
Supporting

Version 3.11 of Fast Path Analyzer/EP
Version 3.11 of Fast Path Indexer/EP
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
Version 3.11 of Fast Path Online Restructure/EP
Version 3.11 of Fast Path 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


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, 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 UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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](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 23
Related publications .................................. 24
Conventions ........................................ 24
Syntax statements ................................... 25
Syntax diagrams ..................................... 26
Summary of changes ................................ 27

Chapter 1  JCL statements 29
Introduction ........................................ 29
EXEC statement ..................................... 30
  Online EXEC statement .............................. 30
  Offline EXEC statement ............................. 31
  BMP EXEC statement ................................. 31
  64-bit storage EXEC statement for offline processing ................................. 32
DD statements ........................................ 33

Chapter 2  Command language 49
Command language conventions ....................... 54
Input control statements ................................ 56
OPTIONS command keywords and subcommands .......... 57
GLOBAL command keywords and subcommands .......... 58
PFPSYSIN DD statement ................................ 59
  PFPSYSIN commands and subcommands for PFPMAIN ......................... 60
  PFPSYSIN keywords for PFPMAIN ................................ 61
  PFPSYSIN commands and subcommands for PFCMAIN ......................... 73
  PFPSYSIN keywords for PFCMAIN ................................ 74
  PFPSYSIN commands and subcommands for PFPEPR00 ......................... 80
Fast Path/EP operator interface commands ............... 83
Command language description format ................... 84
ACCESS ............................................. 85
ACTIVITY_FILECTL ................................. 87
ACTUATE ........................................... 89
ADD ............................................... 90
ADDN ............................................. 91
ADSN ............................................. 92
ALLOCATE ........................................ 94
ANALYZE .......................................... 95
AREA_KEY ......................................... 96
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTRACT_FORMAT</td>
<td>153</td>
</tr>
<tr>
<td>FIELDS</td>
<td>154</td>
</tr>
<tr>
<td>FLOWER_BOX</td>
<td>155</td>
</tr>
<tr>
<td>FORCE</td>
<td>156</td>
</tr>
<tr>
<td>FORMAT</td>
<td>157</td>
</tr>
<tr>
<td>FRAGMENTATION_PERCENT</td>
<td>158</td>
</tr>
<tr>
<td>FREESPACE_ANALYSIS</td>
<td>159</td>
</tr>
<tr>
<td>FREESPACE_DOVF_IOVF</td>
<td>160</td>
</tr>
<tr>
<td>FREESPACE_RAA_DOVF</td>
<td>161</td>
</tr>
<tr>
<td>FREESPACE_RAA_IOVF</td>
<td>162</td>
</tr>
<tr>
<td>FULLSEG</td>
<td>163</td>
</tr>
<tr>
<td>GENMAX</td>
<td>164</td>
</tr>
<tr>
<td>GSGNAME</td>
<td>164</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>165</td>
</tr>
<tr>
<td>GROUP_KEY</td>
<td>167</td>
</tr>
<tr>
<td>HELP</td>
<td>167</td>
</tr>
<tr>
<td>HISTORY_DDNAME</td>
<td>168</td>
</tr>
<tr>
<td>IAREA</td>
<td>170</td>
</tr>
<tr>
<td>IC</td>
<td>172</td>
</tr>
<tr>
<td>ICACHE</td>
<td>174</td>
</tr>
<tr>
<td>ICJCL</td>
<td>175</td>
</tr>
<tr>
<td>ID</td>
<td>176</td>
</tr>
<tr>
<td>IDCAMS_OPTION</td>
<td>177</td>
</tr>
<tr>
<td>IFP_ACCOUNT</td>
<td>178</td>
</tr>
<tr>
<td>IFP_JOBNAME</td>
<td>178</td>
</tr>
<tr>
<td>IFP_LIMIT</td>
<td>179</td>
</tr>
<tr>
<td>IFP_PROCNAME</td>
<td>180</td>
</tr>
<tr>
<td>IMAGECOPY</td>
<td>181</td>
</tr>
<tr>
<td>INCLUDE</td>
<td>182</td>
</tr>
<tr>
<td>INDEX</td>
<td>183</td>
</tr>
<tr>
<td>INDEX_THREADS</td>
<td>183</td>
</tr>
<tr>
<td>INITIALIZE</td>
<td>185</td>
</tr>
<tr>
<td>INPUT_DSN_MASK</td>
<td>185</td>
</tr>
<tr>
<td>INPUT_THREADS</td>
<td>188</td>
</tr>
<tr>
<td>INSERT_LIMIT_COUNT</td>
<td>190</td>
</tr>
<tr>
<td>IOVF_FREESPACE_PERCENT</td>
<td>190</td>
</tr>
<tr>
<td>IOVF_LOAD_HWM</td>
<td>191</td>
</tr>
<tr>
<td>IOVF_SAVE_THRESHOLD</td>
<td>192</td>
</tr>
<tr>
<td>IOVF_SPACE_ANALYSIS</td>
<td>193</td>
</tr>
<tr>
<td>IOVF_USED_PERCENT</td>
<td>194</td>
</tr>
<tr>
<td>IX</td>
<td>195</td>
</tr>
<tr>
<td>LANGUAGE</td>
<td>196</td>
</tr>
<tr>
<td>LARGEST_DATABASE_RECORDS</td>
<td>197</td>
</tr>
<tr>
<td>LIKE</td>
<td>198</td>
</tr>
<tr>
<td>LINE_COUNT</td>
<td>200</td>
</tr>
<tr>
<td>LIST</td>
<td>200</td>
</tr>
<tr>
<td>LIST_OPTIONS</td>
<td>201</td>
</tr>
<tr>
<td>LOADCTL</td>
<td>202</td>
</tr>
<tr>
<td>LOCATION</td>
<td>203</td>
</tr>
<tr>
<td>Topic</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>MESSAGE_LEVEL</td>
<td>204</td>
</tr>
<tr>
<td>MESSAGE_LIMIT</td>
<td>205</td>
</tr>
<tr>
<td>MESSAGE_NUMBER</td>
<td>206</td>
</tr>
<tr>
<td>MESSAGE_OVERRIDE</td>
<td>207</td>
</tr>
<tr>
<td>MESSAGE_SUPPRESSION</td>
<td>208</td>
</tr>
<tr>
<td>MGMTCLAS</td>
<td>209</td>
</tr>
<tr>
<td>MGMTCLAS2</td>
<td>210</td>
</tr>
<tr>
<td>MODEL_DDNAME</td>
<td>211</td>
</tr>
<tr>
<td>MODIFY</td>
<td>212</td>
</tr>
<tr>
<td>MONITOR</td>
<td>213</td>
</tr>
<tr>
<td>NOTIFICATION</td>
<td>214</td>
</tr>
<tr>
<td>OAREA</td>
<td>215</td>
</tr>
<tr>
<td>OBJECT</td>
<td>217</td>
</tr>
<tr>
<td>OCACHE</td>
<td>218</td>
</tr>
<tr>
<td>OFILECTL</td>
<td>219</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>220</td>
</tr>
<tr>
<td>ORPHANED_SDEP_MSG</td>
<td>222</td>
</tr>
<tr>
<td>OUTAGE_WINDOW</td>
<td>223</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>225</td>
</tr>
<tr>
<td>OUTPUT_DSN_MASK</td>
<td>226</td>
</tr>
<tr>
<td>OUTPUT_THREADS</td>
<td>229</td>
</tr>
<tr>
<td>OVERRIDE</td>
<td>230</td>
</tr>
<tr>
<td>PERFORM</td>
<td>231</td>
</tr>
<tr>
<td>PFPSORT</td>
<td>231</td>
</tr>
<tr>
<td>PLAN_FILECTL</td>
<td>232</td>
</tr>
<tr>
<td>POINTER_ANALYSIS</td>
<td>234</td>
</tr>
<tr>
<td>POINTER_VALIDATION</td>
<td>235</td>
</tr>
<tr>
<td>PREOPEN</td>
<td>237</td>
</tr>
<tr>
<td>PREPARE</td>
<td>238</td>
</tr>
<tr>
<td>PROCESS_AREA</td>
<td>239</td>
</tr>
<tr>
<td>PROCESS_EPR</td>
<td>240</td>
</tr>
<tr>
<td>PRODUCT</td>
<td>241</td>
</tr>
<tr>
<td>PRODUCT_LIMIT</td>
<td>243</td>
</tr>
<tr>
<td>RAA_FREESPACE_PERCENT</td>
<td>244</td>
</tr>
<tr>
<td>RAP_OVERFLOW_PERCENT</td>
<td>245</td>
</tr>
<tr>
<td>RAP_VALIDATION</td>
<td>246</td>
</tr>
<tr>
<td>RECORD_IO_AVERAGE</td>
<td>248</td>
</tr>
<tr>
<td>RECORD_IO_MAXIMUM</td>
<td>249</td>
</tr>
<tr>
<td>RECORD_IOVF_PERCENT</td>
<td>250</td>
</tr>
<tr>
<td>RECORD_LENGTH_ANALYSIS</td>
<td>251</td>
</tr>
<tr>
<td>RECORD_LENGTH_INCREMENT</td>
<td>252</td>
</tr>
<tr>
<td>RECORD_PLACEMENT_ANALYSIS</td>
<td>252</td>
</tr>
<tr>
<td>RECORD_PROFILE_ANALYSIS</td>
<td>253</td>
</tr>
<tr>
<td>RECOVJCL</td>
<td>254</td>
</tr>
<tr>
<td>RECOVPD</td>
<td>255</td>
</tr>
<tr>
<td>RECVJCL</td>
<td>256</td>
</tr>
<tr>
<td>REGISTER</td>
<td>257</td>
</tr>
<tr>
<td>RELOAD</td>
<td>258</td>
</tr>
<tr>
<td>REORGANIZE</td>
<td>259</td>
</tr>
</tbody>
</table>
Chapter 3  Expression syntax  355

Introduction to expressions .............................................. 356
   Using expressions with online and offline data extract ............. 356
   Using expressions with DEDB CHANGE, UNLOAD, and RELOAD ...... 358
   Using expressions with DEDB online restructure .................... 359
Operands ........................................................................... 360
   Literal ........................................................................... 361
   Field variable .................................................................. 362
   Built-in variable ........................................................... 363
   Function ......................................................................... 365
   Descriptions of valid functions .......................................... 366
   Assignment variable ....................................................... 369
Data conversions ............................................................... 371
### Data-type
- Size ........................................ 371
- Precision .................................. 373

### Operators
- Precedence of operators .................. 374
- Assignment operator ....................... 375
- Boolean operator ......................... 376
- Comparison operator ...................... 376
- Numeric operator .......................... 377
- Conditional operator ...................... 379

### Diagnosing problems

---

### Chapter 4  DE DB reports

DE DB analysis reports .................................. 382
- Area Summary Report .......................... 383
- Free Space Analysis Report ................ 386
- IOVF Space Analysis Report ............... 393
- UOW Detailed Analysis Report ............ 394
- Pointer Analysis Report ..................... 396
- Segment I/O Analysis Report .............. 398
- Segment Length Analysis Report .......... 401
- Segment Placement Analysis Report ...... 403
- Record Length Analysis Report .......... 404
- Record Placement Analysis Report ...... 408
- Record Profile Analysis Report .......... 410
- Synonym Chain Analysis Report .......... 413

DE DB Unload Reports .................................. 417
- Unload Input Area Summary Report ....... 417
- Unload Output Area Summary Report .... 421
- Unload Database Summary Report ........ 425

DE DB Reload Reports .................................. 426
- Reload Input Area Summary Report ...... 427
- Reload Output Area Summary Report ...... 430
- Reload Database Summary Report ........ 434

DE DB Extend Report .................................. 435
- Extend Area Summary Report ............. 436

---

### Chapter 5  Supporting utilities

Available utilities .................................. 442
Area Change Modeling Utility ................ 443
- Utility program modules .................. 444
- Input requirements ......................... 445
- Randomizer considerations ............... 446
- Process flow .................................. 447
- PFMD0100 JCL requirements ............... 449
- PFMD0100 control statements .......... 451
- PFMD0100 return codes ................. 455
- SORT13 .................................. 455
<table>
<thead>
<tr>
<th>Reference</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDEP Space Utilization Utility</td>
<td>489</td>
</tr>
<tr>
<td>Randomizer Interface Subroutine</td>
<td>482</td>
</tr>
<tr>
<td>File Sort Utility (PFPSORT command)</td>
<td>477</td>
</tr>
<tr>
<td>Control Interval Dump and Modification Utility (PROCESS_AREA command)</td>
<td>468</td>
</tr>
<tr>
<td>DMAC Print utility (DMAC_PRINT command)</td>
<td>465</td>
</tr>
<tr>
<td>Process flow</td>
<td>491</td>
</tr>
<tr>
<td>Creating and initializing the SDEP history VSAM data set</td>
<td>490</td>
</tr>
<tr>
<td>Randomizer interface subroutine program example</td>
<td>487</td>
</tr>
<tr>
<td>SDEP Space Utilization Utility</td>
<td>489</td>
</tr>
<tr>
<td>Utility program modules</td>
<td>489</td>
</tr>
<tr>
<td>Creating and initializing the SDEP history VSAM data set</td>
<td>490</td>
</tr>
<tr>
<td>Process flow</td>
<td>491</td>
</tr>
<tr>
<td>PFSD0700 JCL requirements</td>
<td>493</td>
</tr>
<tr>
<td>PFSD0700 control statements</td>
<td>493</td>
</tr>
<tr>
<td>PFSD0800 JCL requirements</td>
<td>494</td>
</tr>
<tr>
<td>PFMD0300 JCL requirements</td>
<td>455</td>
</tr>
<tr>
<td>PFMD0300 return codes</td>
<td>457</td>
</tr>
<tr>
<td>SORT12</td>
<td>457</td>
</tr>
<tr>
<td>SORT3</td>
<td>458</td>
</tr>
<tr>
<td>PFMD0500 JCL requirements</td>
<td>459</td>
</tr>
<tr>
<td>PFMD0500 return codes</td>
<td>460</td>
</tr>
<tr>
<td>Area Change Modeling Utility sample JCL</td>
<td>460</td>
</tr>
<tr>
<td>User procedures</td>
<td>462</td>
</tr>
<tr>
<td>User procedures for full-function databases</td>
<td>463</td>
</tr>
<tr>
<td>Comparing randomizing routines</td>
<td>464</td>
</tr>
<tr>
<td>Area Change Modeling Utility sample scenarios</td>
<td>465</td>
</tr>
<tr>
<td>File Sort Utility sample scenarios</td>
<td>482</td>
</tr>
<tr>
<td>File Sort Utility DD statements</td>
<td>480</td>
</tr>
<tr>
<td>Using SORT_OPTION keyword</td>
<td>480</td>
</tr>
<tr>
<td>Specifying the sort sequence</td>
<td>479</td>
</tr>
<tr>
<td>Dynamically allocating input and output</td>
<td>479</td>
</tr>
<tr>
<td>Selecting the database and areas</td>
<td>478</td>
</tr>
<tr>
<td>PFPSORT keywords</td>
<td>478</td>
</tr>
<tr>
<td>Selecting the database and areas</td>
<td>478</td>
</tr>
<tr>
<td>Dynamically allocating input and output</td>
<td>479</td>
</tr>
<tr>
<td>Specifying the sort sequence</td>
<td>479</td>
</tr>
<tr>
<td>Using SORT_OPTION keyword</td>
<td>480</td>
</tr>
<tr>
<td>File Sort Utility DD statements</td>
<td>480</td>
</tr>
<tr>
<td>File Sort Utility sample scenarios</td>
<td>482</td>
</tr>
<tr>
<td>Randomizer Interface Subroutine</td>
<td>482</td>
</tr>
<tr>
<td>PFUT0B50 JCL requirements</td>
<td>483</td>
</tr>
<tr>
<td>PFUT0B50 parameter lists</td>
<td>484</td>
</tr>
<tr>
<td>PFUT0B50 initialization function</td>
<td>484</td>
</tr>
<tr>
<td>PFUT0B50 calculation function</td>
<td>485</td>
</tr>
<tr>
<td>Randomizer module interface exceptions</td>
<td>485</td>
</tr>
<tr>
<td>Randomizer calculations</td>
<td>486</td>
</tr>
<tr>
<td>Randomizer interface subroutine program example</td>
<td>487</td>
</tr>
<tr>
<td>SORT12</td>
<td>457</td>
</tr>
<tr>
<td>PFMD0300 return codes</td>
<td>457</td>
</tr>
<tr>
<td>PFMD0300 JCL requirements</td>
<td>457</td>
</tr>
<tr>
<td>PFMD0700 JCL requirements</td>
<td>457</td>
</tr>
<tr>
<td>PFMD0700 control statements</td>
<td>457</td>
</tr>
<tr>
<td>PFMD0800 JCL requirements</td>
<td>457</td>
</tr>
</tbody>
</table>
Chapter 6  Program extensions 501

