td>
</tr>
<tr>
<td>BMC121153W</td>
<td>dbdName, areaName, rba: Encountered an SDEP block in IMS 5.1 (or earlier) format</td>
</tr>
<tr>
<td>BMC121161W</td>
<td>dbdName, areaName, rba: DOVF block has the &quot;look here&quot; flag set improperly</td>
</tr>
<tr>
<td>BMC121840C</td>
<td>dbdName,: Bad return code code from randomizer</td>
</tr>
</tbody>
</table>
### Changing severity level for a message

Figure 121 shows how the MESSAGE_NUMBER and MESSAGE_LEVEL keywords can be specified on the OVERRIDE subcommand to change the default severity level for message BMC111162 from E (the default) to W. This action will prevent the issuance of this message from terminating subsequent processes when an error is detected.

```plaintext
//PFP EXEC PGM=PFPEPR00,REGION=0M
//STEPLIB DD DSN=PFP.LOAD,DISP=SHR
//                 DSN=IMSVS.RESLIB,DISP=SHR
//PFPSYSIN DD   *
PROCESS_EPR REPOSITORY_DSNAME='PFP.PFPEPR'
OVERRIDE MESSAGE_NUMBER=111162,
   MESSAGE_LEVEL=WARNING
```

### Suppressing repetitious messages

Figure 122 shows how the MESSAGE_NUMBER and MESSAGE_LIMIT keyword can be specified on the OVERRIDE subcommand to set a threshold for a selected message. For all subsequent jobs, message BMC121150W is suppressed after it is issued 15 times by the process that is defined by a primary command.

```plaintext
//PFP EXEC PGM=PFPEPR00,REGION=0M
//STEPLIB DD DSN=PFP.LOAD,DISP=SHR
//                 DSN=IMSVS.RESLIB,DISP=SHR
//PFPSYSIN DD   *
PROCESS_EPR REPOSITORY_DSNAME='PFP.PFPEPR'
OVERRIDE MESSAGE_NUMBER=121150,
   MESSAGE_LIMIT=15
```

### Table 21  Fast Path Online Suite messages available for dynamic modification (part 3 of 3)

<table>
<thead>
<tr>
<th>Message number/default suffix</th>
<th>Message text</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC121841I</td>
<td><code>dbname.*: Root key requested key randomizes to area areaName (areaNumber) that is not within the input list</code></td>
</tr>
<tr>
<td>BMC121842I</td>
<td><code>dbname,areaName: Requested root key (key) not found</code></td>
</tr>
</tbody>
</table>
Listing message customizations

Figure 123 shows how the MESSAGE_OVERRIDE keyword can be specified on the LIST subcommand. The PFPEPR00 utility produces a report that lists all message customizations which were stored in the repository catalog by a previous execution of the utility.

<table>
<thead>
<tr>
<th>Figure 123</th>
<th>Listing all message customizations stored in repository catalog</th>
</tr>
</thead>
<tbody>
<tr>
<td>//PFP EXEC PGM=PFPEPR00,REGION=0M</td>
<td></td>
</tr>
<tr>
<td>//STEPLIB DD DSN=PFP.LOAD,DISP=SHR</td>
<td></td>
</tr>
<tr>
<td>// DD DSN=IMSVS.RESLIB,DISP=SHR</td>
<td></td>
</tr>
<tr>
<td>//PFPSYSIN DD *</td>
<td></td>
</tr>
<tr>
<td>PROCESS_EPR REPOSITORY_DSNAME='PFP.PFPEPR'</td>
<td></td>
</tr>
<tr>
<td>LIST MESSAGE_OVERRIDE</td>
<td></td>
</tr>
</tbody>
</table>

Restoring customized messages to product defaults

The MESSAGE_NUMBER keyword can be specified on the RESET subcommand to restore previous customizations to the product default for all or selected messages.

Figure 124 shows how to restore all previously specified customizations to product defaults by specifying a value of ALL on the MESSAGE_NUMBER keyword. The PFPEPR00 utility deletes all customization entries that were previously stored in the repository catalog, including severity level changes and suppression thresholds.

<table>
<thead>
<tr>
<th>Figure 124</th>
<th>Restoring all message customizations to product defaults</th>
</tr>
</thead>
<tbody>
<tr>
<td>//PFP EXEC PGM=PFPEPR00,REGION=0M</td>
<td></td>
</tr>
<tr>
<td>//STEPLIB DD DSN=PFP.LOAD,DISP=SHR</td>
<td></td>
</tr>
<tr>
<td>// DD DSN=IMSVS.RESLIB,DISP=SHR</td>
<td></td>
</tr>
<tr>
<td>//PFPSYSIN DD *</td>
<td></td>
</tr>
<tr>
<td>PROCESS_EPR REPOSITORY_DSNAME='PFP.PFPEPR'</td>
<td></td>
</tr>
<tr>
<td>RESET MESSAGE_NUMBER=ALL</td>
<td></td>
</tr>
</tbody>
</table>

To restore selected customized messages to product defaults while retaining previously specified customizations for other messages, you can specify a message number on the MESSAGE_NUMBER keyword. You must specify a separate RESET subcommand and associated MESSAGE_NUMBER keyword for each message that you want to restore to product defaults.
The example shown in Figure 125 restores prior customizations to two different messages. The first RESET subcommand restores the default severity (suffix) level for the message that was originally customized by the example shown in Figure 121. The second RESET subcommand will remove the suppression threshold for the message that was originally customized by the example shown in Figure 122.

Figure 125  Restoring selected message customizations to product defaults

//PFP EXEC PGM=PFPEPR00,REGION=OM
//STEPLIB DD DSN=PFP.LOAD,DISP=SHR
// DD DSN=IMSVS.RESLIB,DISP=SHR
//PFPSSYN DD *
PROCESS_EPR REPOSITORY_DSNAME='PFP.PFPEPR'
RESET MESSAGE_NUMBER=111162
RESET MESSAGE_NUMBER=121150

Activating message customizations

To activate message customizations when a Fast Path Online suite primary command is executed, you must specify the name of the repository catalog data set that contains the customizations that you specified by using the PFPEPR00 utility. The repository catalog data set name can be specified for your job input in one of the following ways:

- code the PFPEPR DD statement in the JCL
- code the REPOSITORY_DSNAME keyword on the OPTIONS command
- include the DFSMDA member for PFPEPR in the STEPLIB DD concatenation statements

**NOTE**
If the PFPEPR DD statement is not specified in the JCL, and the REPOSITORY_DSNAME keyword is not specified on the OPTIONS command, dynamic allocation of the repository catalog data set name will be performed by using the DFSMDA member. However, specifying the PFPEPR DD statement or the REPOSITORY_DSNAME overrides dynamic allocation of the DFSMDA member.

Activating customizations by using JCL

An example of coding the repository catalog data set name in the JCL with the PFPEPR DD name is shown in Figure 126.

Figure 126  Activating message customizations by using JCL (part 1 of 2)

//PFP EXEC PGM=PFPMAIN,REGION=OM
//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
Activating customizations by using dynamic allocation of the catalog on OPTIONS command

An example of coding the repository catalog data set name with the REPOSITORY_DSNAME keyword on the OPTIONS command is shown in Figure 127.

```
//         DD  DISP=SHR,DSN=IMS.RESLIB
//PFPEPR   DD  DSN=BMC.PFPEPR,DISP=SHR
//areaname DD DSN=area.dataset,DISP=SHR
//PFPSYSIN DD *
   ANALYZE DBD=dbname,IAREA=areaname
/*
```

**Figure 127** Activating message customizations by using REPOSITORY_DSNAME with OPTIONS command

**Activating dynamic allocation of customizations by using DFSMDA member**

To dynamically allocate the repository data set by using the dynamic allocation (DFSMDA) member for PFPEPR, use the DFSMDA member that is created during product installation. For more information, see the *Installation System User Guide* and the *Database Products for IMS Configuration Guide*, or use the sample JCL that is provided in the Fast Path/EP sample library, member #PFPMDA. Then, place the dynamic allocation member in one of your STEPLIB data sets.

An example of including the DFSMDA member in the STEPLIB DD statement is shown in Figure 128.

```
//         DD  DISP=SHR,DSN=IMS.RESLIB
//PFPEPR   DD  DSN=BMC.PFPEPR,DISP=SHR
//areaname DD DSN=area.dataset,DISP=SHR
//PFPSYSIN DD *
   OPTIONS
   REPOSITORY_DSNAME='BMC.PFPEPR'
//PFPSYSIN DD *
   ANALYZE DBD=dbname,IAREA=areaname
/*
```

**Figure 128** Activating dynamic allocation of customizations by using DFSMDA member (part 2 of 2)
The following items need to be considered when using the DFSMDA member to dynamically allocate the repository:

- Fast Path Analyzer/EP statistics will be added to the repository for each job that runs Fast Path Analyzer/EP. This includes jobs that are run for the reorganization, reload, and change functions.

- Additional repository maintenance should be scheduled, such as deleting entries, to manage the size of the repository catalog.
Activating dynamic allocation of customizations by using DFSMDA member
Appendix C Sample online command scenarios

This appendix contains sample command scenarios for the Fast Path Online Suite product functions. This appendix includes the following topics:

- How to interpret the scenarios
- JCL and control statement
- Descriptive text
- Sample library JCL
- Segment hierarchy for sample DEDB
- DBD for sample DEDB
- Online REORGANIZE command scenarios
  - Simultaneously reorganize, analyze, and create image copy
  - Disable automatic pointer analysis, place dependent segments, and stack image copies
  - Specify UOW selection criteria
  - Extend SDEP
- Online ANALYZE command scenarios
  - Analyze all areas and detect space thresholds
  - Request repository processing, analyze all areas, and store statistics in repository
  - Analyze all areas and create online image copies
  - Analyze a single area and request selected reports
- Online EXTRACT command scenarios
  - Extract and output only selected segments
  - Extract to multiple output files and control data selection
  - Extract all areas and generate a subset of area data
  - Modify segment layout during extract
  - Create a customized extract file format
- Online IMAGECOPY command scenarios
  - Simultaneously create online image copy and analyze pointers
  - Disable automatic pointer analysis and stack image copies
- Online EXTEND command scenarios
  - Simultaneously extend SDEP, analyze, and create image copy
  - Extend SDEP, disable automatic pointer analysis, and stack image copies
  - Extend IOVF
How to interpret the scenarios

The scenarios in this section show how to use each of the Fast Path Online Suite primary commands. Each scenario presents JCL and an associated control statement that combines a primary command with other elements of the Fast Path Online Suite command language to achieve a defined set of desired results. Each line of the JCL and control statement begins with a two-digit line number. Immediately following the JCL and control statement, a table provides textual descriptions of each line in the scenario. The following example shows how each scenario is presented and described.

JCL and control statement

**Figure 129  JCL and control statement**

```
01//PFP EXEC PGM=DFSRRCOO,REGION=OM.
02//       PARM=(IFP,PFPSAMP, DBF#FPU0)
03//STEPLIB DD DISP=SHR, DSN=BMC.PFP.LOAD
04// DD DISP=SHR, DSN=IMSVS.RESLIB
05//PFPSYSIN DD *
06  REORGANIZE POINTER_VALIDATION=FULL
07  REPORT REPORT_HEADING='ANALYSIS DURING REORG'
08  IC DSNAME='PFP.ICOPY.&DBD.&AREA(+1)',
09  UNIT=TAPE,DISP=(NEW,CATLG)
/*
```

**NOTE**

For all of the online scenarios, specifying the DBD name in the PFPSYSIN control card statement is not necessary because the DBD name is taken from the PARM statement on the EXEC statement.