FABEUR6 Reload File Create utility ........................................ 502
FABEUR6 overview .................................................................. 502
FABEUR6 parameter lists ...................................................... 503
FABEUR6 JCL requirements .................................................... 507
FABEUR6 utility control statements ........................................ 509
FABEUR7 Read Unloaded Database utility ............................... 514
FABEUR7 overview ................................................................. 514
FABEUR7 parameter lists ...................................................... 515
FABEUR7 JCL requirements .................................................... 521
FABEUR7 utility control statement for HD Unload input .......... 522
FABEUR7 utility control statements for non-HD Unload input ... 523

Appendix A  History file record layout 535

Appendix B  DEDB Data Extract record layout 537

Appendix C  DEDB unload/reload record layouts 541

Appendix D  Discard file record layout 547

Appendix E  Command syntax diagrams 551
ANALYZE command (continued) ........................................ 554
  Subcommands available with ANALYZE ................................ 554
BACKOUT command ..................................................... 555
  Subcommands available with BACKOUT ................................ 555
BUILD command ....................................................... 556
  Subcommand available with BUILD .................................... 556
CHANGE command ...................................................... 557
CHANGE command (continued) ........................................ 558
  Subcommands available with CHANGE .................................. 559
EXTEND command ....................................................... 560
  Subcommand available with EXTEND .................................... 561
EXTRACT command ..................................................... 563
  Subcommands available with EXTRACT ................................ 564
GLOBAL command ..................................................... 565
GLOBAL command (continued) ........................................ 566
  Subcommands available with GLOBAL .................................. 567
IMAGECOPY command ................................................ 568
  Subcommands available with IMAGECOPY ................................ 569
INITIALIZE command ................................................. 570
  Subcommands available with INITIALIZE ................................. 570
PFPSORT command ................................................... 571
PREPARE command ................................................... 572
  Subcommand available with PREPARE ................................ 573
  Subcommands available with PREPARE ................................ 574
PROCESS_AREA command ............................................ 574
  Subcommand available with PROCESS_AREA ................................ 575
RELOAD command ..................................................... 576
RELOAD command (continued) ....................................... 577
  Subcommands available with RELOAD ................................ 578
REORGANIZE command ............................................... 579
REORGANIZE command (continued) ................................ 580
  Subcommands available with REORGANIZE ................................ 580
RESTART command ................................................... 581
  Subcommand available with RESTART ................................ 581
RESTRUCTURE command ............................................. 582
  Subcommand available with RESTRUCTURE ................................ 583
RESYNC command .................................................... 584
  Subcommand available with RESYNC .................................. 584
RETRIEVE command .................................................. 585
  Subcommand available with RETRIEVE ................................ 585
  Subcommands available with RETRIEVE ................................ 586
SHADOW_INIT command .............................................. 586
  Subcommands available with SHADOW_INIT ................................ 586
STATUS command .................................................... 587
  Subcommand available with STATUS ................................... 587
UNLOAD command ..................................................... 588
UNLOAD command (continued) ...................................... 589
  Subcommands available with UNLOAD .................................. 590
  Subcommands available with UNLOAD (continued) .................. 591
VERIFY command ...................................................... 591
Subcommand available with VERIFY ............................................. 592
XSCAN command ................................................................. 592
Subcommand available with XSCAN .......................................... 593

Appendix F Subcommand syntax diagrams .................................. 595
Subcommand functions .......................................................... 596
ACTIVITY_FILECTL subcommand .......................................... 598
ALLOCATE subcommand ...................................................... 599
ALLOCATE subcommand (continued) ....................................... 600
CORRECTIONS_FILECTL subcommand .................................... 600
DISCARD_FILECTL subcommand .......................................... 601
EXCLUDE subcommand ....................................................... 602
INCLUDE subcommand ........................................................ 602
IC subcommand ................................................................. 603
IX subcommand ................................................................. 604
LOADCTL subcommand ...................................................... 604
OFILECTL subcommand .................................................... 605
OUTPUT subcommand ....................................................... 606
PERFORM subcommand ...................................................... 606
PLAN_FILECTL subcommand .............................................. 607
REGISTER subcommand ..................................................... 608
REPORT subcommand ....................................................... 609
THRESHOLD subcommand ................................................... 610
USER_RECORD subcommand .............................................. 611
SPACE keyword ............................................................... 611

Appendix G Repository maintenance command and subcommand syntax diagrams 613
Subcommand functions ........................................................ 613
PROCESS_EPR command .................................................... 615
Subcommands available with PROCESS_EPR ............................ 615
ADD subcommand .............................................................. 616
ADD subcommand (continued) ............................................. 617
DELETE subcommand ........................................................ 618
LIST subcommand ............................................................. 619
MODIFY subcommand ........................................................ 620
MODIFY subcommand (continued) ...................................... 621
OVERRISE subcommand .................................................... 622
RESET subcommand ........................................................ 622
RETRIEVE subcommand .................................................... 623
Subcommand available with RETRIEVE .................................. 623

Appendix H Sample utility and command scenarios 625
How to interpret the scenarios .................................................. 626
JCL and control statement .................................................... 626
Descriptive text ................................................................. 626
Sample library JCL ............................................................. 627
Segment hierarchy for sample DEDB ........................................ 627
DBD for sample DEDB ....................................................... 627
Scenario conceptual descriptions ........................................... 628
Expressions ........................................................................ 631
  Customize unload output using expressions ........................ 631
  Customize offline extract output using expressions ............ 633
Control Interval Dump and Modification Utility .................... 636
  Use the SNAP function to dump control interval ............... 636
  Repair pointer values online using VER and REP functions .... 637
  Print fields within in-core DMAC ..................................... 638
Area Change Modeling Utility ............................................... 638
  Model effects of changing randomizer for all areas ............ 639
  Model effects of changing UOW and ROOT values for selected areas . 640
  Model effects of changing control interval size for an area and control segment placement ...................... 641
  Model effects of converting from full-function to DEDB format .. 642
File Sort Utility .................................................................. 643
  Sort file in ascending order by root key ............................. 643
  Sort file in ascending order by RAP and sort logical SDEPs in reverse sequence ........................................ 644
Figures

EXEC statement for online execution .................................................. 30
Example of JCL for online execution .................................................. 30
EXEC statement for offline execution .................................................. 31
Example of JCL for offline execution .................................................. 31
EXEC statement for BMP execution ..................................................... 31
Example of JCL for BMP execution ..................................................... 32
EXEC statement for available 64-bit storage ...................................... 32
Sample DBD for 2048-area database .................................................. 36
Using IAREAxxx and IARxxxxx on CHANGE command .......................... 36
DBD for 2048-area database ............................................................. 41
Using OAREAxxx and OARxxxxx on UNLOAD command ....................... 41
Sample Fast Path/EP general JCL ...................................................... 56
Example operator interface command ................................................ 83
Area Summary Report ........................................................................ 383
Free Space Analysis Report ............................................................... 387
IOVF Space Analysis Report ............................................................... 393
UOW Detailed Analysis Report .......................................................... 394
Pointer Analysis Report ..................................................................... 397
Segment I/O Analysis Report ............................................................... 398
Segment Length Analysis Report ........................................................ 401
Segment Placement Analysis Report .................................................... 403
Record Length Analysis Report .......................................................... 405
Record Placement Analysis Report ..................................................... 408
Record Profile Analysis Report ........................................................... 411
Synonym Chain Analysis Report ........................................................ 413
Unload Input Area Summary Report ................................................... 418
Unload Output Area Summary Report ............................................... 421
Unload Database Summary Report ..................................................... 425
Reload Input Area Summary Report ...................................................... 427
Reload Output Area Summary Report ................................................ 431
Reload Database Summary Report ...................................................... 434
Extend Area Summary Report ............................................................. 436
Area Change Modeling Utility process flow ......................................... 448
PFMD0100 Control Statement Syntax Diagram .................................... 452
Sample Area Change Modeling Utility JCL ......................................... 460
Using the offline DMAC_PRINT command to print DMAC blocks for all areas ...................................................... 467
Using the online DMAC_PRINT command to print DMAC blocks for specified areas ...................................................... 467
Sample control statement for processing all areas ................................. 470
Sample control statement for processing specific areas .......................... 470
JCL example for offline area data set input ................................................. 472
JCL example for image copy input .............................................................. 472
JCL example for online area data set input ............................................... 472
Syntax Diagram for Control Interval Dump and Modification Functions .......... 474
Conversion formula for RAP to RBA ......................................................... 487
Randomizer interface subroutine sample program .................................. 487
Create SDEP history VSAM data set JCL ................................................... 490
SDEP update and reporting process flow .................................................. 492
Data extracted from PFPDBM01 ................................................................. 493
Sample SDEP update and report generation JCL ...................................... 497
Sample JCL for generating SDEP utilization reports for all areas of database
    PFPDBM01 ............................................................................................. 498
Reload File Create user exit sample ......................................................... 505
Reload File Create process ........................................................................ 507
Sample FABEUR6 utility control statements ......................................... 511
COBOL Example ...................................................................................... 511
Read unloaded database example user exit ............................................ 519
Sample FABEUR7 utility control statement for HD Unload input .......... 523
Sample JCL for FABEUR7 (with HD Unload input) ................................... 523
Sample FABEUR7 utility control statement for non-HD Unload input ..... 524
Sample JCL for FABEUR7 (non-HD Unload input) ................................... 524
Sample COBOL for extracting date for trial balance report .................... 524
Sample JCL for user-written program calling FABGXDR ....................... 530
Sample COBOL program using FABGXDR ............................................. 531
Sample JCL (#LKEDCAL) ......................................................................... 533
JCL and control statement ..................................................................... 626
Segment hierarchy for sample DEDB ....................................................... 627
DBD for sample DEDB ........................................................................... 627
JCL to customize unload output using expressions ................................ 631
JCL to customize offline extract output using expressions ...................... 633
JCL to use the SNAP function to dump control interval ....................... 636
JCL to repair pointer values online using VER and REP functions .......... 637
JCL to print fields within in-core DMAC .................................................. 638
JCL to model effects of changing randomizer for all areas ..................... 639
JCL to model effects of changing UOW and ROOT values for selected areas . 640
JCL to model effects of changing control interval size for an area and control segment placement ................................................................. 641
JCL to model effects of converting from full-function to DEDB format ..... 642
JCL to sort files in ascending order by root key ..................................... 643
JCL to sort files in ascending order by RAP and sort logical SDEPs in reverse sequence ................................................................................. 645
# Tables

Disposition of dynamically allocated area data set for Fast Path/EP commands ........................................ 35
OPTIONS command keywords ............................................. 57
OVERRIDE and RESET subcommands and keywords .................................................. 58
GLOBAL command keywords .................................................. 59
IC, REPORT, and THRESHOLD subcommands .............................................. 59
PFPSYSIN commands and subcommands for PFPMAIN ............................................. 60
PFPSYSIN commands and keywords for PFPMAIN ............................................. 60
PFPSYSIN commands and subcommands for PFCMAIN ............................................. 73
PFPSYSIN commands and keywords for PFCMAIN ............................................. 74
PFPSYSIN commands and subcommands for PFPEPR00 allocation records ............ 80
PFPSYSIN commands and subcommands for PFPEPR00 statistics records ............. 80
PFPSYSIN commands and subcommands for PFPEPR00 message customization .... 83
Fast Path/EP operator interface commands and keywords ..................................... 83
Available expressions for customizing segment output for data extracts .................. 357
Available expressions for customizing output record for USER format data extracts ........................................... 357
Available expressions for customizing DEDB CHANGE, UNLOAD, and RELOAD processes ........................................................................................................... 358
Available expressions for customizing DEDB online restructure processes ............. 360
Valid data-types used with operands ...................................................................... 360
Programming language data-type equivalents ....................................................... 361
Built-in variables used with expressions .............................................................. 364
Format and example of RUNDATE, TODAY, RUNTIME and RUNDATETIME built-in variables ........................................................................................................... 365
Functions used with expressions ............................................................................ 365
Valid substitution masks for string operands ......................................................... 370
Size of operand after conversion ........................................................................... 372
Size of target (source type=F or X; target type=P) ................................................ 372
Operators used with expressions ............................................................................ 374
Target data-type for operations with mixed operands (used with comparison operator or numeric operator) ........................................................................... 377
Fast Path Analyzer/EP reports and generation keywords ....................................... 382
Area Summary Report fields ................................................................................. 383
RRA fields ......................................................................................... 384
IOVF fields ..................................................................................... 384
SDEP portion fields ..................................................................................... 385
RBA values portion fields ..................................................................................... 385
Free Space Analysis Report fields ......................................................................... 387
Area overview fields ......................................................................................... 388
UOW range fields ......................................................................................... 389
Descriptive text for JCL to sort files in ascending order by RAP and sort logical SDEPs in reverse sequence .......................... 645
About this book

The Fast Path/EP Series of products provides tools for database administrators and technical support personnel involved in the management, maintenance, and performance tuning of DEDB databases. These tools deliver a wide range of functions for completing DEDB reorganizations and restructures, managing DEDB space utilization, and creating and maintaining indexes to DEDBs. The Fast Path/EP Series consists of the following products, all which are discussed in this book:

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

The Fast Path/EP Series command language consists of commands, subcommands, and keywords which can be used in various combinations to control desired product functionality. Additional supporting utilities and program extensions provide customized functionality that extends the benefits of the products beyond day-to-day maintenance.

This book serves as a detailed reference for making all of these product components work for you.

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

The following example illustrates the syntax for a DELETE statement. Because the FROM keyword, alias variable, and WHERE clause are optional, they appear below the main command line. In contrast, the tableName variable appears on the command line because the table name is required. If the statement includes a WHERE clause, the clause must contain a search condition or a CURRENT OF clause. (The searchCondition variable appears on the main line for the WHERE clause, indicating that this choice is required.)
The following guidelines provide additional information about syntax diagrams:

- Read diagrams from left to right and from top to bottom.

- A recursive (left-pointing) arrow above a stack indicates that you may choose more than one item in the stack.

- An underlined item is a default option.

- If a diagram shows punctuation marks, parentheses, or similar symbols, you must enter them as part of the syntax. Asterisks are exceptions. An asterisk in a diagram indicates a reference note.

- In general, IBM MVS™ commands, keywords, clauses, and data types are displayed in uppercase letters. However, if an item can be shortened, the minimum portion of the MVS command or keyword might be displayed in uppercase letters with the remainder of the word in lowercase letters (for example, CANcel).

- The following conventions apply to variables in syntax diagrams:
  
  — Variables typically are displayed in lowercase letters and are always italicized.
  
  — If a variable is represented by two or more words, initial capitals distinguish the second and subsequent words (for example, databaseName).

**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).
Chapter 1: JCL statements

This chapter discusses the JCL requirements for running the Fast Path/EP Series products. Specifically, it provides the following information:

- Introduction ................................................................. 29
- EXEC statement .......................................................... 30
  - Online EXEC statement ............................................. 30
  - Offline EXEC statement ............................................. 31
  - BMP EXEC statement ............................................... 31
  - 64-bit storage EXEC statement for offline processing ........ 32
- DD statements ........................................................... 33

Introduction

Depending on the desired database maintenance or analysis function, Fast Path/EP Series products operate in different modes. For more information about these operating modes as they relate to a specific BMC Fast Path product, see the appropriate user guide.

- Fast Path Offline Suite User Guide
- Fast Path Online Restructure/EP User Guide
- Fast Path Online Suite User Guide

JCL requirements are different for the different operating modes. This chapter discusses the JCL requirements for executing the Fast Path/EP Series products in offline and online modes.

EXEC statement

Fast Path/EP Series products use the EXEC statement for execution. The EXEC statement differs according to operating mode.

Online EXEC statement

The EXEC statement for Fast Path/EP online execution must be in the form shown in Figure 1.

**Figure 1   EXEC statement for online execution**

```
//PFP EXEC PGM=DFSRRC00,REGION=0M,
     PARM=(IFP,dbname,DF#FPU0)
```

The first, second, and third subparameters of the PARM parameter are required. Other subparameters might be required for the IFP regions at your installation. The standard catalogued procedure supplied with IMS (FPUTIL) can be used for Fast Path/EP Series products.

Using the online ANALYZE command as an example, the JCL for Fast Path/EP online execution is shown in Figure 2.