Descriptive text

**Table 22  Descriptive text for JCL (part 1 of 2)**

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB DD statements for Fast Path online execution</td>
</tr>
<tr>
<td>05</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statements</td>
</tr>
<tr>
<td>06</td>
<td>All areas of the DBD are reorganized (IAREA=ALL is default). Full (cross-reference) pointer validation is performed on all areas.</td>
</tr>
</tbody>
</table>
Sample library JCL

The Fast Path/EP sample library contains sample JCL for each scenario that is presented in this appendix. Member $PFPIDX Contains a reference list of the scenario members.

Segment hierarchy for sample DEDB

The scenarios in this appendix are based on processes that are performed on the PFPSAMP sample DEDB, which contains three areas. Figure 130 shows a hierarchy diagram of the segments that are defined in the DBD for each area in the PFPSAMP database.

DBD for sample DEDB

Figure 131 provides the DBD that defines the PFPSAMP sample DEDB.

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>07</td>
<td>A report of the analysis is created with the specified heading.</td>
</tr>
<tr>
<td>08-09</td>
<td>An output image copy is created in the same job step as the reorganization. The DBD and area name are substituted in the image copy data set name to create a unique name for each area.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 22  Descriptive text for JCL (part 2 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The scenarios in this section show how to use the online REORGANIZE command with related keywords and subcommands.

```sql
Online REORGANIZE command scenarios

The scenarios in this section show how to use the online REORGANIZE command with related keywords and subcommands.
### Table 23  REORGANIZE scenarios

<table>
<thead>
<tr>
<th>Primary command/ scenario task</th>
<th>Subcommand/keyword</th>
<th>Concept/process</th>
<th>Page</th>
</tr>
</thead>
</table>
| REORGANIZE                    | POINTER_VALIDATION IC | ■ perform one-step, concurrent maintenance and analysis tasks  
                             |                   | ■ create image copy of reorganized DEDB                                    | 199  |
| Simultaneously Reorganize, Analyze, and Create Image Copy |                   | **REORGANIZE**                                                                 |      |
| REORGANIZE                    | SELECT_UOW        | ■ turn off (disable) default automatic pointer analysis                      | 200  |
| Disable Automatic Pointer Analysis | LOADCTL           | ■ specify all UOWs to ensure correct LOADCTL processing                      |      |
| Place Dependent Segments      | SEGMENT           | ■ specify data-conditional segment placement                                  |      |
| Stack Image Copies            | LOADCTL with WHERE INSERT_LIMIT_COUNT | ■ specify dependent segment placement based on segment count                  |      |
|                               | IC with STACK_NAME | ■ create stacked image copies of reorganized DEDB                             |      |
| REORGANIZE                    | SELECT_UOW        | ■ limit reorganization of a specific area to selected UOWs only              | 201  |
| Specify UOW Selection Criteria | IOVF_SAVE_THRESHOLD= value | ■ specify different UOW selection criteria for different area              |      |
|                               | IOVF_SAVE_THRESHOLD= percentage | ■ limit IOVF UOWs based on count or percentage                                |      |
| REORGANIZE                    | POINTER_VALIDATION IC | ■ request quick (checksum) pointer analysis                                  | 202  |
| Extend SDEP                   | IAREA EXTEND_SDEP_#CIS | ■ limit reorganization and associated tasks to one area only                 |      |
|                               | IC                | ■ extend SDEP during reorganization                                          |      |
|                               |                   | ■ create output image copy of extended area                                  |      |

### Simultaneously reorganize, analyze, and create image copy

**Figure 132  JCL to simultaneously reorganize, analyze, and create image copy (part 1 of 2)**

```
01 //PFP EXEC PGM=DFSRRC00,REGION=0M,  
02     PARM=(IFP,PFPSAMP,DBF#FPU0)  
03 //STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD  
04 // DD DISP=SHR,DSN=IMSVS.RESLIB  
05 //PFPSYSIN DD *  
06 REORGANIZE POINTER_VALIDATION=FULL  
07 IC DSNNAME='PFP.ICOPY.&DBD.&AREA(+1)'.
```
Figure 132  JCL to simultaneously reorganize, analyze, and create image copy (part 2 of 2)

```
08  UNIT=TAPE,DISP=(NEW,CATLG)
/*
```

Table 24  Descriptive text for JCL to simultaneously reorganize, analyze, and create image copy

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB DD statements for Fast Path online execution</td>
</tr>
<tr>
<td>05</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statements</td>
</tr>
<tr>
<td>06</td>
<td>A complete analysis (full pointer validation) of all areas is performed on the DEDB.</td>
</tr>
<tr>
<td>07-08</td>
<td>An output image copy is created in the same job step as the reorganization. The DBD and area name are substituted in the image copy data set name to create a unique name for each area.</td>
</tr>
</tbody>
</table>

**Disable automatic pointer analysis, place dependent segments, and stack image copies**

Figure 133  JCL to disable automatic pointer analysis, place dependent segments, and stack image copies

```
01//PFP EXEC PGM=DFSRRC00,REGION=0M,
02// PARM=(IFP,PFPSAMP,DBF#FPU0)
03//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
04// DD DISP=SHR,DSN=IMSVS.RESLIB
05// DD DISP=SHR,DSN=BMC.DPK.LOAD
06//$$DPICDS DD DISP=SHR,DSN=PFP.DPK.DPICDS
07//$$DPITBL DD DISP=SHR,DSN=PFP.DPK.DPITBL
08//PFPSYSIN DD *
09 REORGANIZE IAREA=ALL,
10  POINTER_VALIDATION=NONE,
11   SELECT_UOW=ALL
12 LOADCTL SEGMENT=(SEGB,DEPENDENTS),
13   LOCATION=IOVF,WHERE=(FLD05 EQ C'INAC')
14 LOADCTL SEGMENT=SEGF,LOCATION=IOVF,
15   INSERT_LIMIT_COUNT=1
16 LOADCTL SEGMENT=SEGG,LOCATION=DOVF
17 IC STACK_NAME=NAME1,
18 UNIT=TAPE,DISP=(NEW,CATLG),
19   DSNAME='PFP.ICOPY.&DBD.&AREA(+1)'
/*
```
Specify UOW selection criteria

Table 25  Descriptive text for JCL to disable automatic pointer analysis, place dependent segments, and stack image copies

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB DD statements for Fast Path online execution</td>
</tr>
<tr>
<td>05-07</td>
<td>DD statements that define data sets for the BMC DATA PACKER/IMS product. These data sets are required because the WHERE statement specifies a field in the compressed portion of the segment.</td>
</tr>
<tr>
<td>08</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statements</td>
</tr>
<tr>
<td>09</td>
<td>All areas of the DEDB are reorganized.</td>
</tr>
<tr>
<td>10</td>
<td>Pointer validation and physical validation are not performed. POINTER_VALIDATION=NONE overrides the default checksum validation.</td>
</tr>
<tr>
<td>11</td>
<td>All UOWs must be processed so that LOADCTL processing executes correctly.</td>
</tr>
<tr>
<td>12-13</td>
<td>This is a conditional LOADCTL statement. The dependent segments (SEGC and SEGD) of SEGB are placed in IOVF if SEGB is marked inactive (FLD05 EQ C'INAC'). This placement leaves RAA and DOVF storage available for more frequently accessed segments.</td>
</tr>
<tr>
<td>14-15</td>
<td>For SEGF, after the first occurrence under a parent, the remaining segments are placed into IOVF.</td>
</tr>
<tr>
<td>16</td>
<td>All SEG occurrences are placed into DOVF as long as DOVF storage is available in the UOW. When DOVF becomes full, any remaining SEG segments for the UOW are written to IOVF.</td>
</tr>
<tr>
<td>17-19</td>
<td>An image copy data set is created for each area in the database. These data sets are stacked onto tape.</td>
</tr>
<tr>
<td>18</td>
<td>All image copy data sets that are part of the same stack group are written in the order processed onto the same tape volume (or volumes) as file number 1, 2, and 3.</td>
</tr>
<tr>
<td>19</td>
<td>Dynamic allocation is always used for stacked image copy data sets. The DBD and area name are substituted in the image copy data set name to create a unique name for each area.</td>
</tr>
</tbody>
</table>

Specify UOW selection criteria

Only two of the areas in the PFPSAMP database are reorganized.

Figure 134  JCL to specify UOW selection criteria

```
01//PFP EXEC PGM=DFSRRC00,REGION=0M,
02// PARM=(IFP,PFPSAMP,DBF#FPU0)
03//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
04// DD DISP=SHR,DSN=IMSVS.RESLIB
05//PFPSYSIN DD *
06 REORGANIZE IAREA=PFPSAMP1, SELECT_UOW=IOVF,IOVF_SAVE_THRESHOLD=10
07 REORGANIZE IAREA=PFPSAMP1, SELECT_UOW=(IOVF,150),
08 REORGANIZE IAREA=PFPSAMP2, SELECT_UOW=(IOVF,150),
10 IOVF_SAVE_THRESHOLD=20%
/*
```
Before executing this control statement, you must ensure that adequate space is present on the primary volume of the target area data set to accommodate the request for additional SDEP CIs. If the necessary space is not present, allocate volumes to the area data set by using IDCAMS ALTER ADDVOLUME command.

**NOTE**

Be sure to revise the VSAM cluster definition to allow for the extended size of the SDEP portion.

### Extend SDEP

Table 26  Descriptive text for JCL to specify UOW selection criteria

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB DD statements for Fast Path online execution</td>
</tr>
<tr>
<td>05</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statements</td>
</tr>
<tr>
<td>06</td>
<td>This REORGANIZE command is restricted to one area in order to apply UOW selection criteria for area PFPSAMP1.</td>
</tr>
<tr>
<td>07</td>
<td>All UOWs in area PFPSAMP1 with IOVF CIs used are selected for the reorganization. A UOW must have a minimum of 10 IOVF CIs allocated to it to be selected for reorganization, and the reorganization must reclaim a minimum of 10 IOVF to be reorganized. Otherwise, the UOW is skipped.</td>
</tr>
<tr>
<td>08</td>
<td>This REORGANIZE command is restricted to one area in order to apply a different UOW selection criteria for area PFPSAMP2.</td>
</tr>
<tr>
<td>09</td>
<td>Only UOWs in area PFPSAMP2 with at least 150 IOVF CIs used are selected for the reorganization. Otherwise, the UOW is skipped.</td>
</tr>
<tr>
<td>10</td>
<td>A minimum percentage of IOVF for a UOW (20% of 150) must be reclaimed by the reorganization. Otherwise, the UOW is skipped.</td>
</tr>
</tbody>
</table>

Figure 135  JCL to extend SDEP