**Figure 2   Example of JCL for online execution**

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

**NOTE**

Only one DEDB can be processed per JOB step, although multiple areas of that DEDB can be processed.
Offline EXEC statement

The EXEC statement for Fast Path/EP offline execution is shown in Figure 3:

Figure 3  EXEC statement for offline execution

```
//PFP      EXEC PGM=PFPMAIN,REGION=0M
```

Virtual storage requirements for Fast Path/EP Series products vary depending on the number of parallel tasks you are running and whether DBRC is active. A region size of 0M indicates to use all available storage. If you specify a region size other than 0M, and the value is insufficient for processing, an error might occur. If processing fails, adjust the region size in your JCL for higher virtual storage requirement.

The JCL for Fast Path/EP offline execution is shown in Figure 4. Dynamic allocation of the area data set and the ACB library is assumed.

Figure 4  Example of JCL for offline execution

```
//PFP      EXEC PGM=PFPMAIN,REGION=0M
//STEPLIB  DD   DSN=BMC.PFP.LOAD,DISP=SHR
//                   DD   DSN=IMS.RESLIB,DISP=SHR
//PFPSYSIN DD   *
   ANALYZE DBD=dbname,AREA=areaname
/*
```

BMP EXEC statement

The BMP operating mode is used only by certain commands associated with the online maintenance of index databases using the Fast Path Indexer/EP product. The EXEC statement for BMP execution must be in the following form:

Figure 5  EXEC statement for BMP execution

```
//PFP      EXEC PGM=DFSRRC00,REGION=0M,
//                   PARM=(BMP,PFPMAIN,psbname)
```

The first, second, and third subparameters of the PARM parameter are required. Other subparameters might be required for the BMP regions at your installation.

Using the index RESYNC command as an example, the JCL for Fast Path Indexer/EP BMP execution is shown in Figure 6.
64-bit storage EXEC statement for offline processing

When you run the offline Analyze function with any of the following Fast Path/EP keywords, ensure you have an adequate amount of 64-bit storage available:

- `POINTER_VALIDATION=FULL`
- `SDEP_VALIDATION=FULL`
- `RAP_VALIDATION=(XREF,...)`

The offline Analyze function uses 64-bit storage for some of its internal data elements. 64-bit storage is allocated in storage segments that are 1 megabyte each. The default number of available segments depends on the IBM z/OS® MEMLIMIT parameter set by the z/OS systems programmer. You can specify a different value by including the MEMLIMIT parameter on the EXEC statement in the JCL. You can specify the amount of 64-bit storage that is available to the job step in 1 megabyte increments.

Figure 7 provides an example EXEC statement for specifying available 64-bit storage.

Figure 7  EXEC statement for available 64-bit storage

```
//jobName EXEC PGM=PFPPMAIN,...MEMLIMIT=nnnnM,...
```

If the offline Analyze function requires more storage than the MEMLIMIT default or the amount you specified on the MEMLIMIT parameter, it does not allocate 64-bit storage but uses traditional 31-bit storage instead.
DD statements

This section describes the offline, online and BMP JCL statements that you can use to run Fast Path/EP jobs. Fast Path/EP Series products use the following DD statements:

**AMSOUT DD**

Optional for the Restructure and Restart functions. The data set that is named in the AMSOUT DD statement contains the restart information that is required to restart post-processing tasks that failed. You can specify a partitioned data set, but BMC recommends that you specify a PDSE. The member name is the name of the restructured database. The AMSOUT data set can be used by all jobs that execute the Restructure and Restart functions.

**NOTE**

Do not modify the AMSOUT data set. Modifying the data set might make the information unusable when executing the Restart function.

The logical record length (LRECL) should be 80. The record format (RECFM) should be FB. The block size (BLKSIZE) can be any value that is appropriate for the LRECL and RECFM. The disposition (DISP) should be SHR.

For more information about using the AMSOUT DD statement, see the *Fast Path Online Restructure/EP User Guide*.

**areaname DD**

Required for offline mode; not used for online or BMP mode. The areaname DD statement identifies the offline area data set to be processed by Fast Path/EP. If the areaname DD statement is omitted from the JCL, Fast Path/EP attempts to dynamically allocate it.

The areaname DD statement works with Fast Path/EP commands as follows:

- **input area** – The areaname DD statement identifies the area data set used as input to the following command functions:
  - ANALYZE
  - BUILD
  - DMAC_PRINT
  - EXTRACT
  - UNLOAD
  - VERIFY
DD statements

- **output area** – The areaname DD statement identifies the area data set used as output from the following command functions:
  
  — CHANGE
  — RELOAD

- **input/output area** – The areaname DD statement identifies the area data set used as both input and output to the following command functions:
  
  — EXTEND
  — INITIALIZE
  — PROCESS_AREA
  — REORGANIZE

The database definition (DBD) is in the ACB library identified by the IMSACB DD statement.

If DBRC is active, and the area is registered with DBRC, and the areaname DD statement refers to an area data set, then the areaname DD statement data set must match the registered area data set name.

The areaname DD statement can identify an image copy data set for the following command functions:

- ANALYZE
- DMAC_PRINT
- EXTRACT
- PROCESS_AREA
- UNLOAD

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

- If the INPUT_DSN_MASK keyword is specified, it is used to generate the data set name for the ANALYZE, BUILD, DMAC_PRINT, EXTEND, INITIALIZE, PROCESS_AREA, REORGANIZE, UNLOAD and VERIFY command functions.

- If the OUTPUT_DSN_MASK keyword is specified, it is used to generate the data set name for the CHANGE and RELOAD command functions.

- If DBRC is active and the area is registered, the registered area data set name is obtained from DBRC.

- The STEPLIB is searched for the DFSMDA member that contains the data set name for this area.
The disposition of the dynamically allocated area data set depends on the function that is processed, as shown in Table 1:

Table 1  Disposition of dynamically allocated area data set for Fast Path/EP commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Disposition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANALYZE</td>
<td>DISP=SHR</td>
</tr>
<tr>
<td>BUILD (inputs)</td>
<td>DISP=SHR</td>
</tr>
<tr>
<td>CHANGE (outputs)</td>
<td>DISP=OLD</td>
</tr>
<tr>
<td>DMAC_PRINT</td>
<td>DISP=SHR</td>
</tr>
<tr>
<td>EXTEND</td>
<td>DISP=OLD</td>
</tr>
<tr>
<td>EXTRACT</td>
<td>DISP=SHR</td>
</tr>
<tr>
<td>INITIALIZE</td>
<td>DISP=OLD</td>
</tr>
<tr>
<td>PROCESS_AREA</td>
<td>DISP=OLD</td>
</tr>
<tr>
<td>RELOAD (outputs)</td>
<td>DISP=OLD</td>
</tr>
<tr>
<td>REORGANIZE</td>
<td>DISP=OLD</td>
</tr>
<tr>
<td>UNLOAD (inputs)</td>
<td>DISP=SHR</td>
</tr>
<tr>
<td>VERIFY (inputs)</td>
<td>DISP=SHR</td>
</tr>
</tbody>
</table>

**IAREAxxx DD**

Used for the change and reload functions; not used for other functions.

**IARxxxxx DD**

These DD statements are used to identify the input data sets for areas to be processed by the CHANGE or RELOAD command. For the change function only, these DD statements can refer to an area data set or an image copy data set. Depending on the total number of areas in a DEDB, the respective DD statements can be used as follows:

- IAREAxxx – to identify an area number less than or equal to 999, where xxx is the area number preceded by leading zeroes (i.e., area number 4 would be IAREA004)

- IARxxxxx – to identify an area number between 1 to 2048, where xxxxx is the area number preceded by leading zeroes (i.e., area number 2048 would be IAR02048)

If either of these DD statements is omitted from the JCL, Fast Path/EP attempts to dynamically allocate the input area or areas by using the INPUT_DSN_MASK keyword. If you specify the INPUT_DSN_MASK keyword to dynamically allocate the input data set, do not include the IAREAxxx DD or IARxxxxx statement.

Figure 8 shows a sample DBD for a database containing 2048 areas. The lines that contain dots indicate that not all areas in the DBD are shown in this example.
The command set shown in Figure 9 uses IAREAxxx and IARxxxxx DD statements to identify input areas for a CHANGE command. This example shows how IAREAxxx and IARxxxxx can be used interchangeably for area numbers less than 1000.

**Figure 8  Sample DBD for 2048-area database**

```
DBD   NAME=TESTDBD,ACCESS=DEDB, RNAME=DBFHDC40
*      
ARA0001 AREA DD1=AR0001,DEVICE=3390,SIZE=8192, UOW=(24,5),ROOT=(10,4)
*      
ARA0002 AREA DD1=AR0002,DEVICE=3390,SIZE=8192, UOW=(24,5),ROOT=(10,4)
*      
ARA0999 AREA DD1=AR0999,DEVICE=3390,SIZE=8192, UOW=(25,6),ROOT=(16,10)
*      
ARA2048 AREA DD1=AR2048,DEVICE=3390,SIZE=8192, UOW=(26,6),ROOT=(10,4)
```

**Figure 9  Using IAREAxxx and IARxxxxx on CHANGE command**

```
//PFP      EXEC PGM=PFPMAIN,REGION=0M
//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
// DD DISP=SHR,DSN=IMSVS.RESLIB
// DD DISP=SHR,DSN=IMSVS.DFSMDA
//OLDACB   DD DSN=IMSVS.ACBLIB.OLD,DISP=SHR
// IMSACB   DD DSN=IMSVS.ACBLIB.NEW,DISP=SHR
//IAREA001 DD DSN=ara0001.database,DISP=SHR
//IAR00002 DD DSN=ara0002.database,DISP=SHR
//IAREA999 DD DSN=ara0999.database,DISP=SHR
//IAR02048 DD DSN=ara2048.database,DISP=SHR
//PFPSYSIN DD *
CHANGE DBD=TESTDBD, IAREA=(ARA0001,ARA0002,ARA0999,ARA2048)
OUTPUT_DSN_MASK='PFP.&DBD.&AREA'
/*
```

**IMSACB DD**

Required for offline mode; ignored in online mode. When it is allocated from the IMS control region, no dynamic allocation is provided for this DD statement. If not found in the JCL, the product searches for the MODSTAT DD statement or the OLCSTAT DD statement. For details about using the DD statements to identify the IMS ACB library, refer to the *IMS/ESA System Administration Guide*. 

NOTE
The IMS/ESA release level of the ACB library must be the same as that of the RESLIB included in the STEPLIB DD statement concatenation.

IMSACBA / IMSACBB DD

Optional for online IMS ACB data sets. The active IMS ACB data set is allocated based on the value that is specified in the MODSTAT or OLCSTAT data set. Dynamic allocation can be used for these DD statements.

IMSDALIB DD

Optional. The IMSDALIB DD statement identifies the non-APF-authorized library containing the dynamic allocation (DFSMDA) member. When a Fast Path/EP product searches for a DFSMDA member, it first searches the library supplied by using the IMSDALIB DD. If the member is not found, it then searches the STEPLIB DD for the member that contains the data set name for the area.

IMSRESLB DD

Optional for all processes. The IMSRESLB DD statement identifies the library or libraries that contain randomizer and compression routines specified in the ACB contained within the IMSACB DD statement. It also identifies the libraries that contain sparsing and partitioning subroutines specified in the registrations within the PFXLIB DD statements.

NOTE
If the IMSRESLB DD statement is specified in the JCL, then the randomizer and compression routines only are loaded directly from the IMSRESLB DD statement. If the IMSRESLB DD statement is not present in the JCL, then these routines are loaded from the STEPLIB DD statement.

Using IMSRESLB enables you to make changes to the randomizer and compression routines for use with the specified process. The following conditions apply:

- If COMPRESS=YES is specified, then the compression routine specified in IMSRESLB is used to compress segment data for the output area.

- The randomizer specified in IMSRESLB is used to write the files for all defined processes except for unload.
indexname DD

Required for offline mode; not used for online or BMP mode. The indexname DD statement identifies the offline index data set to be processed by Fast Path/EP. If the indexname DD statement is omitted from the JCL, Fast Path/EP attempts to dynamically allocate it.

The indexname DD statement works with Fast Path/EP commands as follows:

- **input index** – The indexname DD statement identifies the index data set used as input to the VERIFY command.
- **output index** – The indexname DD statement identifies the index data set used as output from the BUILD command.

The database definition (DBD) is in the ACB library identified by the IMSACB DD statement.

If DBRC is active, and the index is registered with DBRC, and the indexname DD statement refers to an index data set, then the indexname DD statement data set must match the registered index data set name.

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

- If the INPUT_DSN_MASK keyword is specified on the IX subcommand for an index verification, it is used to generate the index data set name.
- If the OUTPUT_DSN_MASK keyword is specified on the IX subcommand for an index build, it is used to generate the index data set name.
- If DBRC is active and the index is registered, the registered index data set name is obtained from DBRC.
- The STEPLIB is searched for the DFSMDA member that contains the data set name for this index.

The disposition of the dynamically allocated index data set depends on the function that is processed:

- BUILD (outputs): DISP=OLD
- VERIFY (inputs): DISP=SHR

**MODSTAT / MODSTAT2**

Optional. Used to identify the active online IMS ACB data set (IMSACBA or IMSACBB). Dynamic allocation can be used for these DD statements.
When one of these DD statements is present, the MODSTAT data set is interrogated to determine whether IMSACBA or IMSACBB is the active library. If the IMSACBA or IMSACBB DD statement is not present in the JCL, the STEPLIB/LINKLIST are searched for a DFSMDA member.

If MODSTAT2 DD is present, the active MODSTAT data set is determined prior to ACBLIB selection. The MODSTAT2 data set is interrogated to determine whether IMSACBA or IMSACBB is the active library. If the IMSACBA or IMSACBB DD statement is not present in the JCL, the STEPLIB/LINKLIST are searched for a DFSMDA member.

If both OLCSTAT and MODSTAT DD statements are present, then OLCSTAT will be used and MODSTAT will be ignored.

**NEWACB DD**

Required for the Prepare, Shadow Initialization, Restructure, and unload functions. No dynamic allocation is provided for this DD statement. The NEWACB DD statement identifies the ACB library that contains the database definition that describes the database as it will appear when it is reloaded or restructured, and the IBM native indexes for the restructured database.

For the Restructure function, all control blocks in the data set will be copied to the IMSACBA/IMSACBB DD statements in your IMS control region.

**NOTE**

The IMS/ESA release level of the ACB library must be the same as that of the RESLIB included in the STEPLIB DD statement concatenation.

**NEWPFXLB DD**

Optional for the Prepare, Restructure, and Restart functions. This DD statement identifies the staging library that contains the modified PFX indexes for the restructured database. During the Restructure function, Fast Path Online Restructure/EP copies the contents of the NEWPFXLB data set to the PFXLIB data set. The NEWPFXLB DD statement is required when the following conditions exist:

- Fast Path Indexer/EP is active in the IMS control region specified on the ACCESS keyword.
- The IMS control region STEPLIB DD statement contains the PFXLEVEL member.
- The IMS control region JCL includes the PFXLIB, PFXLIBA, or PFXLIBB DD statement.
NEWRESLB DD

Optional for the Prepare, Shadow Initialization, Restructure, and unload functions. The NEWRESLB DD statement identifies the library or libraries that contain randomizer and compression routines specified in the ACB contained within the NEWACB DD statement.

**NOTE**

If the NEWRESLB DD statement is specified in the JCL, then the randomizer and compression routines only are loaded directly from the NEWRESLB DD statement. If the NEWRESLB DD statement is not present in the JCL, then these routines are loaded from the STEPLIB DD statement or the IMSRESLB DD statement.

Using NEWRESLB enables you to make changes to the randomizer modules for use with the unload, while still retaining the old modules with the same name. The following conditions apply:

- The compression routine specified in NEWRESLB is not used to write the unload output file. However, if EXPAND=YES is specified, then the compression routine must be specified in NEWRESLB so that Fast Path/EP can validate it.
- The randomizer specified in NEWRESLB is used to write the unload output file.

OAREAxxx DD

Used for the unload and extract functions; not used for other functions.

OARxxxxx DD

These DD statements are used to identify the output area data sets for areas to be processed by the unload or extract command. Depending on the total number of areas in a DEDB, the respective DD statements can be used as follows:

- OAREAxxx – to identify an area number less than or equal to 999, where xxx is the area number preceded by leading zeroes (i.e., area number 4 would be IAREA004)
- OARxxxxx – to identify an area number between 1 to 2048, where xxxxx is the area number preceded by leading zeroes (i.e., area number 2048 would be IAR02048)

If either of these DD statements are omitted from the JCL, Fast Path/EP attempts to dynamically allocate the output area or areas using the OUTPUT_DSN_MASK keyword. If you specify the OUTPUT_DSN_MASK keyword to dynamically allocate the output unload data set or output extract file, do not include the OAREAxxx or OARxxxxx DD statement.
Figure 10 shows a sample DBD for a database containing 2048 areas.

![DBD for 2048-area database](Figure10)

The command set shown in Figure 11 uses OAREAxxx and OARxxxxx DD statements to identify areas for unload output. This example shows how OAREAxxx and OARxxxxx can be used interchangeably for area numbers less than 1000.

![Using OAREAxxx and OARxxxxx on UNLOAD command](Figure11)

OLCSTAT DD

Optional. Used to identify the active online IMS ACB data set (IMSACBA or IMSACBB) in an IMS global online change environment. Dynamic allocation can be used for this DD statement.
When this DD statement is present, the OLCSTAT data set is interrogated to
determine whether IMSACBA or IMSACBB is the active library. If the IMSACBA or
IMSACBB DD statement is not present in the JCL, the STEPLIB/LNKLIST are
searched for a DFSMDA member.

If both OLCSTAT and MODSTAT DD statements are present, then OLCSTAT will be
used and MODSTAT will be ignored.

**OLDACB DD**

Required if using the change function. No dynamic allocation is provided for this DD
statement. The OLDACB DD statement identifies the ACB library containing the
database definitions that describe the database before it is changed.

---

**NOTE**
The IMS/ESA release level of the ACB library must be the same as that of the RESLIB
included in the STEPLIB DD statement concatenation.

---

**OLDLIB DD**

Optional for index processing when using the change function. No dynamic
allocation is provided for this DD statement. The OLDLIB DD statement identifies the
PFX library containing the index registrations that describe the database before it is changed.

**OLDRESLB DD**

Optional for the change function. The OLDRESLB DD statement identifies the library
or libraries that contain randomizer and compression routines specified in the ACB
contained within the OLDACB DD statement. These routines are used by an area
before it is changed.

---

**NOTE**
If the OLDRESLB DD statement is specified in the JCL, then the randomizer and compression
routines only are loaded directly from the OLDRESLB DD statement. If the OLDRESLB DD
statement is not present in the JCL, then these routines are loaded from the STEPLIB DD
statement.

Using OLDRESLB lets you make changes to the randomizer and compression
routines for use with the change, while still retaining the old routines. The following
conditions apply:

- If COMPRESS=YES is specified, then the compression routine specified in
  OLDRESLB is used to expand segment data for the input area.
The randomizer that is specified in OLDRESLB is *not* used to write the changed output area. However, it must be specified in OLDRESLB so that Fast Path/EP can validate it for other processes which might be defined.

**PFPDFLTS DD**

Optional. The PFPDFLTS DD control statement contains members that are used to specify custom keyword parameters for the OPTIONS and GLOBAL commands.

To use the PFPDFLTS DD statement, you must complete the following tasks:

1. Create a partitioned data set (PDS) with DCB characteristics (RECFM=FB and LRECL=80).
2. Create a member named OPTIONS to specify the OPTIONS command and keywords.
3. Create a member named GLOBAL to specify the GLOBAL command and keywords.

**NOTE**

You may have an OPTIONS member, GLOBAL member, or both.

The following concatenation order applies to the OPTIONS command input:

- read and process the OPTIONS member from PFPDFLTS
- read and process the PFPOPTS DD statement

If both PFPDFLTS and PFPOPTS DD statements are specified in the JCL, then a keyword specified in PFPOPTS overrides the same keyword specified in the OPTIONS member.

The following concatenation order applies to the GLOBAL command input:

- read and process the GLOBAL member from PFPDFLTS
- read and process the GLOBAL command from the PFPSYSIN DD statement

If both PFPDFLTS and PFPSYSIN GLOBAL commands are specified in the JCL, then a keyword specified in the PFPSYSIN GLOBAL command overrides the same keyword specified in the GLOBAL member.

You can allocate the PFPDFLTS DD in the JCL or by using a dynamic allocation (DFSMDA) member named PFPDFLTS.
**PFPEPR DD**

Required to activate the Fast Path/EP repository feature and identify the repository catalog data set. If the PFPEPR DD statement is not specified in the JCL, and the REPOSITORY_DSNAME keyword is not specified on the OPTIONS command, Fast Path/EP dynamically allocates the repository data set by using the DFSMDA member. However, specifying the PFPEPR DD statement or the REPOSITORY_DSNAME overrides dynamic allocation of the DFSMDA member.

**PFPOPTS DD**

Optional. The PFPOPTS DD control statement contains the input control statements that set Fast Path/EP options. PFPOPTS uses the command statements that are in the Fast Path/EP command language. If no options are set with the PFPOPTS statement, internal defaults are used.

The PFPOPTS DD statement can refer to a standard input file, a sequential data set or PDS member that contains options settings.

**PFPPRINT DD**

Required *only* if you want to catalog your log output. The PFPPRINT DD statement identifies the processing log output data set. If the PFPPRINT DD statement is *not* specified in the JCL, Fast Path/EP dynamically allocates the data set using the following DCB characteristics:

- RECFM=VBA
- LRECL=137
- BLKSIZE=4096

The output data set must be a standard SYSOUT or sequential data set. The record format (RECFM) can specify fixed or variable length records, blocked or unblocked, and can include ANSI carriage control.

The logical record length (LRECL) can be any length. If an output record exceeds the LRECL, Fast Path/EP attempts to split the record into multiple lines along word boundaries. If an output record is shorter than the LRECL, the record is padded with trailing blanks.

The block size (BLKSIZE) can be any value appropriate for the LRECL and RECFM.

If you specify ANSI carriage control, a control character is generated for each logical record. If you do not specify ANSI carriage control, the Fast Path/EP Series product generates blank lines to simulate any carriage control function requested.
PFPRPTS DD

Required only if you want to catalog your reports. The PFPRPTS DD statement identifies the report output data set. If omitted from the JCL, Fast Path/EP dynamically allocates the data set using the following DCB characteristics:

- RECFM=VBA
- LRECL=137
- BLKSIZE=4096

The output data set must be a standard SYSOUT or sequential data set. The record format (RECFM) can specify fixed or variable length records, blocked or unblocked, and can include ANSI carriage control.

The logical record length (LRECL) can be any length. If an output record exceeds the LRECL, Fast Path/EP attempts to split the record into multiple lines along word boundaries. If an output record is shorter than the LRECL, the record is padded with trailing blanks.

The block size (BLKSIZE) can be any value appropriate for the LRECL and RECFM.

If you specify ANSI carriage control, a control character is generated for each logical record. If you do not specify ANSI carriage control, the Fast Path/EP Series product generates blank lines to simulate any carriage control function requested.

PFPSYSIN DD

Required. The PFPSYSIN DD statement identifies the input control statement data set that specifies the Fast Path/EP functions. The PFPSYSIN DD statement can be coded as a standard SYSIN file, a sequential data set, or PDS member.

PFPTOTAL DD

Optional for the unload, reload, or change function. This statement defines the area totals data set. When this DD is present, the unload, reload, or change process writes an output totals record, by area, for each segment type that is written. The record format is:

- Area Number: PL2
- Segment Name : CL8
- Total Occurrences: PL5
- Unload Date : PL4  (format 00YYDDD)
- Unload Time : XL4  (format HHMMSSHH)
- DBDname: CL8
- Areaname: CL8
- Reserved: XL21
The following DCB attributes of the file are set by the unload, reload, or change process, and do not need to be provided in the JCL:

- DSORG=PS
- LRECL=60
- BLKSIZE=6000

**PFXLIB DD**

Required for index processing. This DD statement identifies the PFX registration library. If not found in the JCL, the product searches for the MODSTAT DD statement or the OLCSTAT DD statement.

**PFXLIBA / PFXLIBB DD**

Optional, for online PFX registration data sets. The active PFX registration data set is allocated based on the active IMS ACB entry in the MODSTAT or OLCSTAT data set. Dynamic allocation can be used for these DD statements.

**RECONn DD**

Required when using the Data Base Recovery Control facility (DBRC) in offline mode; not used in online mode. If the RECONn DD statements are omitted from the JCL, Fast Path/EP attempts to dynamically allocate them using dynamic allocation members found in STEPLIB. The RECONn DD statements (RECON1, RECON2, and RECON3) define the data sets needed by DBRC.

**STEPLIB DD**

Required. The STEPLIB DD statement identifies the libraries containing the Fast Path/EP load modules (the Fast Path/EP library that was allocated and unloaded during installation) and the IMS RESLIB. It also identifies any libraries used for dynamic allocation.

The following requirements apply to the STEPLIB DD statement for certain Fast Path/EP products and functions:

- The Fast Path/EP online products require that the installation library containing the BMC Software-supplied region controller module, BMCRRC00 (alias DFSRRC00), be concatenated preceding the IMS RESLIB data set in the STEPLIB DD statement. The BMC Software DFSRRC00 module is a replacement of the IMS region controller module.
For the following Fast Path/EP functions that provide a sort option, Fast Path/EP can optionally use the IMS zIIP Sort package, LGBSORT:

- BUILD
- EXTRACT
- PFPSORT
- RELOAD
- RESYNC
- VERIFY
- XSCAN

LGBSORT is located in the hlq.XXLIB (merged installation) or hlq.LGBLIB (non-merged installation) data set. To use LGBSORT, you must add the data set to the STEPLIB DD concatenation; otherwise, Fast Path/EP uses a standard sort, such as DFSORT or SYNSORT.

**NOTE**
The variable hlq is a high-level qualifier that you provide.

The Fast Path Online Restructure/EP product requires the installation library containing IMS Database Utilities (DBU) load modules be included in the STEPLIB DD statement. BMC recommends that you also include the installation library containing the optional DELTA IMS (DLA) load modules in the STEPLIB DD statement.

**NOTE**
All data sets in this concatenation must be APF authorized.
Command language

The Fast Path/EP Series command language consists of commands, subcommands, and keywords which can be used in various combinations to control desired product functionality. This chapter provides descriptions and usage conventions for each command, subcommand and keyword.

Command language conventions ........................................... 54
Input control statements ...................................................... 56
OPTIONS command keywords and subcommands ...................... 57
GLOBAL command keywords and subcommands ....................... 58
PFPSYSIN DD statement ...................................................... 59
   PFPSYSIN commands and subcommands for PFPMAIN ................ 60
   PFPSYSIN keywords for PFPMAIN ..................................... 61
   PFPSYSIN commands and subcommands for PFCMAIN ................. 73
   PFPSYSIN keywords for PFCMAIN ..................................... 74
   PFPSYSIN commands and subcommands for PFPEPR00 .................. 80
Fast Path/EP operator interface commands .............................. 83
Command language description format ................................. 84
ACCESS ................................................................. 85
ACTIVITY_FILECTL .................................................... 87
ACTUATE .............................................................. 89
ADD ................................................................. 90
ADDN ............................................................... 91
ADSN ............................................................... 92
ALLOCATE .......................................................... 94
ANALYZE .......................................................... 95
AREA_KEY .......................................................... 96
AVGREC ........................................................... 97
BACKOUT .......................................................... 99
BREAK ............................................................ 99
BUILD ............................................................. 100
BYPASS_RECORD .................................................... 102
CASE .............................................................. 103
CHANGE .......................................................... 103
CHECKPOINT ....................................................... 105
COMPRESS ........................................................ 106
COMPRESSION .................................................... 107
Chapter 2  Command language  53

REQUIRE_AREA ......................................................... 271
RESET ................................................................. 273
RESTART ............................................................... 274
RESTRUCTURE ......................................................... 275
RESYNC ................................................................. 276
RETAINED_SUFFIX .................................................. 277
RETPD ................................................................. 278
RETRIEVE .............................................................. 279
REUSE ................................................................. 280
ROOT_IO_AVERAGE ................................................... 281
ROOT_IO_MAXIMUM ................................................... 282
ROUTCDE .............................................................. 283
SAMPLE_INTERVAL .................................................... 284
SAMPLE_LIMIT ......................................................... 285
SCAN ................................................................. 286
SCRIPT ................................................................. 287
SDEP_PROCESS ........................................................ 287
SDEP_VALIDATION ...................................................... 290
SEGMENT ............................................................... 291
SEGMENT_IO_ANALYSIS ............................................. 292
SEGMENT_LENGTH_ANALYSIS ....................................... 293
SEGMENT_PLACEMENT_ANALYSIS ................................... 294
SEGMENT_RECORD_PREFIX .......................................... 295
SEGMENT_RECORD_SUFFIX ........................................... 296
SELECT_AREA ........................................................ 297
SELECT_DATE ........................................................ 299
SELECT_DBD ........................................................... 300
SELECT_GROUP ........................................................ 301
SELECT_LIMIT ........................................................ 302
SELECT_UOW ........................................................ 302
SET ................................................................. 304
SHADOW_INIT ........................................................ 305
SHADOW_SUFFIX ....................................................... 306
SHADOW2_DSNAME ................................................... 307
SHADOW2_SUFFIX ...................................................... 309
SHUTDOWN ............................................................ 310
SORT ................................................................. 311
SORT_NAME ............................................................ 312
SORT_OPTION .......................................................... 313
SORT_SEQUENCE ..................................................... 315
SPACE ............................................................... 316
STACK_NAME ............................................................ 318
STARTUOW ............................................................. 319
STATUS ................................................................. 320
STOPUOW ............................................................... 321
STORCLAS ............................................................... 322
STORCLAS2 ............................................................. 323
SUBSET_POINTERS .................................................... 324
SYNONYM_CHAIN_ANALYSIS ....................................... 325
The Fast Path/EP Series language consists of commands, subcommands, and keywords. Keywords are used instead of parameters because parameters in these products are all keyword=parameter constructs; positional parameters are not part of the command language definition.

**Commands**

The PFPSYSIN or PFPOPTS DD statement must be followed by a valid Fast Path/EP command. A command set is defined as a command, the command keywords and parameters, any subcommands, and the subcommand keywords and parameters.

You can abbreviate command names within certain guidelines. You can truncate the command name to as few as three characters as long as the truncated string uniquely identifies the command. For example, the ANALYZE command can be coded as ANA since these leading characters are unique to the ANALYZE command.
Subcommands

Subcommands can be used with certain commands to modify or enhance the primary function. A subcommand set is defined as a subcommand, its keywords, and keyword parameters. Valid commands and subcommands are shown beginning with Table 6.

You can abbreviate subcommand names within certain guidelines. You can truncate the subcommand name to as few as three characters as long as the truncated string uniquely identifies the subcommand. For example, the REPORT subcommand can be coded as REP since these leading characters are unique to the REPORT subcommand.

Keywords

Keywords follow a command or subcommand and invoke specified or default parameters. All keywords are nonpositional and can be specified in any order. Separate the keyword and its parameter with an equal sign, e.g., `keyword=parameter`, or by using parentheses, e.g., `keyword(parameter)`. If multiple parameters are specified, the list of parameters must be enclosed within parentheses, e.g., `keyword=(parameter, parameter, parameter)`; the equals sign (=) is optional. Keywords must be separated by commas.

You can abbreviate keyword names within certain guidelines. You can truncate the keyword to as few as three characters as long as the truncated string uniquely identifies the keyword. For example, the POINTER_VALIDATION keyword can be coded as POI since these leading characters are unique to the POINTER_VALIDATION keyword.

A truncated keyword name must be unique only among valid keywords for the command or subcommand on which it is coded. The truncated name does not have to be unique across with all Fast Path/EP Series command language keywords. For example, on the ANALYZE command, the INPUT_DSN_MASK keyword can be truncated to INP, because the truncated keyword is distinguishable from all other valid ANALYZE command keywords. On the UNLOAD command, however INP would not be sufficient to distinguish it from the INPUT_THREADS keyword.

**NOTE**

If new keywords are added to the various commands and subcommands in future maintenance levels, the minimum lengths for the various truncated keywords might change.

Parameters

Some keywords require only one parameter and some require more than one. Some keywords accept a list of parameters. Keywords require either numeric, character, or character string values. Lists of parameters must be contained within parentheses and separated by commas.
Some parameters are specified as predefined names. For example, the valid values for the POINTER_VALIDATION keyword are FULL, QUICK, OFF and NONE. The value names can be truncated to as few as three characters as long as the truncated string uniquely identifies a valid value for the keyword.

Comments

Comments in the PFPSYSIN or PFPOPTS command set are designated by an asterisk (*) character in column 1. If two consecutive slash characters (//) appear anywhere in a command line, the remainder of that command line is treated as a comment.

Separator characters

One or more commas can appear between keywords. Commas are used to separate parameters.

Parentheses are used to group listed parameters following a keyword. Commas are used to separate listed parameter entries inside parentheses, following the keyword construct. For example, IAREA=(a,b,c,d,...).

Continuation characters

There are no continuation characters. All commands, subcommands keywords are free in form, and can be continued from one line to the next.

Input control statements

Fast Path/EP Series products use two DD statements (PFPOPTS and PFPSYSIN) for input control statements. In offline mode, an operator interface is available for diagnostic command input.