```plaintext
01//PFP EXEC PGM=DFSRRCO0,REGION=OM,
02//       PARM=(IFP,PFPSAMP,DFBF#FPU0)
03//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
04//       DD DISP=SHR,DSN=IMSVS.RESLIB
05//PFPSYSIN DD *
06 REORGANIZE IAREA=PFPSAMP1.
07       POINTER_VALIDATION=QUICK,
08 EXTEND_SDEP_#CIS=1000
09 IC DSNNAME='PFP.ICOPY.&DBD.&AREA(+1)',
10 UNIT=TAPE,DISP=(NEW,CATLG)
/*
```
Online ANALYZE command scenarios

The scenarios in this section show how to use the online ANALYZE command with related keywords and subcommands.

Table 27  Descriptive text for JCL to extend SDEP

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB DD statements for Fast Path online execution</td>
</tr>
<tr>
<td>05</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statements</td>
</tr>
<tr>
<td>06</td>
<td>Area PFPSAMP1 is identified on the REORGANIZE statement.</td>
</tr>
<tr>
<td>07</td>
<td>If you have a license for the BMC Fast Path Online Analyzer/EP product, quick (checksum) pointer validation is performed on area PFPSAMP1 by default, even if you do not specify it explicitly.</td>
</tr>
<tr>
<td>08</td>
<td>A minimum of 1000 CIs are added to the SDEP portion of area PFPSAMP1. The requested number of CIs might be rounded up to the next control area (CA) boundary.</td>
</tr>
<tr>
<td>09-10</td>
<td>During the area extend, an output image copy is taken simultaneously with standard storage and disposition parameters. The DBD and area name are substituted in the image copy data set name to create a unique name for each area.</td>
</tr>
</tbody>
</table>

Table 28  ANALYZE scenarios (part 1 of 2)

<table>
<thead>
<tr>
<th>Primary Command/ Scenario Task</th>
<th>Subcommand/Keyword</th>
<th>Concept/Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANALYZE</td>
<td>POINTER_VALIDATION</td>
<td></td>
</tr>
<tr>
<td>Analyze All Areas</td>
<td>THRESHOLD</td>
<td>request full pointer analysis</td>
</tr>
<tr>
<td>Detect Space Thresholds</td>
<td>DOVF_FREESPACE_PERCENT</td>
<td>detect various freespace/overflow thresholds</td>
</tr>
<tr>
<td></td>
<td>IOVF_FREESPACE_PERCENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RAA_FREESPACE_PERCENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RAP_OVERFLOW_PERCENT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FREESPACE_RAA_DOVF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FREESPACE_RAA_IOVF</td>
<td></td>
</tr>
<tr>
<td>ANALYZE</td>
<td>OPTIONS command with following keywords:</td>
<td></td>
</tr>
<tr>
<td>Request Repository Processing</td>
<td>REPOSITORY_DSNAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>REPOSITORY_GROUP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>REPOSITORY_RETENTION_COUNT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>REPOSITORY_OVERWRITE</td>
<td></td>
</tr>
<tr>
<td>Analyze All Areas</td>
<td>Store Statistics in Repository</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ request quick (checksum) pointer analysis</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ activate repository processing by using OPTIONS command</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ specify name of repository and repository group</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ request overwriting of existing statistics data set</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ specify maximum number of repository statistics catalog entries that remain recorded within the repository catalog</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ store analysis statistics gathered during analysis process in repository</td>
<td></td>
</tr>
</tbody>
</table>
Table 28  ANALYZE scenarios (part 2 of 2)

<table>
<thead>
<tr>
<th>Primary Command/ Scenario Task</th>
<th>Subcommand/Keyword</th>
<th>Concept/Process</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANALYZE</td>
<td>POINTER_VALIDATION IC</td>
<td>request full pointer analysis</td>
<td>206</td>
</tr>
<tr>
<td>Analyze All Areas</td>
<td></td>
<td>create concurrent online image copy for each area</td>
<td></td>
</tr>
<tr>
<td>Create Online Image Copies</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANALYZE</td>
<td>REPORT</td>
<td>disable default generation of all reports</td>
<td>207</td>
</tr>
<tr>
<td>Analyze Single Area</td>
<td>REPORT_DEFAULT</td>
<td>request generation of selected reports only (override default)</td>
<td></td>
</tr>
<tr>
<td>Request Selected Reports</td>
<td>FREESPACE_ANALYSIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IOVF_SPACE_ANALYSIS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Analyze all areas and detect space thresholds

Figure 136  JCL to analyze all areas and detect space thresholds

```jcl
01//PFP EXEC PGM=DFSRRC00,REGION=0M,
02//   PARM=(IFP,PFPSAMP,DBF#FPUO)
03//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
04//   DD DISP=SHR,DSN=IMSVS.RESLIB
05//PFPSYSIN DD *
06 ANALYZE POINTER_VALIDATION=FULL
07   THRESHOLD
08   DOVF_FREESPACE_PERCENT=10,
09   IOVF_FREESPACE_PERCENT=15,
10   RAA_FREESPACE_PERCENT=20,
11   RAP_OVERFLOW_PERCENT=15,
12   FREESPACE_DOVF_IOVF=(15,50),
13   FREESPACE_RAA_DOVF=(15,25),
14   FREESPACE_RAA_IOVF=(15,50)
/*
```

Table 29  Descriptive text for JCL to analyze all areas and detect space thresholds (part 1 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB DD statements for Fast Path online execution</td>
</tr>
<tr>
<td>05</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statements</td>
</tr>
<tr>
<td>06</td>
<td>Analysis is performed for all areas (IAREA=ALL is default). Full (cross-reference) pointer validation is performed on all areas in the DEDB.</td>
</tr>
<tr>
<td>07</td>
<td>The THRESHOLD subcommand and the keywords that follow it help to determine whether there are any potential problems because of free space limitations.</td>
</tr>
<tr>
<td>08</td>
<td>If the analysis process detects less than 10% of free space available in DOVF in the area, a warning message is issued.</td>
</tr>
</tbody>
</table>
Request repository processing, analyze all areas, and store statistics in repository

Statistics from the analysis process are stored in the repository. The REPOSITORY_DSNAME keyword identifies the repository catalog to be used. This job also overwrites any previously existing statistics data set by the same name. An allocation rule previously has been set up in the repository to create statistics data sets by using the data set name mask ‘PFP.&IMSID.&DBD.&AREA’. For more information, see the Statistics Repository Facility in the *Fast Path Offline Suite User Guide*.

### Table 29 Descriptive text for JCL to analyze all areas and detect space thresholds (part 2 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>09</td>
<td>If the analysis process detects less than 15% of free space available in IOVF in the area, a warning message is issued.</td>
</tr>
<tr>
<td>10</td>
<td>If the analysis process detects less than 20% of free space available in RAA in the area, a warning message is issued.</td>
</tr>
<tr>
<td>11</td>
<td>If the analysis process detects that the percentage of RAP blocks that overflow into DOVF/IOVF exceeds 15%, a warning message is issued.</td>
</tr>
<tr>
<td>12</td>
<td>If the analysis process detects that the computed DOVF free space is greater than 15% and the computed IOVF free space is less than 50%, a warning message is issued.</td>
</tr>
<tr>
<td>13</td>
<td>If the analysis process detects that the computed RAA free space is greater than 15% and the computed DOVF free space is less than 25%, a warning message is issued.</td>
</tr>
<tr>
<td>14</td>
<td>If the analysis process detects that the computed RAA free space is greater than 15% and the computed IOVF free space is less than 50%, a warning message is issued.</td>
</tr>
</tbody>
</table>

### Figure 137 JCL to request repository processing, analyze all areas, and store statistics in repository

```
01//PFP EXEC   PGM=DFSRRCOO,REGION=0M,
02//          PARM=(IFP,PFPSAMP,DBF#FPUO)
03//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
04//          DD DISP=SHR,DSN=IMSVS.RESLIB
05//PFPOPTS DD *
06  OPTIONS
07  REPOSITORY_DSNAME='BMC.PFP.PFPEPR',
08  REPOSITORY_GROUP=TEST,
09  REPOSITORY_RETENTION_COUNT=5,
10  REPOSITORY_OVERWRITE=YES
11/*
12//PFPSYSIN DD *
13 ANALYZE IAREA=ALL,POINTER_VALIDATION=QUICK
/*
```
**Analyze all areas and create online image copies**

Table 30  Descriptive text for JCL to request repository processing, analyze all areas, and store statistics in repository

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB DD statements for Fast Path/EP online execution.</td>
</tr>
<tr>
<td>05</td>
<td>PFPOPTS DD for Fast Path/EP OPTIONS statement (line 10).</td>
</tr>
<tr>
<td>06</td>
<td>OPTIONS command is required to specify subsequent repository keywords.</td>
</tr>
<tr>
<td>07</td>
<td>Defines the repository to be used.</td>
</tr>
<tr>
<td>08</td>
<td>Identifies the group from which the area’s allocation rules should be derived.</td>
</tr>
<tr>
<td>09</td>
<td>Specifies that up to five statistics data sets for these database areas are stored in the repository catalog at any one time.</td>
</tr>
<tr>
<td>10</td>
<td>Indicates that if the statistics data set name defined for these areas exists, the existing statistics will be overwritten with the new statistics data set created by this job.</td>
</tr>
<tr>
<td>11</td>
<td>End of OPTIONS control card input.</td>
</tr>
<tr>
<td>12</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statement</td>
</tr>
<tr>
<td>13</td>
<td>All areas of the database are analyzed. Quick (checksum) pointer validation is performed on all areas in the DEDB. This pointer validation creates the statistics for database PFPsamp to be stored in the repository that is identified in line 07.</td>
</tr>
</tbody>
</table>

**Analyze all areas and create online image copies**

Figure 138  JCL to analyze all areas and create online image copies

```plaintext
01//PFP EXEC  PGM=DFSRRC00,REGION=0M,
02//     PARM=(IFP,PFPsamp,DBF#FPUO)
03//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
04//     DD DISP=SHR,DSN=IMSVS.RESLIB
05//PFPSYSIN DD *
06 ANALYZE IAREA=ALL, |
07     POINTER_VALIDATION=FULL, |
08 IC DSNAME='PFP.ICOPY.&DBD.&AREA(+1)', |
09     UNIT=TAPE,DISP=(NEW,CATLG) */
```

Table 31  Descriptive text for JCL to analyze all areas and create online image copies (part 1 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB DD statements for Fast Path online execution</td>
</tr>
<tr>
<td>05</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statements</td>
</tr>
<tr>
<td>06</td>
<td>Analysis is performed on all areas.</td>
</tr>
</tbody>
</table>
Analyze a single area and request selected reports

Table 31  Descriptive text for JCL to analyze all areas and create online image copies (part 2 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>07</td>
<td>Full (cross-reference) pointer validation is performed on all areas in the DEDB.</td>
</tr>
<tr>
<td>08-09</td>
<td>An output image copy for each area is created in the same job step as the analysis. The DBD and area name are substituted in the image copy data set name to create a unique name for each area.</td>
</tr>
</tbody>
</table>

Analyze a single area and request selected reports

Figure 139  JCL to analyze a single area and request selected reports

```jcl
01//PFP EXEC PGM=DFSRRC00,REGION=OM,
02//     PARM=(IFP,PFPSAMP,DBF#FPUO)
03//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
04//     DD DISP=SHR,DSN=IMSVS.RESLIB
05//PFPSYSIN DD *
08 ANALYZE IAREA=PFPSAMP1
09     REPORT REPORT_DEFAULT=NO,
10     FREESPACE_ANALYSIS=YES,
11     IOVF_SPACE_ANALYSIS=YES
/*
```

Table 32  Descriptive text for JCL to analyze a single area and request selected reports

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB DD statements for Fast Path/EP online execution.</td>
</tr>
<tr>
<td>05</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statements</td>
</tr>
<tr>
<td>06</td>
<td>Only the PFPSAMP1 area is analyzed. Quick (checksum) pointer validation is performed on area PFPSAMP1 by default.</td>
</tr>
<tr>
<td>07</td>
<td>Sets report generation default to NO (no reports are generated unless specified by name).</td>
</tr>
<tr>
<td>08</td>
<td>The Free Space Analysis Report is generated.</td>
</tr>
<tr>
<td>09</td>
<td>The IOVF Space Analysis Report is generated.</td>
</tr>
</tbody>
</table>

Online EXTRACT command scenarios

The scenarios in this section show how to use the online EXTRACT command with keywords and subcommands. Collectively, these examples demonstrate all options that are available for customizing the output of a data extract. To exemplify segment selection/exclusion, the scenarios in this section use the segment hierarchy shown on page 197.
### Table 33  EXTRACT scenarios (part 1 of 2)

<table>
<thead>
<tr>
<th>Primary Command/ Scenario Task</th>
<th>Subcommand/Keyword</th>
<th>Concept/Process</th>
<th>Page</th>
</tr>
</thead>
</table>
| Extract and Output Only Selected Segments | EXTRACT_FORMAT= EXTRACT EXPAND EXCLUDE SEGMENT EXCLUDE SEGMENT with WHERE OFILECTL with OAREA and DDNAME OUTPUT SEGMENT OUTPUT SEGMENT with FIELDS | - create output file in default EXTRACT format  
- expand compressed data  
- exclude selected segments  
- conditionally exclude selected segments  
- define segments to be written to output file  
- specify fields to be extracted from segment identified for output  
- manipulate output fields in specified segments  
- specify output file name and attributes                                                                                                                                                                                                                   | 209  |
| Extract to Multiple Output Files | EXTRACT_FMT=HDUNLOAD INCLUDE SEGMENT INCLUDE SEGMENT with WHERE EXCLUDE SEGMENT with WHERE OFILECTL with OAREA and DSNAME OFILECTL with OAREA and DDNAME | - create output file in HDUNLOAD format  
- include selected segment  
- conditionally include selected segment  
- specify relationship between excluded data and output data  
- specify first output file name and attributes using data set mask  
- specify second output file name and attributes from provided DD statement                                                                                                                                                                                                 | 210  |
| Extract All Areas | EXTRACT_FORMAT= HDUNLOAD OUTPUT_DSN_MASK EXPAND INCLUDE SEGMENT with SAMPLE_INTERVAL and SAMPLE_LIMIT | - create output file in HDUNLOAD format  
- dynamically allocate output files  
- expand compressed data  
- include sampling of root segments based on root segment count                                                                                                                                                                                                                                      | 212  |
Table 33  EXTRACT scenarios (part 2 of 2)

<table>
<thead>
<tr>
<th>Primary Command/ Scenario Task</th>
<th>Subcommand/Keyword</th>
<th>Concept/Process</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTRACT Modify Segment Layout during Extract</td>
<td>INPUT_DSN_MASK EXTRACT_FORMAT= EXTRACT SORT EXPAND OFILECTL OUTPUT SEGMENT with FIELDS</td>
<td>create output file in default EXTRACT format sort output data expand compressed data define and dynamically allocate output file manipulate output fields in specified segments</td>
<td>213</td>
</tr>
<tr>
<td>EXTRACT Create a Customized Extract File Format</td>
<td>EXTRACT_FORMAT=USER USER_RECORD with BREAK / BEFORE USER_RECORD with BREAK / AFTER FIELDS SEGMENT_RECORD_PREFIX</td>
<td>create customized format for output file define multiple header records define trailer record specify prefix for segment record</td>
<td>214</td>
</tr>
</tbody>
</table>

Extract and output only selected segments

This scenario extracts and writes only two segment types to the output extract file.

Figure 140  JCL to extract and output only selected segments

```
01//PFPMAIN EXEC PGM=DFSRRC00,REGION=0M,
02//       PARM=(IFP,PFPAMP,DFB#FPU0)
03//STEPLIB DD DSN=BMC.PFP.LOAD,DISP=SHR
04//       DD DSN=IMSVS.RESLIB,DISP=SHR
05//       DD DISP=SHR,DSN=BMC.DPK.LOAD
06//$$DPICDS DD DISP=SHR,DSN=PFP.DPK.DPICDS
07//$$DPITBL DD DISP=SHR,DSN=PFP.DPK.DPITBL
08//EXTFILE DD DSN=BMC.PFP.EXTFILE,
09//       DISP=(,CATLG,DELETE),
10//       UNIT=DISK,
11//       SPACE=(CYL,(200,8),RLSE)
12//PFPSSIN DD *
13 EXTRACT IAREA=ALL,EXTRACT_FORMAT=EXTRACT,EXPAND=YES
14 EXCLUDE SEGMENT=SDEP
15 EXCLUDE SEGMENT=SEGE
16 EXCLUDE SEGMENT=SEGG
17 EXCLUDE SEGMENT=SEGB,
18 WHERE=(SEGA.SEGAKEY LT C'19980101')
19 OFILECTL OAREA=ALL,DDNAME=EXTFILE
20 OUTPUT SEGMENT=SEGA
21 OUTPUT SEGMENT=SEGB,
22 FIELDS=(SEGBKEY,FLD05,20:8,30:25)
/*
```
Extract to multiple output files and control data selection

All areas of database PFPSAMP are extracted into HD Unload format as specified by the EXTRACT_FORMAT=HDUNLOAD keyword. When HD Unload format is used, correct hierarchical files are written. The extracted output from the three areas is written to two files by using two OFILECTL subcommands.

Table 34 Descriptive text for JCL to extract and output only selected segments

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB statements for Fast Path online execution</td>
</tr>
<tr>
<td>02</td>
<td>DBD name is specified as second parameter of PARM statement, which is required for an IFP execution.</td>
</tr>
<tr>
<td>05-07</td>
<td>DD statements that define data sets for the BMC DATA PACKER/IMS product. These data sets are required because the EXPAND=YES keyword requested data expansion, and because of data manipulation for compressed fields.</td>
</tr>
<tr>
<td>08-11</td>
<td>The extracted output for all areas is written to the DD name EXTFILE.</td>
</tr>
<tr>
<td>12</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statements</td>
</tr>
<tr>
<td>13</td>
<td>The EXTRACT command specifies all areas to be processed. Specifying a DBD name is optional, because it has been specified as an IFP parameter. Data is extracted in the standard default format (EXTRACT_FORMAT=EXTRACT). The format of the records is the layout is specified in the <em>Fast Path/EP Series Reference Manual</em>. All compressed segment data is to be expanded.</td>
</tr>
<tr>
<td>14</td>
<td>Exclude segment SDEP from selection.</td>
</tr>
<tr>
<td>15</td>
<td>Exclude segment SEGE from selection, which also excludes segment SEGF.</td>
</tr>
<tr>
<td>16</td>
<td>Exclude segment SEGG from selection.</td>
</tr>
<tr>
<td>17-18</td>
<td>SEGB segments are excluded from selection if the SEGAKEY field for parent segment SEGA contains a date value less than ‘19980101’.</td>
</tr>
<tr>
<td>19</td>
<td>All areas are written to a single output file using the data set specified on the EXTFILE DD statement.</td>
</tr>
<tr>
<td>20</td>
<td>The resulting output file contains all SEGA segments.</td>
</tr>
<tr>
<td>21-22</td>
<td>The resulting output file contains only SEGB segments that were not excluded by the EXCLUDE statement. These segments are output with the following format (all offsets are from the input image): the SEGBKEY field starting at offset 3, for 8 bytes; the search field FLD05 offset 11, for 4 bytes; offset 20, for 8 bytes; and offset 30, for 25 bytes.</td>
</tr>
</tbody>
</table>

Extract to multiple output files and control data selection

Figure 141 JCL to extract to multiple output files and control data selection (part 1 of 2)

```plaintext
01//PFPPMAIN EXEC PGM=DFSRRC00,REGION=0M,
02//       PARM=(IFP,PFPSAMP,DBF#FPU0)
03//STEPLIB DD DSN=BMC.PFP.LOAD,DISP=SHR
04//       DD DSN=IMSVS.RESLIB,DISP=SHR
05//EXTFILE2 DD DSN=BMC.PFP.EXTRACT2,
06//       DISP=(,CATLG,DELETE),
07//       UNIT=DISK,SPACE=(CYL,(200,8),RLSE)
```
Figure 141  JCL to extract to multiple output files and control data selection (part 2 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB statements for Fast Path online execution</td>
</tr>
<tr>
<td>02</td>
<td>DBD name is specified as second parameter of PARM statement, which is required for an IFP execution.</td>
</tr>
<tr>
<td>05-07</td>
<td>The extracted output for all areas is written to the DD name EXTFILE.</td>
</tr>
<tr>
<td>08</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statements</td>
</tr>
<tr>
<td>09</td>
<td>The EXTRACT command specifies all areas to be processed, and specifies HDUNLOAD as the format for the output file. Specifying a DBD name is optional, because it has been specified as an IFP parameter.</td>
</tr>
<tr>
<td>10</td>
<td>All SEGA segments are selected.</td>
</tr>
<tr>
<td>11-12</td>
<td>Only SEGB segments that have a ‘9’ at offset 5 is selected.</td>
</tr>
<tr>
<td>13-14</td>
<td>SEGC segments from the selected SEGB segments are excluded where field SEGCKEY is greater than a character value of 19961231.</td>
</tr>
<tr>
<td>15-18</td>
<td>Area PFPSAMP1 is written to an output file using the provided data set name. The resulting output files contains all SEGA segments, all SDEP segments, any SEGB and SEGC that meet the include/exclude selection criteria, all SEGD segments under selected SEGB segments, and all SEGE, SEGF, and SEGG segments.</td>
</tr>
<tr>
<td>19-20</td>
<td>Areas PFPSAMP2 and PFPSAMP3 are written to a single output file using the data set specified on the EXTFILE DD statement. The resulting output files contains all SEGA segments, all SDEP segments, any SEGB and SEGC that meet the include/exclude selection criteria, all SEGD segments under selected SEGB segments, and all SEGE, SEGF, and SEGG segments.</td>
</tr>
</tbody>
</table>
Extract all areas and generate a subset of area data

This scenario creates an HD unload file that contains a subset of the area data. The unload file can then be input into reload to create a smaller test database.

Figure 142  JCL to extract all areas and generate a subset of area data