Figure 12  Sample Fast Path/EP general JCL

```plaintext
//PFP      EXEC PGM=PFPMAIN,REGION=0M
//STEPLIB  DD DSN=BMC.PFP.LOAD,DISP=SHR
//         DD DSN=IMSVS.RESLIB,DISP=SHR
//PFPOPTS  DD *
OPTIONS
  REPOSITORY_DSN='repository.dsname',
  LINE_COUNT=110
/*
//PFPSYSIN  DD *
GLOBAL DBRC=YES END
REORGANIZE DBD=dbdnam1,IAREA=areaname
LOADCTL
  SEGMENT=segname,
  INSERT_LIMIT_COUNT=1000,
```
The standard set of input control statements have the following requirements:

- You can enter commands anywhere in positions 1 through 72 of the input statement (positions 73 through 80 are ignored).
- There are no continuation characters.
- They must contain 80-character fixed length records.
- A command can have keywords and subcommands, separated where necessary with separator characters.
- An asterisk (*) in column 1 indicates a comment.

**OPTIONS command keywords and subcommands**

Fast Path/EP accepts the OPTIONS command from one of the following items:

- PFPOPTS DD statement
- OPTIONS module from the PFPDFLTS DD

The OPTIONS command tells Fast Path/EP which options to use.

Table 2 shows the OPTIONS command and its keywords, including a brief description of use.

**Table 2  OPTIONS command keywords (part 1 of 2)**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASE</td>
<td>specify character display</td>
</tr>
<tr>
<td>DATE_TIME_FORMAT</td>
<td>specify default format for date and time data</td>
</tr>
<tr>
<td>DESC</td>
<td>set WTO message descriptor code</td>
</tr>
<tr>
<td>IFP_ACCOUNT</td>
<td>specify accounting information for IFP region</td>
</tr>
<tr>
<td>IFP_JOBNAME</td>
<td>specify job name for IFP region</td>
</tr>
<tr>
<td>IFP_LIMIT</td>
<td>specify maximum number of concurrent IFP regions</td>
</tr>
</tbody>
</table>
GLOBAL command keywords and subcommands

Table 2  OPTIONS command keywords (part 2 of 2)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFP_PROCNAME</td>
<td>specify cataloged procedure name for IFP region</td>
</tr>
<tr>
<td>LANGUAGE</td>
<td>specify language code</td>
</tr>
<tr>
<td>LINE_COUNT</td>
<td>specify report lines per page</td>
</tr>
<tr>
<td>LIST_OPTIONS</td>
<td>list option settings</td>
</tr>
<tr>
<td>MONITOR</td>
<td>periodically initiates operator DISPLAY</td>
</tr>
<tr>
<td>PRODUCT_LIMIT</td>
<td>limit concurrent tasks</td>
</tr>
<tr>
<td>REPOSITORY_DSNAME</td>
<td>specify data set name for repository</td>
</tr>
<tr>
<td>REPOSITORY_GROUP</td>
<td>specify group parameter for repository</td>
</tr>
<tr>
<td>REPOSITORY_OVERWRITE</td>
<td>control repository allocation</td>
</tr>
<tr>
<td>REPOSITORY_RETENTION_COUNT</td>
<td>control repository retention (by number of entries)</td>
</tr>
<tr>
<td>REPOSITORY_RETENTION_PERIOD</td>
<td>control repository retention (by time)</td>
</tr>
<tr>
<td>ROUTCDE</td>
<td>set WTO message routing codes</td>
</tr>
<tr>
<td>SORT_NAME</td>
<td>specify name of the sort package</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>generate time stamp for messages</td>
</tr>
<tr>
<td>WARNING</td>
<td>specify warning effects</td>
</tr>
<tr>
<td>WORK_DATASET</td>
<td>specify dynamic allocation options</td>
</tr>
</tbody>
</table>

Table 3 shows the OVERRIDE and RESET subcommands, which can be specified under the OPTIONS command to customize eligible product messages.

Table 3  OVERRIDE and RESET subcommands and keywords

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Keyword</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVERRIDE</td>
<td>MESSAGE_LEVEL</td>
<td>change default severity level for message</td>
</tr>
<tr>
<td></td>
<td>MESSAGE_LIMIT</td>
<td>specify threshold limit (number) for suppression of message</td>
</tr>
<tr>
<td></td>
<td>MESSAGE_NUMBER</td>
<td>specify ID number of message to be customized</td>
</tr>
<tr>
<td>RESET</td>
<td>MESSAGE_NUMBER</td>
<td>specify ID number of message to be restored to default</td>
</tr>
</tbody>
</table>

GLOBAL command keywords and subcommands

Use the GLOBAL command to establish implicit job step keyword values.

Fast Path/EP accepts the GLOBAL command from one of the following items:

- PFPSYSIN DD statement input control card statement
- GLOBAL module from the PFPDFLTS DD
Table 4 shows the GLOBAL command and its keywords, including a brief description of use.

### Table 4  GLOBAL command keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCESS</td>
<td>specify area access mode</td>
</tr>
<tr>
<td>DBRC</td>
<td>request DBRC for PFP</td>
</tr>
<tr>
<td>EARLY_TERMINATION</td>
<td>specify when to terminate post-processing</td>
</tr>
<tr>
<td>FLOWER_BOX</td>
<td>specify flowerbox border</td>
</tr>
<tr>
<td>HISTORY_DDNAME</td>
<td>specify statistics extract</td>
</tr>
<tr>
<td>LARGEST_DATABASE_RECORDS</td>
<td>specify number of largest database records</td>
</tr>
<tr>
<td>MESSAGE_SUPPRESSION</td>
<td>suppress repetitious messages</td>
</tr>
<tr>
<td>ORPHANED_SDEP_MSG</td>
<td>control SDEP error messages</td>
</tr>
<tr>
<td>OUTAGE_WINDOW</td>
<td>specify outage time for database</td>
</tr>
<tr>
<td>POINTER_VALIDATION</td>
<td>request pointer validation</td>
</tr>
<tr>
<td>RAP_VALIDATION</td>
<td>request rap validation</td>
</tr>
<tr>
<td>SCAN</td>
<td>request syntax scan</td>
</tr>
<tr>
<td>SDEP_VALIDATION</td>
<td>request SDEP pointer validation</td>
</tr>
<tr>
<td>TYPE_RUN</td>
<td>specify execution mode</td>
</tr>
</tbody>
</table>

Table 5 shows the IC, REPORT, and THRESHOLD subcommands that can be specified under the GLOBAL command.

### Table 5  IC, REPORT, and THRESHOLD subcommands

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Mode Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC</td>
<td>offline and online</td>
</tr>
<tr>
<td>REPORT</td>
<td>offline and online</td>
</tr>
<tr>
<td>THRESHOLD</td>
<td>offline and online</td>
</tr>
</tbody>
</table>

**PFPSYSIN DD statement**

Fast Path/EP Series products accept commands, subcommands, and keywords from the PFPSYSIN DD statement. These commands control which functions are performed by either the PFPMAIN program (primary maintenance and analysis functions), the PFCMAIN program (Fast Path Online Restructure/EP maintenance and restructure functions), or the PFPEPR00 utility program (Fast Path Analyzer/EP repository maintenance and retrieval functions).
PFPSYSIN commands and subcommands for PFPMAIN

Table 6 shows PFPSYSIN commands available for execution by PFPMAIN, the subcommands available for each command, and the processing modes in which they are available.

Table 6  PFPSYSIN commands and subcommands for PFPMAIN (part 1 of 2)

<table>
<thead>
<tr>
<th>Command</th>
<th>Subcommand</th>
<th>Mode availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANALYZE</td>
<td>IC</td>
<td>offline and online</td>
</tr>
<tr>
<td></td>
<td>CORRECTIONS_FILECTL</td>
<td>offline and online</td>
</tr>
<tr>
<td></td>
<td>REPORT</td>
<td>offline and online</td>
</tr>
<tr>
<td></td>
<td>THRESHOLD</td>
<td>offline and online</td>
</tr>
<tr>
<td>BUILD</td>
<td>IX</td>
<td>offline only</td>
</tr>
<tr>
<td>CHANGE</td>
<td>ALLOCATE</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>EXCLUDE</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>IC</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>INCLUDE</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>IX</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>LOADCTL</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>OUTPUT</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>REPORT</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>THRESHOLD</td>
<td>offline only</td>
</tr>
<tr>
<td>DMAC_CLEANUP</td>
<td>None</td>
<td>offline and online</td>
</tr>
<tr>
<td>DMAC_PRINT</td>
<td>None</td>
<td>offline and online</td>
</tr>
<tr>
<td>END</td>
<td>None</td>
<td>offline and online</td>
</tr>
<tr>
<td>EXTEND</td>
<td>IC</td>
<td>offline and online</td>
</tr>
<tr>
<td></td>
<td>REPORT</td>
<td>offline and online</td>
</tr>
<tr>
<td></td>
<td>THRESHOLD</td>
<td>offline and online</td>
</tr>
<tr>
<td>EXTRACT</td>
<td>EXCLUDE</td>
<td>offline and online</td>
</tr>
<tr>
<td></td>
<td>INCLUDE</td>
<td>offline and online</td>
</tr>
<tr>
<td></td>
<td>OFILECTL</td>
<td>offline and online</td>
</tr>
<tr>
<td></td>
<td>OUTPUT</td>
<td>offline and online</td>
</tr>
<tr>
<td></td>
<td>USER_RECORD</td>
<td>offline and online</td>
</tr>
<tr>
<td>IMAGECOPY</td>
<td>IC</td>
<td>online only</td>
</tr>
<tr>
<td></td>
<td>REPORT</td>
<td>online only</td>
</tr>
<tr>
<td></td>
<td>THRESHOLD</td>
<td>online only</td>
</tr>
<tr>
<td>INITIALIZE</td>
<td>ALLOCATE</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>IC</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>IX</td>
<td>offline only</td>
</tr>
<tr>
<td>PFPSORT</td>
<td>None</td>
<td>offline only</td>
</tr>
</tbody>
</table>
Table 6  PFPSYSIN commands and subcommands for PFPMAIN (part 2 of 2)

<table>
<thead>
<tr>
<th>Command</th>
<th>Subcommand</th>
<th>Mode availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESS_AREA</td>
<td>PERFORM</td>
<td>online and offline</td>
</tr>
<tr>
<td>RELOAD</td>
<td>ALLOCATE</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>DISCARD_FILECTL</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>EXCLUDE</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>IC</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>INCLUDE</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>IX</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>LOADCTL</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>OUTPUT</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>REPORT</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>THRESHOLD</td>
<td>offline only</td>
</tr>
<tr>
<td>REORGANIZE</td>
<td>IC</td>
<td>offline and online</td>
</tr>
<tr>
<td></td>
<td>LOADCTL</td>
<td>offline and online</td>
</tr>
<tr>
<td></td>
<td>REPORT</td>
<td>offline and online</td>
</tr>
<tr>
<td></td>
<td>THRESHOLD</td>
<td>offline and online</td>
</tr>
<tr>
<td>RESYNC</td>
<td>IX</td>
<td>BMP only</td>
</tr>
<tr>
<td>RETRIEVE</td>
<td>REPORT</td>
<td>offline only</td>
</tr>
<tr>
<td>UNLOAD</td>
<td>EXCLUDE</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>IC</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>INCLUDE</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>OFILECTL</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>OUTPUT</td>
<td>offline only</td>
</tr>
<tr>
<td>VERIFY</td>
<td>IX</td>
<td>offline and BMP</td>
</tr>
<tr>
<td>XSCAN</td>
<td>IX</td>
<td>offline and IFP</td>
</tr>
</tbody>
</table>

PFPSYSIN keywords for PFPMAIN

Table 7 shows the keywords available for PFPMAIN commands and subcommands, including a brief description of use and keyword aliases. Refer to Table 6 for a list of subcommands available for each command.
### Table 7  PFPSYSIN commands and keywords for PFPMAIN (part 1 of 11)

<table>
<thead>
<tr>
<th>Command or Subcommand</th>
<th>Keyword</th>
<th>Function</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ALLOCATE (subcommand)</strong></td>
<td>ACTUATE</td>
<td>request optional VSAM cluster processing</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>AVGREC</td>
<td>identify space requirements</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>CONFIGURE_AREA</td>
<td>specify area data set allocation characteristics</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>CONFIGURE_IOVF</td>
<td>specify IOVF allocation characteristics</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>CONFIGURE_RAA</td>
<td>specify RAA allocation characteristics</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>CONFIGURE_SDEP</td>
<td>specify SDEP allocation characteristics</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DATACLAS</td>
<td>specify SMS data class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>IDCAMS_OPTION</td>
<td>specify custom Access Method Services options</td>
<td>AMSOPT</td>
</tr>
<tr>
<td></td>
<td>MGMTCLAS</td>
<td>specify SMS management class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>OAREA</td>
<td>specify output area(s)</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>REPORT_DDNAME</td>
<td>specify report destination</td>
<td>RPTDD</td>
</tr>
<tr>
<td></td>
<td>SPACE</td>
<td>specify space requirements</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>STORCLAS</td>
<td>specify SMS storage class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLSER</td>
<td>specify volume serial number</td>
<td>None</td>
</tr>
<tr>
<td><strong>ANALYZE</strong></td>
<td>ACCESS</td>
<td>specify area access mode</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td></td>
<td>HISTORY_DDNAME</td>
<td>specify statistics extract</td>
<td>HISDD</td>
</tr>
<tr>
<td></td>
<td>IAREA</td>
<td>specify input area(s)</td>
<td>AREA</td>
</tr>
<tr>
<td></td>
<td>ICACHE</td>
<td>specify input cache</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>INPUT_DSN_MASK</td>
<td>create dynamic allocation mask</td>
<td>IDM</td>
</tr>
<tr>
<td></td>
<td>LARGEST_DATABASE_RECORDS</td>
<td>specify number of largest database records</td>
<td>LDR KEYS</td>
</tr>
<tr>
<td></td>
<td>MESSAGE_SUPPRESSION</td>
<td>suppress repeated messages</td>
<td>MSGSUP</td>
</tr>
<tr>
<td></td>
<td>MODEL_DDNAME</td>
<td>specify modeler output</td>
<td>MDLDD</td>
</tr>
<tr>
<td></td>
<td>ORPHANED_SDEP_MSG</td>
<td>control SDEP error messages</td>
<td>OSM</td>
</tr>
<tr>
<td></td>
<td>POINTER_VALIDATION</td>
<td>request pointer validation</td>
<td>PTR or FPA</td>
</tr>
<tr>
<td></td>
<td>RAP_VALIDATION</td>
<td>request RAP validation</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SDEP_VALIDATION</td>
<td>request SDEP pointer validation</td>
<td>None</td>
</tr>
</tbody>
</table>
Table 7  PFPSYSIN commands and keywords for PFPMAIN (part 2 of 11)