```plaintext
01//PFPMAIN EXEC PGM=DFSRRC00,REGION=OM,
 02   PARM=(IFP,PFPSAMP,DBF#FPU0)
03//STEPLIB DD DSN=BMC.PFP.LOAD,DISP=SHR
 04   DD DSN=IMSVS.RESLIB,DISP=SHR
 05   DD DISP=SHR,DSN=BMC.DPK.LOAD
 06//$DPICDS DD DISP=SHR,DSN=PF.PPK.DPICDS
 07//$DPITBL DD DISP=SHR,DSN=PF.PPK.DPITBL
08//PFPSYSIN DD *
 09   EXTRACT IAREA=ALL,
10      OUTPUT_DSN_MASK='BMC.PFP.&DBD.AR&AREA#',
11   EXPAND=YES,EXTRACT_FORMAT=HDUNLOAD
12   INCLUDE SEGMENT=SEGA,
13      SAMPLE_INTERVAL=10,
14      SAMPLE_LIMIT=2000
/*
```

Table 36  Descriptive text for JCL to extract all areas and generate a subset of area data

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB statements for Fast Path online execution</td>
</tr>
<tr>
<td>02</td>
<td>DBD name is specified as second parameter of PARM statement, which is required for an IFP execution.</td>
</tr>
<tr>
<td>05-07</td>
<td>DD statements that define data sets for the BMC DATA PACKER/IMS product. These data sets are required because the EXPAND=YES keyword requested data expansion, and because of data manipulation for compressed fields.</td>
</tr>
<tr>
<td>08</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statements</td>
</tr>
<tr>
<td>09</td>
<td>The EXTRACT command specifies all areas to be processed.</td>
</tr>
<tr>
<td>10</td>
<td>Each area is written to a preallocated output file using the data set name mask for dynamic allocation. The previously allocated data set names are</td>
</tr>
<tr>
<td></td>
<td>■ 'BMC.PFP.PFPSAMP.AR001'</td>
</tr>
<tr>
<td></td>
<td>■ 'BMC.PFP.PFPSAMP.AR002'</td>
</tr>
<tr>
<td></td>
<td>■ 'BMC.PFP.PFPSAMP.AR003'</td>
</tr>
<tr>
<td></td>
<td>where the area number allows three characters.</td>
</tr>
<tr>
<td>11</td>
<td>All compressed data is expanded. HDUNLOAD is specified as the format for the output file.</td>
</tr>
<tr>
<td>12</td>
<td>Include segment SEGA and all of its dependents.</td>
</tr>
<tr>
<td>13-14</td>
<td>Limit the unload file by writing every tenth SEGA segment and its dependent segments, up to a maximum of 2000 SEGA segments.</td>
</tr>
</tbody>
</table>
Modify segment layout during extract

Three segments are modified and output in preparation for future application program processing.

**Figure 143  JCL to modify segment layout during extract**

```plaintext
01//PFPMAIN EXEC PGM=DFSRRC00,REGION=OM,
02//         PARM=(IFP,PFPSAMP,DBF#FPU0)
03//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
04//         DD DISP=SHR,DSN=IMSVS.RESLIB
05//         DD DISP=SHR,DSN=BMC.DPK.LOAD
06//$DPICDS DD DISP=SHR,DSN=PF.PPK.DPK.DPICDS
07//$DPITBL DD DISP=SHR,DSN=PF.PPK.DPK.DPITBL
08//PFPSYSIN DD *
09 EXTRACT DBD=PFPSAMP,
10     EXTRACT_FORMAT=EXTRACT,
11     SORT=YES,
12     EXPAND=YES
13 OFILECTL OAREA=ALL,
14     DSNAME='BMC.PFP.PFPSAMP.EXTRACT',
15     DISP=(NEW,CATLG),UNIT=SYSDA,
16     SPACE=(CYL,60,5,RLSE)
17 OUTPUT SEGMENT=SDEP,
18     FIELDS=(3:*,'0',8C' ')
19 OUTPUT SEGMENT=SEGC,
20     FIELDS=(3:12,3P'0',15:*)
21 OUTPUT SEGMENT=SEGG,
22     FIELDS=(SEGGKEY,
23         8:9,5:3,
24         (18:2 EQ 2X'0000') THEN 2C' ' ELSE 18:2,
25         20:*))
/*
```

**Table 37  Descriptive text for JCL to modify segment layout during extract (part 1 of 2)**

<table>
<thead>
<tr>
<th>Line No.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB statements for Fast Path online execution</td>
</tr>
<tr>
<td>02</td>
<td>DBD name is specified as second parameter of PARM statement, which is required for an IFP execution.</td>
</tr>
<tr>
<td>05-07</td>
<td>DD statements that define data sets for the BMC DATA PACKER/IMS product. These data sets are required because the EXPAND=YES keyword requested data expansion, and because of data manipulation for compressed fields.</td>
</tr>
<tr>
<td>08</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statements</td>
</tr>
<tr>
<td>09</td>
<td>Defines DEDB name for EXTRACT process. All areas are input by default.</td>
</tr>
<tr>
<td>10</td>
<td>Data is extracted in the standard default format (EXTRACT_FORMAT=EXTRACT). The format of the records is the layout specified in tables B-3 and B-4 in the Fast Path/EP Series Reference Manual.</td>
</tr>
<tr>
<td>11</td>
<td>The extracted output file is sorted and written in root key sequence.</td>
</tr>
</tbody>
</table>
Create a customized extract file format

The EXTRACT_FORMAT=USER option is specified to enable a customized extract file format. Because no EXCLUDE or OUTPUT subcommands are specified in the scenario, all segments in all areas of the PFPSAMP database are output. Table 39 lists some examples of the output data.

Table 37  Descriptive text for JCL to modify segment layout during extract (part 2 of 2)

<table>
<thead>
<tr>
<th>Line No.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Compressed data is expanded.</td>
</tr>
<tr>
<td>13-16</td>
<td>All areas are written to a single output file using the provided data set name.</td>
</tr>
<tr>
<td>17-18</td>
<td>Modify data in SDEP segments, by increasing the output length by 12 bytes. Copy the original input segment in its entirety, starting at column 3. Then, append a 4-byte packed field (initialized to zero), and an 8-byte character field (initialized to blanks).</td>
</tr>
<tr>
<td>19-20</td>
<td>Modify data in SEGC segments by increasing the output length by 3 bytes. Copy the first 12 bytes (columns 3-14) unchanged, then insert a new 3-byte packed field (initialized to zero), followed by the remainder of the input segment from column 15 to the end.</td>
</tr>
<tr>
<td>21</td>
<td>Modify data in SEGG segments.</td>
</tr>
<tr>
<td>22</td>
<td>The first output field in SEGG is the segment key.</td>
</tr>
<tr>
<td>23</td>
<td>The next two fields that are written to the output are swapped from where they were positioned on the input file: Starting at column 8 from the input, copy 9 bytes to the output. Next, starting at column 5 from the input, copy 3 bytes to the output.</td>
</tr>
<tr>
<td>24</td>
<td>Because the output position in the FIELDS list is the same as the input position of the field being tested, the effect is to conditionally modify the field. If the contents of the two-byte field in column 18 from the input segment is equal to binary zeroes, then write two blanks to the output file. Otherwise, write the input field unchanged.</td>
</tr>
<tr>
<td>25</td>
<td>Write the remainder of the segment unchanged.</td>
</tr>
</tbody>
</table>

Create a customized extract file format

The EXTRACT_FORMAT=USER option is specified to enable a customized extract file to be produced in the format that is required by an existing application program. Because no EXCLUDE or OUTPUT subcommands are specified in the scenario, all segments in all areas of the PFPSAMP database are output. Table 39 lists some examples of the output data.

Figure 144  JCL to create a customized extract file format (part 1 of 2)

```
01//PFPMAIN EXEC PGM=DFSRRCD0,REGION=OM,
02 PARM=(IFP,PFPsamp,DF#FPUO)
03//STEPLIB DD DSN=BMC.PFP.LOAD,DISP=SHR
04// DD DSN=IMSVS.RESLIB,DISP=SHR
05//OAREA001 DD DSN=BMC.PFP.USEROUT1,DISP=(,CATLG,DELETE),
06 // UNIT=SYSDA,SPACE=(CYL,(5,1),RLSE)
07//OAREA002 DD DSN=BMC.PFP.USEROUT2,DISP=(,CATLG,DELETE),
08 // UNIT=SYSDA,SPACE=(CYL,(5,1),RLSE)
09//OAREA003 DD DSN=BMC.PFP.USEROUT3,DISP=(,CATLG,DELETE),
10 // UNIT=SYSDA,SPACE=(CYL,(5,1),RLSE)
11//PFPSYSIN DD *
12 EXTRACT DBD=PFPSAMP,IAREA=ALL,EXTRACT_FORMAT=USER,
13 SEGMENT_RECORD_PREFIX=(X'0003',
14 :2X :=SEGMENT_CODE.
```
Create a customized extract file format

Figure 144  JCL to create a customized extract file format (part 2 of 2)

```java
15 SEGMENT_NAME)
16 USER_RECORD BREAK=(AREA,BEFORE),
17 FIELDS=(X'0000',
18 AREA_NAME,
19 :2X :=AREA_NUMBER,
20 :%YYYY%%MO%%DD%%24%%MI%%SS%':=SYSDATETIME())
21 USER_RECORD BREAK=(AREA,BEFORE),
22 FIELDS=(X'0001',
23 DBD_NAME,
24 DBD_AREAS, DBD_SEGMENTS)
25 USER_RECORD BREAK=(AREA,AFTER),
26 FIELDS=(X'0004',
27 X'0000',
28 SEGMENT_COUNT(1),
29 SEGMENT_COUNT(),
30 F'0')
/*

Table 38  Descriptive text for JCL to create a customized extract file format (part 1 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB statements for Fast Path online execution</td>
</tr>
<tr>
<td>02</td>
<td>DBD name is specified as second parameter of PARM statement, which is required for an IFP execution.</td>
</tr>
<tr>
<td>05-06</td>
<td>Extract output file for area PFPSAMP1.</td>
</tr>
<tr>
<td>07-08</td>
<td>Extract output file for area PFPSAMP2.</td>
</tr>
<tr>
<td>09-10</td>
<td>Extract output file for area PFPSAMP3.</td>
</tr>
<tr>
<td>11</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statements</td>
</tr>
<tr>
<td>12</td>
<td>Defines DEDB name for EXTRACT process. All areas are input. Data is extracted from all segments for all areas, using EXTRACT_FORMAT=USER. This format enables use of any customized format that is needed. The areas are dynamically allocated from their DFSMDA members from the STEPLIB IMSVS.DFSMDA data set.</td>
</tr>
<tr>
<td>13</td>
<td>All segments are written to the output file. Segment records have a common prefix that is placed first, before each segment data (segment data will include LL field for variable length). The contents of the common prefix is specified by using the SEGMENT_RECORD_PREFIX keyword. The first prefix field for the segment record is x'0003'.</td>
</tr>
<tr>
<td>14</td>
<td>The next field of the segment prefix is doing an assignment (=) of the one-byte built-in variable of SEGMENT_CODE into a two-byte assignment variable (:2X)</td>
</tr>
<tr>
<td>15</td>
<td>The last field of the segment prefix is an eight-byte built-in variable for SEGMENT_NAME.</td>
</tr>
<tr>
<td>16, 21</td>
<td>For each area, two header records are produced which are written first (AREA,BEFORE) in the extract output file.</td>
</tr>
<tr>
<td>16</td>
<td>The first header record is created by specifying the USER_RECORD subcommand. The BREAK keyword signifies that this record is to be written before the rest of the area information is written (AREA,BEFORE).</td>
</tr>
</tbody>
</table>
Create a customized extract file format

Table 38  Descriptive text for JCL to create a customized extract file format (part 2 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>The FIELDS keyword specifies the content of the header record, as follows: record identifier of X’0000’</td>
</tr>
<tr>
<td>18</td>
<td>followed by an 8-byte built-in variable of AREA_NAME</td>
</tr>
<tr>
<td>19</td>
<td>Next is the assignment (:=) of 1-byte built-in variable AREA_NUMBER (2-byte built-in variable for IMS version 8.1 and above) into a two-byte assignment variable (2X)</td>
</tr>
<tr>
<td>20</td>
<td>Next is a date field with an assignment (:=) of 12-byte calendar date, of the built-in function SYSDATETIME() into the specified string literal. The literal indicates the time in military time by specifying hours with %24%.</td>
</tr>
<tr>
<td>21</td>
<td>The second header record is created by specifying the USER_RECORD subcommand. The BREAK keyword signifies that this record is to be written before the rest of the area information is written.</td>
</tr>
<tr>
<td>22</td>
<td>FIELDS keyword specifies the content of the second header record, as follows: record identifier X’0001’</td>
</tr>
<tr>
<td>23</td>
<td>followed by an 8-byte built-in variable for DBD_NAME</td>
</tr>
<tr>
<td>24</td>
<td>Next are the built-in variables for the number of DBD AREAS and the number of segment types for DBD SEGMENTS, each specified as two fixed bytes.</td>
</tr>
<tr>
<td>25</td>
<td>An area trailer record follows all segment records from the area. The area trailer record is created by specifying the USER_RECORD subcommand. The BREAK keyword signifies that this record is to be written after the rest of the area information is written (AREA,AFTER).</td>
</tr>
<tr>
<td>26</td>
<td>FIELDS keyword specifies the content of the trailer record, as follows: record identifier X’0004’</td>
</tr>
<tr>
<td>27</td>
<td>followed by two bytes of zeroes (X’0000’)</td>
</tr>
<tr>
<td>28</td>
<td>Next is the 4-byte built-in function of SEGMENT_COUNT (1), giving the total number of root segments to be processed. The root segment is indicated by the segment code ‘1’ (default with no parm given).</td>
</tr>
<tr>
<td>29</td>
<td>Next is the 4-byte built-in function of SEGMENT_COUNT(), giving the total of all segments processed.</td>
</tr>
<tr>
<td>30</td>
<td>Next is 4 bytes of zeroes as specified by (F’0’) using the default length 4 for the ‘F’ data type.</td>
</tr>
</tbody>
</table>

Table 39  Data output from JCL to create a customized extract file format (part 1 of 2)

<table>
<thead>
<tr>
<th>Format</th>
<th>Data output</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>..PFPSAMP1..19990520175735 00DCECD710119990520175735</td>
<td>record=x’0000’ areaname=PFPSAMP1 area number=x’0001’ date-time=1999/05/20 17:57:35</td>
</tr>
<tr>
<td>hex</td>
<td>..PFPSAMP1.... 00DCECD700000 01767214720306</td>
<td>record=x’0001’ dbname=PFPSAMP number of areas=x’0003’ number of segments=x’0006’</td>
</tr>
</tbody>
</table>
The scenarios in this section show how to use the online IMAGECOPY command with keywords and subcommands.
Table 40  IMAGECOPY scenarios

<table>
<thead>
<tr>
<th>Primary Command/ Scenario Task</th>
<th>Subcommand/Keyword</th>
<th>Concept/Process</th>
<th>Page</th>
</tr>
</thead>
</table>
| IMAGECOPY Simultaneously Create Online Image Copy and Analyze Pointers | IAREA POINTER_VALIDATION IC              | ■ create online image copy of specified area in DEDB  
■ request full pointer analysis | 218  |
| IMAGECOPY Disable Automatic Pointer Analysis Stack Image Copies | POINTER_VALIDATION IC with STACK_NAME COMPRESSION NOTIFY | ■ turn off (disable) default automatic pointer analysis  
■ create stacked image copies of all areas in DEDB  
■ request recording of image copy data sets in system catalog  
■ request registration of image copy data sets with DBRC  
■ request backup image copy to be written to offsite tape volume | 219  |

Simultaneously create online image copy and analyze pointers

Figure 145  JCL to simultaneously create online image copy and analyze pointers

```plaintext
01//PFP EXEC PGM=DFSRRC00,REGION=0M,  
02// PARM=(IFP,PFPSAMP,DBF#FPU0)  
03//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD  
04// DD DISP=SHR,DSN=IMSVS.RESLIB  
05//ICOPY DD DSN=PFPSAMP.ICOPY.PFPSAMP.PFPSAMP1(+1),UNIT=TAPE,  
06// DISP=(NEW,CATLG)  
07//PFPSYSIN DD *  
08 IMAGECOPY IAREA=PFPSAMP1,POINTER_VALIDATION=FULL  
09 IC DDNAME=ICOPY /*
```

Table 41  Descriptive text for JCL to simultaneously create online image copy and analyze pointers (part 1 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB DD statements for Fast Path online execution</td>
</tr>
<tr>
<td>05-06</td>
<td>Specifies the image copy data set for area PFPSAMP1.</td>
</tr>
<tr>
<td>07</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statements</td>
</tr>
</tbody>
</table>
 Disable automatic pointer analysis and stack image copies

Table 41  Descriptive text for JCL to simultaneously create online image copy and analyze pointers (part 2 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>08</td>
<td>A complete analysis (full pointer validation) of area PFPSAMP1 is performed.</td>
</tr>
<tr>
<td>09</td>
<td>An output image copy of area PFPSAMP1 is written to DD ICOPY.</td>
</tr>
</tbody>
</table>

Figure 146  JCL to disable automatic pointer analysis and stack image copies

```jcl
01//PFP EXEC PGM=DFSRRC00,REGION=0M,
02//   PARM=(IFP,PFPSAMP,DBF#FPU0)
03//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
04//   DD DISP=SHR,DSN=IMSVS.RESLIB
05//PFPSYSIN DD *
06// IMAGECOPY IAREA=ALL,
07//   POINTER_VALIDATION=NONE
08// IC STACK_NAME=ONSITE,
09//   UNIT=TAPE,DISP=(NEW,CATLG),NOTIFY=YES,COMPRESSION=FSE,
10//   DSNAME='PFP.ICOPY.&DBD.&AREA(+1)'
11// IC STACK_NAME=OFFSITE,
12//   UNIT=TAPE,DISP=(NEW,KEEP),NOTIFY=NO,
13//   DSNAME='PFP.ICOPY.&DBD.&AREA.&DATE.&TIME'
/*
```

Table 42  Descriptive text for JCL to disable automatic pointer analysis and stack image copies (part 1 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB DD statements for Fast Path online execution</td>
</tr>
<tr>
<td>05</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statements</td>
</tr>
<tr>
<td>06</td>
<td>All areas of the DEDB are image copied.</td>
</tr>
<tr>
<td>07</td>
<td>Pointer validation and physical validation are not performed. POINTER_VALIDATION=NONE overrides the default checksum validation.</td>
</tr>
<tr>
<td>08-10</td>
<td>A compressed image copy data set is created for each area in the database. The image copy data sets are registered with DBRC. These data sets are stacked onto tape.</td>
</tr>
<tr>
<td>09</td>
<td>All image copy data sets that are part of the same stack group are written in the order processed onto the same tape volume (or volumes) as file number 1, 2, and 3.</td>
</tr>
<tr>
<td>10</td>
<td>Dynamic allocation is always used for stacked image copy data sets. The DBD and area name are substituted in the image copy data set name to create a unique name for each area.</td>
</tr>
<tr>
<td>11-13</td>
<td>A non-compressed image copy data set is created for each area in the database. If creation of the image copies that were requested on line 08 was successful, these image copy data sets are not registered with DBRC. These data sets are stacked onto a tape device.</td>
</tr>
</tbody>
</table>
Online EXTEND command scenarios

The scenarios in this section show how to use the online EXTEND command with related keywords and subcommands.

Table 43  EXTEND scenarios

<table>
<thead>
<tr>
<th>Primary Command/ Scenario Task</th>
<th>Subcommand/Keyword</th>
<th>Concept/Process</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTEND Simultaneously Extend SDEP, Analyze, and Create Image Copy</td>
<td>EXTEND_SDEP(pointer_validation) IC</td>
<td>■ perform one-step, concurrent maintenance and analysis tasks ■ create image copy of extended DEDB</td>
<td>221</td>
</tr>
<tr>
<td>EXTEND Extend SDEP Disable Automatic Pointer Analysis Stack Image Copies</td>
<td>IAREA EXTEND_SDEP(pointer_validation) IC with STACK_NAME</td>
<td>■ turn off (disable) automatic pointer analysis ■ create stacked image copies of extended DEDB</td>
<td>221</td>
</tr>
<tr>
<td>EXTEND Extend IOVF</td>
<td>IAREA EXTEND_IOVF IC</td>
<td>■ limit space extend and associated tasks to one area only ■ extend IOVF ■ create output image copy of extended area</td>
<td>221</td>
</tr>
</tbody>
</table>
Simultaneously extend SDEP, analyze, and create image copy

Figure 147  JCL to simultaneously extend SDEP, analyze, and create image copy

```
01//PFP EXEC  PGM=DFSRRCOO,REGION=OM.
02//  PARM=(IFP,PFPSAMP,DBF#FPU0)
03//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
04// DD DISP=SHR,DSN=IMSVS.RESLIB
05//PFPSYSIN DD *
06 EXTEND DBD=PFPSAMP, EXTEND_SDEP=(CONTROL_INTERVALS,100)
07  POINTER_VALIDATION=FULL
08  IC DSNAME='PFP.ICOPY.&DBD.&AREA(+1)',
09  UNIT=TAPE,DISP=(NEW,CATLG)
/*
```

Table 44  Descriptive text for JCL to simultaneously extend SDEP, analyze, and create image copy

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB DD statements for Fast Path online execution</td>
</tr>
<tr>
<td>05</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statements</td>
</tr>
<tr>
<td>06</td>
<td>All areas of the DBD are extended. Specifying the DBD name is not necessary and must match the DBD name in the IFP PARM statement in line 2. If the names do not match, an error is displayed and the job will terminate. Adding 100 control intervals to SDEP increases the SDEP portion.</td>
</tr>
<tr>
<td>07</td>
<td>A complete analysis (full pointer validation) of all areas is performed on the DEDB.</td>
</tr>
<tr>
<td>08-09</td>
<td>An output image copy is created in the same job step as the extend. The DBD and area name are substituted in the image copy data set name to create a unique name for each area.</td>
</tr>
</tbody>
</table>