<table>
<thead>
<tr>
<th>Command or Subcommand</th>
<th>Keyword</th>
<th>Function</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUILD</td>
<td>DBD</td>
<td>specify database</td>
<td>DBD</td>
</tr>
<tr>
<td></td>
<td>IAREA</td>
<td>specify input area(s)</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>ICACHE</td>
<td>specify input cache</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>INDEX_THREADS</td>
<td>limit index threads</td>
<td>XTHREADS</td>
</tr>
<tr>
<td></td>
<td>INPUT_DSN_MASK</td>
<td>input dynamic allocation mask</td>
<td>IDM</td>
</tr>
<tr>
<td></td>
<td>INPUT_THREADS</td>
<td>limit input threads</td>
<td>ITHREADS</td>
</tr>
<tr>
<td></td>
<td>MESSAGE_SUPPRESSION</td>
<td>suppress repetitious messages</td>
<td>MSGSUP</td>
</tr>
<tr>
<td></td>
<td>SORT</td>
<td>sort index database</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SORT_OPTION</td>
<td>specify custom sort options</td>
<td>None</td>
</tr>
<tr>
<td>CHANGE</td>
<td>BYPASS_RECORD</td>
<td>control processing of roots randomized to non-processed areas</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td></td>
<td>ERROR_THRESHOLD</td>
<td>control pointer error tolerance</td>
<td>ERT</td>
</tr>
<tr>
<td></td>
<td>EXPAND</td>
<td>expand compressed data</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>IAREA</td>
<td>specify input area(s)</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>ICACHE</td>
<td>specify input cache</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>INDEX_THREADS</td>
<td>limit index threads</td>
<td>XTHREADS</td>
</tr>
<tr>
<td></td>
<td>INPUT_DSN_MASK</td>
<td>input dynamic allocation mask</td>
<td>IDM</td>
</tr>
<tr>
<td></td>
<td>INPUT_THREADS</td>
<td>limit input threads</td>
<td>ITHREADS</td>
</tr>
<tr>
<td></td>
<td>IOVF_LOAD_HWM</td>
<td>limit space usage in IOVF blocks</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>LARGEST_DATABASE_RECORDS</td>
<td>specify number of largest database records</td>
<td>LDR, KEYS</td>
</tr>
<tr>
<td></td>
<td>MESSAGE_SUPPRESSION</td>
<td>suppress repetitious messages</td>
<td>MSGSUP</td>
</tr>
<tr>
<td></td>
<td>OAREA</td>
<td>specify output area(s)</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>OCACHE</td>
<td>specify output cache</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>ORPHANED_SDEP_MSG</td>
<td>control SDEP error messages</td>
<td>OSM</td>
</tr>
<tr>
<td></td>
<td>OUTPUT_DSN_MASK</td>
<td>output dynamic allocation mask</td>
<td>ODM</td>
</tr>
<tr>
<td></td>
<td>OUTPUT_THREADS</td>
<td>limit output threads</td>
<td>OTHREADS</td>
</tr>
<tr>
<td></td>
<td>POINTER_VALIDATION</td>
<td>request pointer validation</td>
<td>PTR or FPA</td>
</tr>
<tr>
<td></td>
<td>RAP_VALIDATION</td>
<td>request RAP validation</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SDEP_PROCESS</td>
<td>process SDEP segments</td>
<td>SDEP</td>
</tr>
<tr>
<td></td>
<td>SDEP_VALIDATION</td>
<td>request SDEP pointer validation</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SUBSET_POINTERS</td>
<td>set subset pointer processing</td>
<td>SSPTR</td>
</tr>
<tr>
<td>Command or Subcommand</td>
<td>Keyword</td>
<td>Function</td>
<td>Alias</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------</td>
<td>-----------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>CORRECTIONS_FILECTL (subcommand)</td>
<td>AVGREC</td>
<td>identify space requirements</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DDNAME</td>
<td>specify ddname for pointer error corrections file</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DATACLAS</td>
<td>specify SMS data class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DISP</td>
<td>specify disposition of pointer error corrections file</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DSNAME</td>
<td>specify data set mask</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>EXPT</td>
<td>specify expiration date</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>LIKE</td>
<td>specify model data set</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>MGMTCLAS</td>
<td>specify SMS management class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>RETPD</td>
<td>specify retention period</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SPACE</td>
<td>specify space requirements</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>STORCLAS</td>
<td>specify SMS storage class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>UNIT</td>
<td>specify physical device</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLCNT</td>
<td>specify number of volumes</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLSER</td>
<td>specify volume serial numbers</td>
<td>None</td>
</tr>
<tr>
<td>DMAC_CLEANUP</td>
<td>DBD</td>
<td>update DMAC for a database</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>IAREA</td>
<td>specify input area(s)</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>MESSAGE SUPPRESSION</td>
<td>suppress repeated messages</td>
<td>MSGSUP</td>
</tr>
<tr>
<td>DMAC_PRINT</td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td></td>
<td>IAREA</td>
<td>specify input area(s)</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>INPUT_DSN_MASK</td>
<td>input dynamic allocation mask</td>
<td>IDM</td>
</tr>
<tr>
<td></td>
<td>MESSAGE SUPPRESSION</td>
<td>suppress repeated messages</td>
<td>MSGSUP</td>
</tr>
<tr>
<td>DISCARD_FILECTL (subcommand)</td>
<td>AVGREC</td>
<td>identify space requirements</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DATACLAS</td>
<td>specify SMS data class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DDNAME</td>
<td>specify ddname for discard file</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DISP</td>
<td>specify disposition of discard file data set</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DSNAME</td>
<td>specify data set mask</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>EXPT</td>
<td>specify expiration date</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>LIKE</td>
<td>specify model data set</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>MGMTCLAS</td>
<td>specify SMS management class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>RETPD</td>
<td>specify retention period</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SPACE</td>
<td>specify space requirements</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>STORCLAS</td>
<td>specify SMS storage class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>UNIT</td>
<td>specify physical device</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLCNT</td>
<td>specify number of volumes</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLSER</td>
<td>specify volume serial numbers</td>
<td>None</td>
</tr>
</tbody>
</table>
### Table 7  PFPSYSIN commands and keywords for PFPMAIN (part 4 of 11)

<table>
<thead>
<tr>
<th>Command or Subcommand</th>
<th>Keyword</th>
<th>Function</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXCLUDE</td>
<td>SAMPLE_INTERVAL</td>
<td>specify sampling interval</td>
<td>INTERVAL</td>
</tr>
<tr>
<td></td>
<td>SAMPLE_LIMIT</td>
<td>specify sample limit</td>
<td>LIMIT</td>
</tr>
<tr>
<td></td>
<td>SEGMENT</td>
<td>specify segment name</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>WHERE</td>
<td>specify conditional selection</td>
<td>None</td>
</tr>
<tr>
<td>EXTEND</td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td></td>
<td>EXTEND_IOVF</td>
<td>increase IOVF storage size</td>
<td>IOVF</td>
</tr>
<tr>
<td></td>
<td>EXTEND_SDEP</td>
<td>increase SDEP storage size</td>
<td>SDEP</td>
</tr>
<tr>
<td></td>
<td>IAREA</td>
<td>specify input area(s)</td>
<td>AREA</td>
</tr>
<tr>
<td></td>
<td>INPUT_DSN_MASK</td>
<td>input dynamic allocation mask</td>
<td>IDM</td>
</tr>
<tr>
<td></td>
<td>LARGEST_DATABASE_RECORDS</td>
<td>specify number of largest database</td>
<td>LDR KEYS</td>
</tr>
<tr>
<td></td>
<td>MESSAGE_SUPPRESSION</td>
<td>suppress repetitious messages</td>
<td>MSGSUP</td>
</tr>
<tr>
<td></td>
<td>ORPHANED_SDEP_MSG</td>
<td>control SDEP error messages</td>
<td>OSM</td>
</tr>
<tr>
<td></td>
<td>POINTER_VALIDATION</td>
<td>request pointer validation</td>
<td>PTR or FPA</td>
</tr>
<tr>
<td></td>
<td>RAP_VALIDATION</td>
<td>request RAP validation</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SDEP_VALIDATION</td>
<td>request SDEP pointer validation</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>TYPE_RUN</td>
<td>specify execution mode</td>
<td>None</td>
</tr>
<tr>
<td>EXTRACT</td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td></td>
<td>EXPAND</td>
<td>expand compressed data</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>EXTRACT_FORMAT</td>
<td>specify extract format</td>
<td>FORMAT</td>
</tr>
<tr>
<td></td>
<td>IAREA</td>
<td>specify input area(s)</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>ICACHE</td>
<td>specify input cache</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>INPUT_DSN_MASK</td>
<td>input dynamic allocation mask</td>
<td>IDM</td>
</tr>
<tr>
<td></td>
<td>OUTPUT_DSN_MASK</td>
<td>output dynamic allocation mask</td>
<td>ODM</td>
</tr>
<tr>
<td></td>
<td>SEGMENT_RECORD_PREFIX</td>
<td>specify prefix contents</td>
<td>PREFIX</td>
</tr>
<tr>
<td></td>
<td>SEGMENT_RECORD_SUFFIX</td>
<td>specify suffix contents</td>
<td>SUFFIX</td>
</tr>
<tr>
<td></td>
<td>SORT</td>
<td>sort extract file</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SORT_OPTION</td>
<td>specify custom sort options</td>
<td>None</td>
</tr>
</tbody>
</table>
Table 7  PFPSYSIN commands and keywords for PFPMAIN (part 5 of 11)

<table>
<thead>
<tr>
<th>Command or Subcommand</th>
<th>Keyword</th>
<th>Function</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC (subcommand)</td>
<td>AVGREC</td>
<td>identify space requirements</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>COMPRESSION</td>
<td>specify data compression</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DATAACLAS</td>
<td>specify SMS data class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DDNAME</td>
<td>specify ddname for image copy</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DISP</td>
<td>specify disposition of image</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td>copy data set</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DSNAME</td>
<td>specify data set mask</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>EXPDT</td>
<td>specify expiration date</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>LIKE</td>
<td>specify model data set</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>MGMTCLAS</td>
<td>specify SMS management class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>NOTIFY</td>
<td>set DBRC notification</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>RETPD</td>
<td>specify retention period</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SPACE</td>
<td>specify space requirements</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>STACK_NAME</td>
<td>specify name of group of stacked</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td>output data sets</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STORCLAS</td>
<td>specify SMS storage class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>UNIT</td>
<td>specify physical device</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLCNT</td>
<td>specify number of volumes</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLSER</td>
<td>specify volume serial numbers</td>
<td>None</td>
</tr>
<tr>
<td>IMAGECOPY</td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td></td>
<td>IAREA</td>
<td>specify input area</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>LARGEST_DATABASE_RECORDS</td>
<td>specify number of largest database records</td>
<td>LDR KEYS</td>
</tr>
<tr>
<td></td>
<td>MESSAGE_SUPPRESSION</td>
<td>suppress repetitious messages</td>
<td>MSGSUP</td>
</tr>
<tr>
<td></td>
<td>ORPHANED_SDEP_MSG</td>
<td>control SDEP error messages</td>
<td>OSM</td>
</tr>
<tr>
<td></td>
<td>POINTER_VALIDATION</td>
<td>request pointer validation</td>
<td>PTR or FPA</td>
</tr>
<tr>
<td></td>
<td>RAP_VALIDATION</td>
<td>request RAP validation</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SDEP_VALIDATION</td>
<td>request SDEP pointer validation</td>
<td>None</td>
</tr>
<tr>
<td>INCLUDE (subcommand)</td>
<td>SAMPLE_INTERVAL</td>
<td>specify sampling interval</td>
<td>INTERVAL</td>
</tr>
<tr>
<td></td>
<td>SAMPLE_LIMIT</td>
<td>specify sample limit</td>
<td>LIMIT</td>
</tr>
<tr>
<td></td>
<td>SEGMENT</td>
<td>specify segment name</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>WHERE</td>
<td>specify conditional selection</td>
<td>None</td>
</tr>
<tr>
<td>INITIALIZE</td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td></td>
<td>IAREA</td>
<td>specify output area(s)</td>
<td>AREA</td>
</tr>
<tr>
<td></td>
<td>INPUT_DSN_MASK</td>
<td>create dynamic allocation mask</td>
<td>IDM</td>
</tr>
<tr>
<td></td>
<td>MESSAGE_SUPPRESSION</td>
<td>suppress repetitious messages</td>
<td>MSGSUP</td>
</tr>
</tbody>
</table>
### Table 7  PFPSYSIN commands and keywords for PFPMAIN (part 6 of 11)

<table>
<thead>
<tr>
<th>Command or Subcommand</th>
<th>Keyword</th>
<th>Function</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>IX (subcommand)</td>
<td>AVGREC</td>
<td>identify space requirements</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DATACLAS</td>
<td>specify SMS data class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DDNAME</td>
<td>specify ddname for file</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DISP</td>
<td>specify disposition of index</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DSNAME</td>
<td>specify data set name mask</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>EXPDT</td>
<td>specify expiration date</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>INDEX</td>
<td>specify index database</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>INPUT_DSN_MASK</td>
<td>create dynamic allocation mask</td>
<td>IDM</td>
</tr>
<tr>
<td></td>
<td>LIKE</td>
<td>specify model data set</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>MGMTCLAS</td>
<td>specify SMS management</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>OUTPUT_DSN_MASK</td>
<td>output dynamic allocation mask</td>
<td>ODM</td>
</tr>
<tr>
<td></td>
<td>RETPD</td>
<td>specify retention period</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SORT</td>
<td>sort index file</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SORT_OPTION</td>
<td>specify sort options for index</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SPACE</td>
<td>specify space requirements</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>STORCLAS</td>
<td>specify SMS storage class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>UNIT</td>
<td>specify physical device</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLCNT</td>
<td>specify number of volumes</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLSER</td>
<td>specify volume serial numbers</td>
<td>None</td>
</tr>
<tr>
<td>IX (subcommand)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>INSERT_LIMIT_COUNT</td>
<td>control segment placement</td>
<td>ILC</td>
</tr>
<tr>
<td></td>
<td>LOCATION</td>
<td>control segment placement</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SEGMENT</td>
<td>control segment placement</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>WHERE</td>
<td>specify conditional selection</td>
<td>None</td>
</tr>
</tbody>
</table>

**IX (continued)**

**LOADCTL (subcommand)**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Function</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSERT_LIMIT_COUNT</td>
<td>control segment placement</td>
<td>ILC</td>
</tr>
<tr>
<td>LOCATION</td>
<td>control segment placement</td>
<td>None</td>
</tr>
<tr>
<td>SEGMENT</td>
<td>control segment placement</td>
<td>None</td>
</tr>
<tr>
<td>WHERE</td>
<td>specify conditional selection</td>
<td>None</td>
</tr>
</tbody>
</table>
Table 7  PFPSYSIN commands and keywords for PFPMAIN (part 7 of 11)

<table>
<thead>
<tr>
<th>Command or Subcommand</th>
<th>Keyword</th>
<th>Function</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFILECTL (subcommand)</td>
<td>AVGREC</td>
<td>identify space requirements</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DATACLAS</td>
<td>specify SMS data class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DDNAME</td>
<td>specify ddname for file</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DISP</td>
<td>specify data set disposition</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DSNAME</td>
<td>specify data set name mask</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>EXPDT</td>
<td>specify expiration date</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>LIKE</td>
<td>specify model data set</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>MGMTCLAS</td>
<td>specify SMS management class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>OAREA</td>
<td>specify area selection</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>RETPD</td>
<td>specify retention period</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SPACE</td>
<td>specify space requirements</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>STORCLAS</td>
<td>specify SMS storage class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>UNIT</td>
<td>specify physical device</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLCNT</td>
<td>specify number of volumes</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLSER</td>
<td>specify volume serial numbers</td>
<td>None</td>
</tr>
<tr>
<td>OUTPUT (subcommand)</td>
<td>FIELDS</td>
<td>specify record contents</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SEGMENT</td>
<td>specify segment name</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>WHERE</td>
<td>specify conditional selection</td>
<td>None</td>
</tr>
<tr>
<td>PERFORM (subcommand)</td>
<td>SCRIPT</td>
<td>specify procedure for control</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td>interval dump or modification</td>
<td></td>
</tr>
<tr>
<td>PFPSORT</td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td></td>
<td>INPUT_DSN_MASK</td>
<td>input dynamic allocation mask</td>
<td>IDM</td>
</tr>
<tr>
<td></td>
<td>OAREA</td>
<td>specify output area(s)</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>OUTPUT_DSN_MASK</td>
<td>output dynamic allocation mask</td>
<td>ODM</td>
</tr>
<tr>
<td></td>
<td>SORT_OPTION</td>
<td>specify custom sort options</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SORT_SEQUENCE</td>
<td>specify sort sequence parameters</td>
<td>SORTSEQ</td>
</tr>
<tr>
<td>PROCESS_AREA</td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td></td>
<td>IAREA</td>
<td>specify input area(s)</td>
<td>AREA</td>
</tr>
<tr>
<td></td>
<td>INPUT_DSN_MASK</td>
<td>input dynamic allocation mask</td>
<td>IDM</td>
</tr>
</tbody>
</table>
## Table 7  PFPSYSIN commands and keywords for PFPMAIN (part 8 of 11)