Extend SDEP, disable automatic pointer analysis, and stack image copies

Figure 148  JCL to extend SDEP, disable automatic pointer analysis, and stack image copies

```
01//PFP EXEC  PGM=DFSRRCOO,REGION=OM.
02//  PARM=(IFP,PFPSAMP,DBF#FPU0)
03//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
04// DD DISP=SHR,DSN=IMSVS.RESLIB
05//PFPSYSIN DD *
06 EXTEND DBD=PFPSAMP, IAREA=ALL,
07  EXTEND_SDEP=(CONTROL_INTERVALS,500)
08  POINTER_VALIDATION=NONE.
09  IC STACK_NAME=NAME1,
10  UNIT=TAPE,DISP=(NEW,CATLG),
11  DSNAME='PFP.ICOPY.&DBD.&AREA(+1)'
/*
### Extend IOVF

#### Table 45  Descriptive text for JCL to extend SDEP, disable automatic pointer analysis, and stack image copies

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB DD statements for Fast Path online execution.</td>
</tr>
<tr>
<td>05</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statements.</td>
</tr>
<tr>
<td>06</td>
<td>All areas of the DBD are extended. Specifying the DBD name is not necessary and must match the DBD name in the IFP PARM statement in line 2. If the names do not match, an error is displayed and the job will terminate.</td>
</tr>
<tr>
<td>07</td>
<td>Adding 500 control intervals to SDEP increases the SDEP portion.</td>
</tr>
<tr>
<td>08</td>
<td>Pointer validation and physical validation are not performed. POINTER_VALIDATION=NONE overrides the default checksum validation.</td>
</tr>
<tr>
<td>09-11</td>
<td>An image copy data set is created for each area in the database. These data sets are stacked onto tape.</td>
</tr>
<tr>
<td>09-10</td>
<td>All image copy data sets that are part of the same stack group are written in the order processed onto the same tape volume (or volumes) as file number 1, 2, and 3.</td>
</tr>
<tr>
<td>11</td>
<td>Dynamic allocation is always used for stacked image copy data sets. The DBD and area name are substituted in the image copy data set name to create a unique name for each area.</td>
</tr>
</tbody>
</table>

#### Figure 149  JCL to extend IOVF