<table>
<thead>
<tr>
<th>Command or Subcommand</th>
<th>Keyword</th>
<th>Function</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELOAD</td>
<td>BYPASS_RECORD</td>
<td>control processing of roots randomized to non-processed areas</td>
<td>None</td>
</tr>
<tr>
<td>COMPRESS</td>
<td></td>
<td>specify data compression</td>
<td>CMP</td>
</tr>
<tr>
<td>DBD</td>
<td></td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td>ERROR_THRESHOLD</td>
<td></td>
<td>control pointer error tolerance</td>
<td>ERT</td>
</tr>
<tr>
<td>EXCEPTION_LIMIT</td>
<td></td>
<td>specify exception toleration limit</td>
<td>None</td>
</tr>
<tr>
<td>EXPAND</td>
<td></td>
<td>expand compressed data</td>
<td>None</td>
</tr>
<tr>
<td>IAREA</td>
<td></td>
<td>specify input area(s)</td>
<td>None</td>
</tr>
<tr>
<td>INDEX_THREADS</td>
<td></td>
<td>limit index threads</td>
<td>XTHREADS</td>
</tr>
<tr>
<td>INPUT_DSN_MASK</td>
<td></td>
<td>input dynamic allocation mask</td>
<td>IDM</td>
</tr>
<tr>
<td>INPUT_THREADS</td>
<td></td>
<td>limit input threads</td>
<td>ITHREADS</td>
</tr>
<tr>
<td>IOVF_LOAD_HWM</td>
<td></td>
<td>limit space usage in IOVF blocks</td>
<td>None</td>
</tr>
<tr>
<td>LARGEST_DATABASE_RECORDS</td>
<td></td>
<td>specify number of largest database records</td>
<td>LDR or KEYS</td>
</tr>
<tr>
<td>MESSAGE_SUPPRESSION</td>
<td></td>
<td>suppress repeated messages</td>
<td>MSGSUP</td>
</tr>
<tr>
<td>OAREA</td>
<td></td>
<td>specify output area(s)</td>
<td>None</td>
</tr>
<tr>
<td>OCACHE</td>
<td></td>
<td>specify output cache</td>
<td>None</td>
</tr>
<tr>
<td>ORPHANED_SDEP_MSG</td>
<td></td>
<td>control SDEP error messages</td>
<td>OSM</td>
</tr>
<tr>
<td>OUTPUT_DSN_MASK</td>
<td></td>
<td>output dynamic allocation mask</td>
<td>ODM</td>
</tr>
<tr>
<td>OUTPUT_THREADS</td>
<td></td>
<td>limit output threads</td>
<td>OTHREADS</td>
</tr>
<tr>
<td>POINTER_VALIDATION</td>
<td></td>
<td>request pointer validation</td>
<td>PTR or FPA</td>
</tr>
<tr>
<td>RAP_VALIDATION</td>
<td></td>
<td>request RAP validation</td>
<td>None</td>
</tr>
<tr>
<td>SDEP_PROCESS</td>
<td></td>
<td>specify SDEP processing</td>
<td>SDEP</td>
</tr>
<tr>
<td>SDEP_VALIDATION</td>
<td></td>
<td>request SDEP pointer validation</td>
<td>None</td>
</tr>
<tr>
<td>SORT</td>
<td></td>
<td>sort reload file</td>
<td>None</td>
</tr>
<tr>
<td>SORT_OPTION</td>
<td></td>
<td>specify custom sort options</td>
<td>None</td>
</tr>
<tr>
<td>SUBSET_POINTERS</td>
<td></td>
<td>set subset pointer processing</td>
<td>SSPTR</td>
</tr>
</tbody>
</table>
### Table 7 PFPSYSIN commands and keywords for PFPMAIN (part 9 of 11)

<table>
<thead>
<tr>
<th>Command or Subcommand</th>
<th>Keyword</th>
<th>Function</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>REORGANIZE</strong></td>
<td>COMPRESS</td>
<td>specify data compression</td>
<td>CMP</td>
</tr>
<tr>
<td></td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td></td>
<td>ERROR_THRESHOLD</td>
<td>control pointer error tolerance</td>
<td>ERT</td>
</tr>
<tr>
<td></td>
<td>EXTEND_IOVF_#UOWS</td>
<td>Increase IOVF storage size</td>
<td>XIOVF</td>
</tr>
<tr>
<td></td>
<td>EXTEND_SDEP_#CIS</td>
<td>increase SDEP storage size</td>
<td>XSDEP</td>
</tr>
<tr>
<td></td>
<td>FRAGMENTATION_PERCENT</td>
<td>percent of fragmentation</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>IAREA</td>
<td>specify input area(s)</td>
<td>AREA</td>
</tr>
<tr>
<td></td>
<td>INPUT_DSN_MASK</td>
<td>input dynamic allocation mask</td>
<td>IDM</td>
</tr>
<tr>
<td></td>
<td>IOVF_SAVE_THRESHOLD</td>
<td>specify IOVF threshold</td>
<td>ISAVE</td>
</tr>
<tr>
<td></td>
<td>LARGEST_DATABASE_RECORDS</td>
<td>specify number of largest database records</td>
<td>LDR KEYS</td>
</tr>
<tr>
<td></td>
<td>MESSAGE_SUPPRESSION</td>
<td>suppress repetitious messages</td>
<td>MSGSUP</td>
</tr>
<tr>
<td></td>
<td>ORPHANED_SDEP_MSG</td>
<td>control SDEP error messages</td>
<td>OSM</td>
</tr>
<tr>
<td></td>
<td>POINTER_VALIDATION</td>
<td>request pointer validation</td>
<td>PTR or FPA</td>
</tr>
<tr>
<td></td>
<td>RAP_VALIDATION</td>
<td>request RAP validation</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SDEP_VALIDATION</td>
<td>request SDEP pointer validation</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SELECT_UOW</td>
<td>limit UOWs selected and allocate maximum buffers</td>
<td>None</td>
</tr>
<tr>
<td><strong>REPORT (subcommand)</strong></td>
<td>FREESPACE_ANALYSIS</td>
<td>request reports</td>
<td>FSA</td>
</tr>
<tr>
<td></td>
<td>IOVF_SPACE_ANALYSIS</td>
<td>request reports</td>
<td>ISA</td>
</tr>
<tr>
<td></td>
<td>POINTER_ANALYSIS</td>
<td>request reports</td>
<td>PA or PTR</td>
</tr>
<tr>
<td></td>
<td>RECORD_LENGTH_ANALYSIS</td>
<td>request reports</td>
<td>RLA</td>
</tr>
<tr>
<td></td>
<td>RECORD_LENGTH_INCREMENT</td>
<td>control reporting interval</td>
<td>RLI</td>
</tr>
<tr>
<td></td>
<td>RECORD_PLACEMENT_ANALYSIS</td>
<td>request reports</td>
<td>RPLA</td>
</tr>
<tr>
<td></td>
<td>RECORD_PROFILE_ANALYSIS</td>
<td>request reports</td>
<td>RPRA</td>
</tr>
<tr>
<td></td>
<td>REPORT_DDNNAME</td>
<td>specify report destination</td>
<td>RPTTDD</td>
</tr>
<tr>
<td></td>
<td>REPORT_DEFAULT</td>
<td>set report default</td>
<td>DEFAULT</td>
</tr>
<tr>
<td></td>
<td>REPORT_HEADING</td>
<td>specify report heading</td>
<td>RPTH</td>
</tr>
<tr>
<td></td>
<td>REPORT_LINE_COUNT</td>
<td>specify report page size</td>
<td>RPTLC</td>
</tr>
<tr>
<td></td>
<td>SEGMENT_IO_ANALYSIS</td>
<td>request reports</td>
<td>SIA</td>
</tr>
<tr>
<td></td>
<td>SEGMENT_LENGTH_ANALYSIS</td>
<td>request reports</td>
<td>SLA</td>
</tr>
<tr>
<td></td>
<td>SEGMENT_PLACEMENT_ANALYSIS</td>
<td>request reports</td>
<td>SPLA</td>
</tr>
<tr>
<td></td>
<td>STARTUOW</td>
<td>specify UOW range</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>STOPUOW</td>
<td>specify UOW range</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SYNONYM_CHAIN_ANALYSIS</td>
<td>request reports</td>
<td>SCA</td>
</tr>
<tr>
<td>Command or Subcommand</td>
<td>Keyword</td>
<td>Function</td>
<td>Alias</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------</td>
<td>----------</td>
<td>-------</td>
</tr>
<tr>
<td>REPORT (subcommand) (continued)</td>
<td>SYNONYM_CHAIN_INCREMENT</td>
<td>control reporting interval</td>
<td>SCI</td>
</tr>
<tr>
<td></td>
<td>UOW_DETAILED_ANALYSIS</td>
<td>request reports</td>
<td>UDA</td>
</tr>
<tr>
<td>RESYNC</td>
<td>CHECKPOINT</td>
<td>specify checkpoint frequency</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td></td>
<td>IAREA</td>
<td>specify input area(s)</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>INDEX_THREADS</td>
<td>limit index threads</td>
<td>XTHREADS</td>
</tr>
<tr>
<td></td>
<td>INPUT_DSN_MASK</td>
<td>input dynamic allocation mask</td>
<td>IDM</td>
</tr>
<tr>
<td></td>
<td>MESSAGE_SUPPRESSION</td>
<td>suppress repetitious messages</td>
<td>MSGSUP</td>
</tr>
<tr>
<td></td>
<td>SORT</td>
<td>sort index database</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SORT_OPTION</td>
<td>specify custom sort option</td>
<td>None</td>
</tr>
<tr>
<td>RETRIEVE</td>
<td>HISTORY_DDNAME</td>
<td>specify statistics extract</td>
<td>HISDD</td>
</tr>
<tr>
<td></td>
<td>SELECT_AREA</td>
<td>specify area(s)</td>
<td>AREA</td>
</tr>
<tr>
<td></td>
<td>SELECT_DATE</td>
<td>specify date range(s)</td>
<td>DATE</td>
</tr>
<tr>
<td></td>
<td>SELECT_DB</td>
<td>specify database(s)</td>
<td>DBD</td>
</tr>
<tr>
<td></td>
<td>SELECT_GROUP</td>
<td>specify group(s)</td>
<td>GROUP</td>
</tr>
<tr>
<td></td>
<td>SELECT_LIMIT</td>
<td>limit number of entries</td>
<td>LIMIT</td>
</tr>
<tr>
<td>THRESHOLD (subcommand)</td>
<td>DOVF_FREESPACE_PERCENT</td>
<td>set threshold level</td>
<td>DFP</td>
</tr>
<tr>
<td></td>
<td>FREESPACEDOVFDOVF</td>
<td>set threshold level</td>
<td>FDI</td>
</tr>
<tr>
<td></td>
<td>FREESPACERAA_DOVF</td>
<td>set threshold level</td>
<td>FRD</td>
</tr>
<tr>
<td></td>
<td>FREESPACERAA_IOVF</td>
<td>set threshold level</td>
<td>FRI</td>
</tr>
<tr>
<td></td>
<td>IOVF_FREESPACETPERCENT</td>
<td>set threshold level</td>
<td>IFP</td>
</tr>
<tr>
<td></td>
<td>IOVF_USED_PERCENT</td>
<td>set threshold level</td>
<td>IUP</td>
</tr>
<tr>
<td></td>
<td>RAA_FREESPACETPERCENT</td>
<td>set threshold level</td>
<td>RFP</td>
</tr>
<tr>
<td></td>
<td>RAP_OVERFLOW_PERCENT</td>
<td>set threshold level</td>
<td>ROP</td>
</tr>
<tr>
<td></td>
<td>RECORD_IO_AVERAGE</td>
<td>set threshold level</td>
<td>RECIOA</td>
</tr>
<tr>
<td></td>
<td>RECORD_IO_MAXIMUM</td>
<td>set threshold level</td>
<td>RECIOM</td>
</tr>
<tr>
<td></td>
<td>RECORD_IOVF_PERCENT</td>
<td>set threshold level</td>
<td>RIP</td>
</tr>
<tr>
<td></td>
<td>ROOT_IO_AVERAGE</td>
<td>set threshold level</td>
<td>RTIOA</td>
</tr>
<tr>
<td></td>
<td>ROOT_IO_MAXIMUM</td>
<td>set threshold level</td>
<td>RTIOM</td>
</tr>
<tr>
<td></td>
<td>SYNONYM_CHAIN_LENGTH</td>
<td>set threshold level</td>
<td>SCL</td>
</tr>
<tr>
<td></td>
<td>SYNONYM_CHAIN_MAXIMUM</td>
<td>set threshold level</td>
<td>SCM</td>
</tr>
<tr>
<td></td>
<td>SYNONYM_CHAIN_PERCENT</td>
<td>set threshold level</td>
<td>SCP</td>
</tr>
<tr>
<td></td>
<td>UOW_DOVF_PERCENT</td>
<td>set threshold level</td>
<td>UDP</td>
</tr>
<tr>
<td></td>
<td>UOW_IOVF_AVERAGE</td>
<td>set threshold level</td>
<td>UIA</td>
</tr>
<tr>
<td></td>
<td>UOW_IOVF_MAXIMUM</td>
<td>set threshold level</td>
<td>UIM</td>
</tr>
<tr>
<td></td>
<td>UOW_IOVF_PERCENT</td>
<td>set threshold level</td>
<td>UIP</td>
</tr>
</tbody>
</table>
### PFPSYSIN commands and keywords for PFPMAIN (part 11 of 11)

<table>
<thead>
<tr>
<th>Command or Subcommand</th>
<th>Keyword</th>
<th>Function</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNLOAD</td>
<td>BYPASS_RECORD</td>
<td>control processing of roots randomized to non-processed areas</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>COMPRESS</td>
<td>specify data compression</td>
<td>CMP</td>
</tr>
<tr>
<td></td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td></td>
<td>ERROR_THRESHOLD</td>
<td>control pointer error tolerance</td>
<td>ERT</td>
</tr>
<tr>
<td></td>
<td>EXPAND</td>
<td>expand compressed data</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>FORMAT</td>
<td>specify unload format</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>IAREA</td>
<td>specify input area(s)</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>ICACHE</td>
<td>specify input cache</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>INPUT_DSN_MASK</td>
<td>input dynamic allocation mask</td>
<td>IDM</td>
</tr>
<tr>
<td></td>
<td>INPUT_THREADS</td>
<td>limit input threads per task</td>
<td>ITHREADS</td>
</tr>
<tr>
<td></td>
<td>MESSAGE_SUPPRESSION</td>
<td>suppress repetitious messages</td>
<td>MSGSUP</td>
</tr>
<tr>
<td></td>
<td>OAREA</td>
<td>specify output area(s)</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>OUTPUT_DSN_MASK</td>
<td>output dynamic allocation mask</td>
<td>ODM</td>
</tr>
<tr>
<td></td>
<td>SDEP_PROCESS</td>
<td>process SDEP segments</td>
<td>SDEP</td>
</tr>
<tr>
<td></td>
<td>SUBSET_POINTERS</td>
<td>set subset pointer processing</td>
<td>SSPTR</td>
</tr>
<tr>
<td>USER_RECORD (subcommand)</td>
<td>BREAK</td>
<td>specify trigger event</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>FIELDS</td>
<td>specify record contents</td>
<td>None</td>
</tr>
<tr>
<td>VERIFY</td>
<td>CHECKPOINT</td>
<td>specify checkpoint frequency</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td></td>
<td>IAREA</td>
<td>specify input area(s)</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>ICACHE</td>
<td>specify input cache</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>INDEX_THREADS</td>
<td>limit index threads</td>
<td>XTHREADS</td>
</tr>
<tr>
<td></td>
<td>INPUT_DSN_MASK</td>
<td>input dynamic allocation mask</td>
<td>IDM</td>
</tr>
<tr>
<td></td>
<td>INPUT_THREADS</td>
<td>limit input threads</td>
<td>ITHREADS</td>
</tr>
<tr>
<td></td>
<td>MESSAGE_SUPPRESSION</td>
<td>suppress repetitious messages</td>
<td>MSGSUP</td>
</tr>
<tr>
<td></td>
<td>SORT</td>
<td>sort index database</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SORT_OPTION</td>
<td>specify custom sort option</td>
<td>None</td>
</tr>
<tr>
<td>XSCAN</td>
<td>DBD</td>
<td>specify database</td>
<td>DBD</td>
</tr>
<tr>
<td></td>
<td>IAREA</td>
<td>specify input area(s)</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>ICACHE</td>
<td>specify input cache</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>INPUT_DSN_MASK</td>
<td>input dynamic allocation mask</td>
<td>IDM</td>
</tr>
<tr>
<td></td>
<td>SORT</td>
<td>sort index database</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SORT_OPTION</td>
<td>specify custom sort option</td>
<td>None</td>
</tr>
</tbody>
</table>
PFPSYSIN commands and subcommands for PFCMAIN

Table 8 shows PFPSYSIN commands available for execution by PFCMAIN, the subcommands available for each command, and the processing modes in which they are available.

Table 8  PFPSYSIN commands and subcommands for PFCMAIN

<table>
<thead>
<tr>
<th>Command</th>
<th>Subcommand</th>
<th>Mode Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>BACKOUT</td>
<td>PLAN_FILECTL</td>
<td>online only&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>PREPARE</td>
<td>ACTIVITY_FILECTL</td>
<td>online only&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>OUTPUT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PLAN_FILECTL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>REGISTER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>REPORT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>THRESHOLD</td>
<td></td>
</tr>
<tr>
<td>RESTART</td>
<td>PLAN_FILECTL</td>
<td>online only&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>RESTRUCTURE</td>
<td>PLAN_FILECTL</td>
<td>online only&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>SHADOW_INIT</td>
<td>ALLOCATE</td>
<td>offline only</td>
</tr>
<tr>
<td></td>
<td>PLAN_FILECTL</td>
<td></td>
</tr>
<tr>
<td>STATUS</td>
<td>PLAN_FILECTL</td>
<td>online only&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

<sup>a</sup> This command is executed by using offline JCL.
PFPSYSIN keywords for PFCMAIN

Table 9 shows the keywords available for PFCMAIN commands and subcommands, including a brief description of use and keyword aliases. Refer to Table 8 for a list of subcommands available for each command.