```
01//PFP EXEC PGM=DFSRRC00,REGION=0M
02//       PARM=(IFP,PFPSAMP,DBF#FPU0)
03//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
04//       DD DISP=SHR,DSN=IMSVS.RESLIB
05//PFPSYSIN DD *
06 EXTEND DBD=PFPSAMP,IAREA=PFPSAMP1,
07       POINTER_VALIDATION=QUICK.
08 EXTEND_IOVF=(UNITS_OF_WORK,50)
09 IC DSNAME='PFP.ICOPY.&DBD.&AREA(+1)',
10 UNIT=TAPE,DISP=(NEW,CATLG)
/*
```

#### Table 46  Descriptive text for JCL to extend IOVF (part 1 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-04</td>
<td>EXEC and STEPLIB DD statements for Fast Path online execution.</td>
</tr>
<tr>
<td>05</td>
<td>PFPSYSIN DD for Fast Path Online Suite control statements.</td>
</tr>
<tr>
<td>06</td>
<td>Area PFPSAMP1 is identified on the EXTEND statement, and is dynamically allocated from its dynamic allocation member in the IMSVS.DFSMDA data set in the STEPLIB concatenation.</td>
</tr>
</tbody>
</table>
Table 46  Descriptive text for JCL to extend IOVF (part 2 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>07</td>
<td>If you have a license for the BMC Fast Path Analyzer/EP product, quick (checksum) pointer validation is performed on area PFPSAMP1 by default. This pointer validation is performed by default, even if you do not specify it explicitly.</td>
</tr>
<tr>
<td>08</td>
<td>Adding 50 UOWs to IOVF increases space for IOVF. Increasing space in this manner allows you to perform a DBD change of the UOW value at a later time.</td>
</tr>
<tr>
<td>09-10</td>
<td>During the area extend, an output image copy is taken simultaneously using standard storage and disposition parameters. The DBD and area name are substituted in the image copy data set name to create a unique name for each area.</td>
</tr>
</tbody>
</table>
Index

A
ACCESS keyword 76
active SDEP boundaries 83
analysis function 25, 70
analyze an entire area 77
analyze an entire DEDB 78
analyze specific areas 78
available types of analysis 71, 72
enhancing performance 98
generating an image copy during analysis 98
generating analysis reports 88
input caching 98
keywords and subcommands
ANALYZE command 76
monitoring exception conditions 93
performing concurrent analysis 74
performing online analysis 72
report summaries 101
selecting reports 88
specifying threshold settings 93
suppressing repetitious messages 95
ANALYZE command 70
analyzing an area while reorganizing 49, 155
APF authorization 27
Area Change Modeling Utility 95
area data set allocation 79
area integrity, verifying 79
Area Summary Report 101
EXCLUDE subcommand on online EXTRACT 211
EXTEND 220
EXTRACT 207
how to interpret examples 196
IC (image copy) subcommand 200, 202, 206, 219, 221, 222
INCLUDE subcommand on online EXTRACT 211, 212
list of tasks 196
offline DEDB data extract function 207
OFILECTL subcommand on online EXTRACT 219, 221
OUTPUT subcommand on EXTRACT 213
OUTPUT subcommand on online EXTRACT 209
REORGANIZE 198
REPORT subcommand 207
requesting analysis reports 207
USER_RECORD subcommand on EXTRACT 215
command sets 26
commands
ANALYZE 70, 76
DUMP 176
EXTEND 152
EXTRACT 111, 119
GLOBAL 37, 75
IMAGECOPY 99, 139, 140
OPTIONS 40, 192
REORGANIZE 44
COMPRESS keyword 51, 56
COMPRESSION keyword 141
concurrent analysis 74
concurrent analysis function 26
DBRC considerations 76
MADS considerations 76
console commands 177
conventions, documentation 18
CORRECTIONS_FILECTL subcommand 76, 84
creating an output image copy while reorganizing 49, 155
customer support 3

B
BMC Software Support
diagnostic commands 176
DUMP facility 176
Problem Documentation form 176
Problem Summary form 175
BMC Software, contacting 2
BREAK keyword 135, 215

C
change function 28
changes to product 19
command set examples
ANALYZE 203

D
DBD keyword 51, 52, 76, 77, 119, 141, 156, 157
DBRC keyword 75
DBRC region services 36
DDNAME keyword 76, 86, 122, 141
DFSRRC00 module 34
diagnostic DUMP facility 176
diagnostic procedures 167
direct mode 98
DOVF_FREESPACE_PERCENT keyword 93, 204
DSNAME keyword 76, 87, 123, 141
DUMP command 176
DUMP console command 177
DUMP facility 176
DUMP_TYPE keyword 176, 177

E
electronic documentation 17
error queue elements (EQEs) 76
error tolerance 66
ERROR_THRESHOLD keyword 51, 66
exception conditions, monitoring 93
EXCLUDE subcommand 124
EXPAND keyword 132, 213
expressions 177
EXTEND command 152
extend function 25, 154
extend an entire DEDB 157
extend specific areas 157
keywords and subcommands 156
requirements 152
extend the IOVF and SDEP 152
EXTEND_IOVF keyword 156
EXTEND_IOVF_#UOWS keyword 51, 202, 222
EXTEND_SDEP keyword 156
EXTEND_SDEP_#CIS keyword 51
extending IOVF 57, 159
extending SDEP 59
EXTRACT command 111, 119
extract file 96
extract file format 115
extract formatting options 110
extract function 25
keywords and subcommands 113
sample scenarios 207
EXTRACT_DDNAME keyword 96
EXTRACT_FORMAT keyword 114, 209, 213, 214
extracting data from an online DEDB 110
keywords and subcommands 113
extracting segment data 110

F
Fast Path Online Analyzer/EP product functions 25
Fast Path Online Analyzer/EP reports 88
effect of pointer validation options 90
excluding selected reports 90
producing only selected reports 90
report formatting 91
report routing 91
requesting all reports 89
specifying a UOW range 90
Fast Path Online Image Copy/EP product functions 26
Fast Path Online Reorg/EP product functions 24
Fast Path Online Suite products 21
features summary 26
product capabilities 22
product installation 26
products included in suite 21, 22
FIELDS keyword 133, 136, 213, 215
FRAGMENTATION_PERCENT keyword 51, 54
Free Space Analysis Report 101
free space threshold 93
FREESPACE_ANALYSIS keyword 88, 207
FREESPACE_DOVF_IOVF keyword 93
FREESPACE_RAA_DOVF keyword 93
FREESPACE_RAA_IOVF keyword 93
frequency of segment occurrence 128

G
generation data group (GDG) 144
GLOBAL command 37, 75
global keyword values 37
global values for keywords 37

H
header records 135
Help, online 17
historical summary records 96
history extract file 96
history file 96
HISTORY_DDNAME keyword 37, 77, 96

I
I/O threshold 93
IAREA keyword 51, 52, 76, 77, 119, 141, 142, 156, 157
IBM HD Reorganization Unload Utility 117
IC subcommand 37, 51, 68, 77, 141, 156, 164
ICACHE keyword 77, 98
IDCAMS 202
image copy function 139
analyzing area during image copy 147
create image copy of an entire DEDB 142
create image copy of specific areas 142
creating image copy during reorganization 149
during reorganization 68, 164
inputs and outputs 140
keywords and subcommands 140, 141
requesting analysis during image copy 99
sample scenarios 200, 202, 206, 219, 221, 222
stacking image copies 144
IMAGECOPY command 99, 139, 140
IMS region controller module 34
IMS RESLIB 27
IMS/VS commands 36
IMS/VS logging 48
INCLUDE subcommand 124
input caching:
  analysis function 98
INPUT_DSN_MASK keyword 76
INSERT_LIMIT_COUNT keyword 200
IOVF extend 57, 158
IOVF extension 152
IOVF Space Analysis Report 102
IOVF_FREESPACE_PERCENT keyword 93, 204
IOVF_SAVE_THRESHOLD keyword 51, 201
IOVF_SPACE_ANALYSIS keyword 88, 207
IOVF_USED_PERCENT keyword 93
MESSAGE_LEVEL 40, 186
MESSAGE_LIMIT 40, 42, 186
MESSAGE_NUMBER 40, 186
MESSAGE_OVERRIDE 190
NOTIFY 141
ORPHANED_SDEP_MSG 76, 77, 84, 156
POINTER_VALIDATION 51, 76, 80, 99, 141, 147, 156
RAA_FREESPACE_PERCENT 93, 204
RAP_OVERFLOW_PERCENT 93, 204
RAP_VALIDATION 51, 76, 141, 156
RECORD_IO_AVERAGE 93
RECORD_IO_MAXIMUM 93
REPOSITORY_DSNAME 186, 192, 205
REPOSITORY_GROUP 205
REPOSITORY_OVERWRITE 205
REPOSITORY_RETENTION_COUNT 205
ROOT_IO_MAXIMUM 94
SAMPLE_INTERVAL 128, 212
SAMPLE_LIMIT 128, 212
SDEP_VALIDATION 51, 76, 82, 99, 141, 148, 156
SEGMENT 200, 209, 213
SEGMENT_RECORD_PREFIX 134, 214
SEGMENT_RECORD_SUFFIX 134
SELECT_UOW 51, 55, 200, 201
SORT 136
SORT_OPTION 136
STACK_NAME 141, 144, 200, 219, 221
STARTUOW 90
STOPUOW 90
SYNONYM_CHAIN_LENGTH 94
SYNONYM_CHAIN_LENGTH keyword 94
SYNONYM_CHAIN_MAXIMUM 94
SYNONYM_CHAIN_PERCENT 94
TYPE_RUN 156
UOW_DETAILED_ANALYSIS 90
UOW_DOVF_PERCENT 94
UOW_DOVF_AVERAGE 94
UOW_IOVF_MAXIMUM 94
UOW_IOVF_PERCENT 94
WHERE 126, 129, 134, 200

K

keywords
  ACCESS 76
  BREAK 135, 215
  COMPRESS 51, 56
  COMPRESSION 141
  DBD 51, 52, 76, 77, 119, 141, 156, 157
  DBRC 75
  DDNAME 76, 86, 122, 141
  DOVF_FREESPACE_PERCENT 93, 204
  DSNAME 76, 87, 123, 141
  DUMP_TYPE 176, 177
  ERROR_THRESHOLD 51, 66
  EXPAND 132, 213
  EXTEND_IOVF 156
  EXTEND_IOVF_#UOWS 51, 202, 222
  EXTEND_SDEP 156
  EXTEND_SDEP_#CIS 51
  EXTRACT_DDNAME 96
  EXTRACT_FORMAT 114, 209, 213, 214
  FIELDS 133, 136, 213, 215
  FRAGMENTATION_PERCENT 51, 54
  FREESPACE_RAA_DOVF 93
  FREESPACE_ANALYSIS 207
  FREESPACERATIO_DOVF_IOVF 93
  FREESPACERATIO_RAA_IOVF 93
  HISTORY_DDNAME 37, 77
  IAREA 51, 52, 76, 77, 119, 141, 142, 156, 157
  ICACHE 77, 98
  INPUT_DSN_MASK 76
  INSERT_LIMIT_COUNT 200
  IOVF_FREESPACE_PERCENT 93, 204
  IOVF_SAVE_THRESHOLD 51
  IOVF_SAVE_THRESHOLD 201
  IOVF_SPACE_ANALYSIS 207
  IOVF_USED_PERCENT 93
  LOCATION 200

L

LB 83
LE 83
LIST subcommand 186, 190
load control 60
LOADCTL subcommand 51, 61
LOCATION keyword 200
M

MADS (multiple area data sets)
  multiple area data sets 76
message modification 183
MESSAGE_LEVEL keyword 40, 186
MESSAGE_LIMIT keyword 40, 42, 186
MESSAGE_NUMBER keyword 40, 186
MESSAGE_OVERRIDE keyword 190
MESSAGE_SUPPRESSION keyword 42, 50, 77, 96, 155, 156
messages
  changing severity level 189
  listing stored message customizations 190
  permanent customization options 39, 184
  permanently modifying messages 184
  restoring overrides 41, 190
  suppressing 42, 50, 95, 155, 189
  temporarily modifying messages 40
  temporary customization options 40
MODEL_DDDNAME keyword 77, 95
Modeling Utility 95
monitoring area characteristics
  reporting area characteristics 88

N

NOTIFY keyword 141

O

offline mode processing 32
online analysis function 72
  sample scenarios 203
online DEDB data extract function 25, 110
  controlling content of segment header and trailer records 135
  controlling segment expansion 132
  controlling the content of segment record prefixes and suffixes 134
  controlling the output of segment records 133
  extract an entire DEDB 119
  extract data from an entire DEDB 119
  extract data from specific areas 120
  including and excluding segments for input 124
  keywords and subcommands 113
  limiting generation of output segment records 129
  output format options 110
  sample scenarios 207
  selecting the extract file format 114
  sorting the extract file 136
online Extend function
  sample scenarios 220
online extend function 154
online Help 17
online image copy function 26
  online IOVF extend 57, 158
  online mode processing 32
  online reorganization function 48
    sample scenarios 198
  online SDEP extend 57, 158
  OPTIONS command 40, 192
  orphaned sequential dependents 84
  ORPHANED_SDEP_MSG keyword 76, 77, 84, 156
  out-of-space condition 57, 152, 158
  OUTPUT subcommand 124, 129, 133
  overflow thresholds 94
  OVERRIDE subcommand 40, 186

P

PFPEPR00 utility program 185
physical integrity 79
physical validation 79
Pointer Analysis Report 103
pointer integrity validation 80
pointer validation 80
POINTER_ANALYSIS keyword 88
POINTER_VALIDATION keyword 51, 76, 80, 99, 141, 147, 156
problems 170
product changes 19
product installation 26
product support 3, 167
publications, related 18

R

RAA_FREESPACE_PERCENT keyword 93, 204
RAP utilization threshold 94
RAP_OVERFLOW_PERCENT keyword 93, 204
RAP_VALIDATION keyword 51, 76, 141, 156
RECON data sets 36
Record Length Analysis Report 104
Record Placement Analysis report 105
Record Profile Analysis Report 105
RECORD_IO_AVERAGE keyword 93
RECORD_IO_MAXIMUM keyword 93
RECORD_IOVF_PERCENT keyword 94
RECORD_LENGTH_ANALYSIS keyword 89
RECORD_PLACEMENT_ANALYSIS keyword 89
RECORD_PROFILE_ANALYSIS keyword 89
region controller module 34
region size 33
related publications 18
reload function 28
reorganization function 25, 48
  analyzing the area during reorganization 49, 66, 155, 163
  buffer usage 49
  controlling UOW error tolerance 66
creating an image copy during reorganization 68, 164
extend IOVF and SDEP online 57, 158
image copying while reorganizing 49, 155
IMS/VS logging 48
inputs and outputs 46, 153
keywords and subcommands 51
performance considerations 48, 155
placing segments with load control 60
reorganize an entire DEDB 52
reorganize specific areas 52
restrictions 46
UOW locking 48
REORGANIZE 156
REORGANIZE command 44
report formatting 91
report keywords
FREESPACE_ANALYSIS 88
IOVF_SPACE_ANALYSIS 88
POINTER_ANALYSIS 88
RECORD_LENGTH_ANALYSIS 89
RECORD_PLACEMENT_ANALYSIS 89
RECORD_PROFILE_ANALYSIS 89
SEGMENT_IO_ANALYSIS 88
SEGMENT_LENGTH_ANALYSIS 89
SEGMENT_PLACEMENT_ANALYSIS 89
SYNONYM_CHAIN_ANALYSIS 89
UOW_DETAILED_ANALYSIS 88
report routing 91
REPORT subcommand 37, 51, 76, 88, 141, 156
report summaries 101
REPORT_DDNAME keyword 91
REPORT_DEFAULT keyword 89, 90
REPORT_HEADING keyword 91
REPORT_LINE_COUNT keyword 91
reports
Area Summary Report 101
Free Space Analysis 101
IOVF Space Analysis 102
Pointer Analysis Report 103
Record Length Analysis Report 104
Record Placement Analysis 105
Record Profile Analysis Report 105
Segment I/O Analysis Report 103
Segment Placement Analysis Report 104
Synonym Chain Analysis Report 105
UOW Detailed Analysis Report 102
repository facility
activating message customizations 191
REPOSITORY_DSNAME keyword 186, 192, 205
REPOSITORY_GROUP keyword 205
REPOSITORY_OVERWRITE keyword 205
REPOSITORY_RETENTION_COUNT keyword 205
RESET subcommand 41, 186
ROOD_IO_AVERAGE keyword 94
ROOT_IO_MAXIMUM keyword 94
sample command scenarios
EXCLUDE subcommand on online EXTRACT 211
EXTRACT 207
how to interpret samples 196
IC (image copy) subcommand 200, 202, 206, 219, 221, 222
INCLUDE subcommand on online EXTRACT 211, 212
list of tasks 196
REPORT subcommand 207
requesting analysis reports 207
USER_RECORD subcommand on EXTRACT 215
SAMPLE_INTERVAL keyword 128, 212
SAMPLE_LIMIT keyword 128, 212
SDEP extend 57, 158
SDEP extension 152
SDEP logical begin 83
SDEP logical end 83
SDEP validation 82
SDEP_VALIDATION keyword 51, 76, 82, 99, 141, 148, 156
Segment I/O Analysis Report 103
SEGMENT keyword 200, 209, 213
segment occurrence frequency 128
Segment Placement Analysis Report 104
segment record prefixes and suffixes 134
segment sampling 128
SEGMENT_IO_ANALYSIS keyword 88
SEGMENT_LENGTH_ANALYSIS keyword 89
SEGMENT_PLACEMENT_ANALYSIS keyword 89
SEGMENT_RECORD_PREFIX keyword 134, 214
SEGMENT_RECORD_SUFFIX keyword 134
SELECT_UOW keyword 51, 55, 200, 201
segmenting UOWs for reorganization 44
sequential dependents, orphaned 84
sequential mode 98
SORT keyword 136
SORT_OPTION keyword 136
STACK_NAME keyword 141, 144, 200, 219, 221
stacking image copies 144
STARTUOW keyword 90
STOPUOW keyword 90
subcommands
CORRECTIONS_FILECTL 76, 84
EXCLUDE 124
IC 37, 51, 68, 77, 141, 156, 164
INCLUDE 124
LIST 186, 190
LOADCTL 51, 61
OUTPUT 124, 129, 133
OVER 40, 186
REPORT 37, 51, 76, 88, 141, 156
RESET 41, 186
THRESHOLD 37, 51, 77, 93, 141, 156, 204
USER_RECORD 135, 215
summary of changes 19
support, customer 3
Synonym Chain Analysis Report 105
SYNONYM_CHAIN_ANALYSIS keyword 89
SYNONYM_CHAIN_LENGTH 94
SYNONYM_CHAIN_MAXIMUM keyword 94
SYNONYM_CHAIN_PERCENT keyword 94
SYNONYM_CHAIN_PERCENT,SYNONYM_CHAIN_PERCENT keyword 94
syntax statement conventions 19

T

technical support 3
threshold keywords 94
DOVF_FREESPACE_PERCENT 93
FREESPACE_RAA_IOVF 93
FREESPACE_DOVF_IOVF 93
FREESPACE_RAA_DOVF 93
IOVF_FREESPACE_PERCENT 93
IOVF_USED_PERCENT 93
RAA_FREESPACE_PERCENT 93
RAP_OVERFLOW_PERCENT keyword 93
RECORD_IO_AVERAGE 93
RECORD_IO_MAXIMUM 93
RECORD_IOVF_PERCENT 94
ROOT_IO_AVERAGE 94
ROOT_IO_MAXIMUM 94
SYNONYM_CHAIN_LENGTH 94
SYNONYM_CHAIN_MAXIMUM 94
SYNONYM_CHAIN_PERCENT 94
UOW_DOVF_PERCENT 94
UOW_IOVF_AVERAGE 94
UOW_IOVF_MAXIMUM 94
UOW_IOVF_PERCENT 94
threshold settings for analysis function 93
THRESHOLD subcommand 37, 51, 77, 93, 141, 156, 204
thresholds
free space 93
I/O 93
overflow 94
RAP utilization 94

V

validation
active SDEP boundaries 83
orphaned sequential dependents 84
physical 79
pointer 80
SDEP 82
verifying area integrity 79
validation function 25

W

WHERE keyword 126, 129, 134, 200

U

unload function 28
unload/reload function 28
UOW Detailed Analysis Report 102
Notes