### Table 9  PFPSYSIN commands and keywords for PFCMAIN (part 1 of 7)

<table>
<thead>
<tr>
<th>Command or subcommand</th>
<th>Keyword</th>
<th>Function</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ACTIVITY_FILECTL</strong></td>
<td>AVGREC</td>
<td>identify space requirements</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DATACLAS</td>
<td>specify SMS data class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DDNAME</td>
<td>specify ddname for file</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DISP</td>
<td>specify data set disposition</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DSNAMES</td>
<td>specify data set name mask</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>EXPDT</td>
<td>specify expiration data</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>LIKE</td>
<td>specify model data set</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>MGMTCLAS</td>
<td>specify SMS management class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>RETPD</td>
<td>specify retention period</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SPACE</td>
<td>specify space requirement</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>STORCLAS</td>
<td>specify SMS storage class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>UNIT</td>
<td>specify physical device type</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLCNT</td>
<td>specify space requirements</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLSER</td>
<td>specify volume serial number</td>
<td>None</td>
</tr>
<tr>
<td><strong>ALLOCATE</strong></td>
<td>AVGREC</td>
<td>identify space requirements</td>
<td>None</td>
</tr>
<tr>
<td>(subcommand)</td>
<td>CONFIGURE_AREA</td>
<td>specify area data set allocation characteristics</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>CONFIGURE_IOVF</td>
<td>specify IOVF allocation characteristics</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>CONFIGURE_RAA</td>
<td>specify RAA allocation characteristics</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>CONFIGURE_SDEP</td>
<td>specify SDEP allocation characteristics</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DATACLAS</td>
<td>specify SMS data class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DATACLAS2</td>
<td>specify SMS data class for secondary shadow area data set</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>MGMTCLAS</td>
<td>specify SMS management class</td>
<td>None</td>
</tr>
</tbody>
</table>
### Table 9  PFPSYSIN commands and keywords for PFCMAIN (part 2 of 7)

<table>
<thead>
<tr>
<th>Command or subcommand</th>
<th>Keyword</th>
<th>Function</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLOCATE (subcommand) (continued)</td>
<td>MGMTCLAS2</td>
<td>specify SMS management class for secondary shadow area data set</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>OAREA</td>
<td>specify output area(s)</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>REPORT_DDNAME</td>
<td>specify report destination</td>
<td>RPTDD</td>
</tr>
<tr>
<td></td>
<td>SPACE</td>
<td>specify space requirements</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>STORCLAS</td>
<td>specify SMS storage class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>STORCLAS2</td>
<td>specify SMS storage class for secondary shadow area data set</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLSER</td>
<td>specify volume serial number</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLSER2</td>
<td>specify volume serial number for secondary shadow area data set</td>
<td>None</td>
</tr>
<tr>
<td>BACKOUT</td>
<td>ACCESS</td>
<td>specify area access mode</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>ACCESS</td>
<td>specify area access mode</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DBRC</td>
<td>request DBRC for PFP</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>EARLY_TERMINATION</td>
<td>specify when to terminate post-processing during the online restructure process</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>FLOWER_BOX</td>
<td>specify flower box border</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>HISTORY_DDNAME</td>
<td>specify statistics extract</td>
<td>HISDD</td>
</tr>
<tr>
<td></td>
<td>LARGEST_DATABASE_RECORDS</td>
<td>specify number of largest database records</td>
<td>LDR KEYS</td>
</tr>
<tr>
<td></td>
<td>MESSAGE_SUPPRESSION</td>
<td>suppress repetitious messages</td>
<td>MSGSUP</td>
</tr>
<tr>
<td></td>
<td>ORPHANED_SDEP_MSG</td>
<td>control SDEP error messages</td>
<td>OSM</td>
</tr>
<tr>
<td></td>
<td>OUTAGE_WINDOW</td>
<td>specify outage time for database</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>POINTER_VALIDATION</td>
<td>request pointer validation</td>
<td>PTR or FPA</td>
</tr>
<tr>
<td></td>
<td>RAP_VALIDATION</td>
<td>request RAP validation</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SCAN</td>
<td>request syntax scan</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SDEP_VALIDATION</td>
<td>request SDEP pointer validation</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>TYPE_RUN</td>
<td>specify execution mode</td>
<td>None</td>
</tr>
<tr>
<td>OUTPUT (subcommand)</td>
<td>FIELDS</td>
<td>specify record contents</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SEGMENT</td>
<td>specify segment name</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>WHERE</td>
<td>specify conditional selection</td>
<td>None</td>
</tr>
</tbody>
</table>
Table 9  PFPSYSIN commands and keywords for PFCMAIN (part 3 of 7)

<table>
<thead>
<tr>
<th>Command or subcommand</th>
<th>Keyword</th>
<th>Function</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLAN_FILECTL (subcommand)</td>
<td>AVGREC</td>
<td>identify space requirements</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DATACLAS</td>
<td>specify SMS data class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DDNAME</td>
<td>specify ddname for file</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DISP</td>
<td>specify data set disposition</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DSNAME</td>
<td>specify data set name mask</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>EXPDT</td>
<td>specify expiration data</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>LIKE</td>
<td>specify model data set</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>MGMTCLAS</td>
<td>specify SMS management class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>RETPD</td>
<td>specify retention period</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SPACE</td>
<td>specify space requirement</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>STORCLAS</td>
<td>specify SMS storage class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>UNIT</td>
<td>specify physical device type</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLCNT</td>
<td>specify space requirements</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLSER</td>
<td>specify volume serial number</td>
<td>None</td>
</tr>
<tr>
<td>PREPARE</td>
<td>ACCESS</td>
<td>specify area access mode</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td></td>
<td>HISTORY_DDNAME</td>
<td>specify statistics extract</td>
<td>HISDD</td>
</tr>
<tr>
<td></td>
<td>LARGEST_DATABASE_RECORDS</td>
<td>specify number of largest database records</td>
<td>LDR KEYS</td>
</tr>
<tr>
<td></td>
<td>MESSAGE_SUPPRESSION</td>
<td>suppress repeated messages</td>
<td>MSGSUP</td>
</tr>
<tr>
<td></td>
<td>ORPHANED_SDEP_MSG</td>
<td>control SDEP error messages</td>
<td>OSM</td>
</tr>
<tr>
<td></td>
<td>OUTAGE_WINDOW</td>
<td>specify outage time for database</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>POINTER_VALIDATION</td>
<td>request pointer validation</td>
<td>PTR or FPA</td>
</tr>
<tr>
<td></td>
<td>RAP_VALIDATION</td>
<td>request RAP validation</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>REQUIRE_AREA</td>
<td>select input area(s)</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>RETAINED_SUFFIX</td>
<td>rename original area data set</td>
<td>ORIGINAL_SUFFIX</td>
</tr>
<tr>
<td></td>
<td>SDEP_VALIDATION</td>
<td>request SDEP pointer validation</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SHADOW_SUFFIX</td>
<td>create name for shadow area data set</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SHADOW2_DSNAME</td>
<td>specify data set name mask for secondary shadow area data set</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SHADOW2_SUFFIX</td>
<td>create name for secondary shadow area data set</td>
<td>None</td>
</tr>
</tbody>
</table>
### Table 9 PFPSYSIN commands and keywords for PFCMAIN (part 4 of 7)

<table>
<thead>
<tr>
<th>Command or subcommand</th>
<th>Keyword</th>
<th>Function</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGISTER (subcommand)</td>
<td>ADDN</td>
<td>specify the DDname for the area data set</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>ADSN</td>
<td>specify an area data set name</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>CFSTR1</td>
<td>specify the name of the first coupling facility structure for the identified VSO area</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>CFSTR2</td>
<td>specify the name of the second coupling facility structure for the identified VSO area</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DEFLTJCL</td>
<td>specify an implicit skeletal JCL default member for the DBDS</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>FULLSEG</td>
<td>specify whether the full segment is logged in the X’5950’ log record when the segment is updated</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>GENMAX</td>
<td>specify the maximum number of image copies that DBRC should maintain for the specified DBD</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>GSGNAME</td>
<td>specify the global service group to which a database is assigned</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>ICJCL</td>
<td>specify the name of a partitioned data set member that contains skeletal JCL</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>LKASID</td>
<td>specify local data caching for the specified area is to be used for buffer lookaside on read requests</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>MAS</td>
<td>specify that the area should reside in a multi-area coupling facility structure</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>OAREA</td>
<td>specify one or more output areas, a range of output areas, or a combination of output areas and output area ranges to the process</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>PRELOAD</td>
<td>specify that the area should be loaded into the data space or coupling facility structure the next time that space or structure is opened</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>PREOPEN</td>
<td>specify whether to pre-open an area</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>RECOVJCL</td>
<td>specify the partitioned data set member that contains skeletal JCL</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>RECOVPD</td>
<td>specify the recovery period for a specified DBDS or DEDB area</td>
<td>None</td>
</tr>
</tbody>
</table>
Table 9  PFPSYSIN commands and keywords for PFCMAIN (part 5 of 7)

<table>
<thead>
<tr>
<th>Command or subcommand</th>
<th>Keyword</th>
<th>Function</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGISTER (subcommand)</td>
<td>RECVJCL</td>
<td>specify the partitioned data set member that contains skeletal JCL</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>REUSE</td>
<td>specify whether the supported image copy utilities should reuse image copy data sets that were previously used</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>TRACK</td>
<td>specify the type of Remote Site Recovery (RSR) tracking (shadowing) for an area that is assigned to a global service group</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VSO</td>
<td>specify whether an area should reside in virtual storage the next time the control region is initialized or when the next /START AREA command is processed</td>
<td>None</td>
</tr>
<tr>
<td>REPORT (subcommand)</td>
<td>FREESPACE_ANALYSIS</td>
<td>request reports</td>
<td>FSA</td>
</tr>
<tr>
<td></td>
<td>IOVF_SPACE_ANALYSIS</td>
<td>request reports</td>
<td>ISA</td>
</tr>
<tr>
<td></td>
<td>POINTER_ANALYSIS</td>
<td>request reports</td>
<td>PA or PTR</td>
</tr>
<tr>
<td></td>
<td>RECORD_LENGTH_ANALYSIS</td>
<td>request reports</td>
<td>RLA</td>
</tr>
<tr>
<td></td>
<td>RECORD_LENGTH_INCREMENT</td>
<td>control reporting interval</td>
<td>RLI</td>
</tr>
<tr>
<td></td>
<td>RECORD-placement-ANALYSIS</td>
<td>request reports</td>
<td>RPLA</td>
</tr>
<tr>
<td></td>
<td>RECORD_PROFILE_ANALYSIS</td>
<td>request reports</td>
<td>RPRA</td>
</tr>
<tr>
<td></td>
<td>REPORT_DDNAME</td>
<td>specify report destination</td>
<td>RPTDD</td>
</tr>
<tr>
<td></td>
<td>REPORT_DEFAULT</td>
<td>set report default</td>
<td>DEFAULT</td>
</tr>
<tr>
<td></td>
<td>REPORT_HEADING</td>
<td>specify report heading</td>
<td>RPTH</td>
</tr>
<tr>
<td></td>
<td>REPORT_LINE_COUNT</td>
<td>specify report page size</td>
<td>RPTLC</td>
</tr>
<tr>
<td></td>
<td>SEGMENT_IO_COUNT</td>
<td>request reports</td>
<td>SIA</td>
</tr>
<tr>
<td></td>
<td>SEGMENT_LENGTH_ANALYSIS</td>
<td>request reports</td>
<td>SLA</td>
</tr>
<tr>
<td></td>
<td>SEGMENT_placement-ANALYSIS</td>
<td>request reports</td>
<td>SPLA</td>
</tr>
<tr>
<td></td>
<td>STARTUOW</td>
<td>specify UOW range</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>STOPUOW</td>
<td>specify UOW range</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SYNONYM_CHAIN_ANALYSIS</td>
<td>request reports</td>
<td>SCA</td>
</tr>
<tr>
<td></td>
<td>SYNONYM_CHAIN_INCREMENT</td>
<td>control reporting interval</td>
<td>SCI</td>
</tr>
<tr>
<td></td>
<td>UOWDetalle ANALYSIS</td>
<td>request reports</td>
<td>UDA</td>
</tr>
</tbody>
</table>
### Table 9  PFPSYSIN commands and keywords for PFCMAIN (part 6 of 7)

<table>
<thead>
<tr>
<th>Command or subcommand</th>
<th>Keyword</th>
<th>Function</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESTART</td>
<td>ACCESS</td>
<td>specify area access mode</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td></td>
<td>DETAIL</td>
<td>specify items that require additional details to be produced on the Restructure Activity Report</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>EARLY_TERMINATION</td>
<td>specify when to terminate post-processing during an online restructure restart</td>
<td>None</td>
</tr>
<tr>
<td>RESTRUCTURE</td>
<td>ACCESS</td>
<td>specify area access mode</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td></td>
<td>DETAIL</td>
<td>specify items that require additional details to be produced on the Restructure Activity Report</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>EARLY_TERMINATION</td>
<td>specify when to terminate post-processing during the online restructure process</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>MESSAGE_SUPPRESSION</td>
<td>suppress repetitious messages</td>
<td>MSGSUP</td>
</tr>
<tr>
<td></td>
<td>OUTAGE_WINDOW</td>
<td>specify outage time for database</td>
<td>None</td>
</tr>
<tr>
<td>SHADOW_INIT</td>
<td>ACCESS</td>
<td>specify area access mode</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td>STATUS</td>
<td>ACCESS</td>
<td>specify area access mode</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DBD</td>
<td>specify database</td>
<td>DBDNAME</td>
</tr>
<tr>
<td></td>
<td>DETAIL</td>
<td>specify items that require additional details to be produced on the Restructure Activity Report</td>
<td>None</td>
</tr>
</tbody>
</table>
Table 10 shows the commands and subcommands available for execution by the PFPEPR00 batch utility program that is used for creating and maintaining allocation rules in the Fast Path/EP statistics repository catalog.
Table 10  PFPSYSIN commands and subcommands for PFPEPR00 allocation records

<table>
<thead>
<tr>
<th>Command or Subcommand</th>
<th>Keyword</th>
<th>Function</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESS_EPR</td>
<td>REPOSITORY_DSNAME</td>
<td>specify repository catalog data set</td>
<td>EPR</td>
</tr>
<tr>
<td>ADD type_allocation</td>
<td>AREA_KEY</td>
<td>specify area identifier</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DATACLAS</td>
<td>specify SMS data class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DBD_KEY</td>
<td>specify database identifier</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DETAIL_LEVEL</td>
<td>specify level of detail to be written into</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td>repository statistics data set</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DSNAME</td>
<td>specify data set name mask</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>EXPDT</td>
<td>specify expiration date</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>GROUP_KEY</td>
<td>specify group identifier</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>MGMTCLAS</td>
<td>specify SMS management class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>RETPD</td>
<td>specify retention period</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>STORCLAS</td>
<td>specify SMS storage class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>UNIT</td>
<td>specify physical device type</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLSER</td>
<td>specify volume serial number</td>
<td>None</td>
</tr>
<tr>
<td>DELETE type_allocation</td>
<td>SELECT_AREA</td>
<td>specify area name</td>
<td>AREA</td>
</tr>
<tr>
<td></td>
<td>SELECT_DBD</td>
<td>specify database name</td>
<td>DBD</td>
</tr>
<tr>
<td></td>
<td>SELECT_GROUP</td>
<td>specify repository group</td>
<td>GROUP</td>
</tr>
<tr>
<td>LIST type_allocation</td>
<td>SELECT_AREA</td>
<td>specify area name</td>
<td>AREA</td>
</tr>
<tr>
<td></td>
<td>SELECT_DBD</td>
<td>specify database name</td>
<td>DBD</td>
</tr>
<tr>
<td></td>
<td>SELECT_GROUP</td>
<td>specify repository group</td>
<td>GROUP</td>
</tr>
<tr>
<td>MODIFY type_allocation</td>
<td>DATACLAS</td>
<td>specify SMS data class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>DETAIL_LEVEL</td>
<td>specify level of detail to be written into</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td>repository statistics data set</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DSNAME</td>
<td>specify data set name mask</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>EXPDT</td>
<td>specify expiration date</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>MGMTCLAS</td>
<td>specify SMS management class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>RETPD</td>
<td>specify retention period</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SELECT_AREA</td>
<td>specify area name</td>
<td>AREA</td>
</tr>
<tr>
<td></td>
<td>SELECT_DBD</td>
<td>specify database name</td>
<td>DBD</td>
</tr>
<tr>
<td></td>
<td>SELECT_GROUP</td>
<td>specify repository group</td>
<td>GROUP</td>
</tr>
<tr>
<td>MODIFY type_allocation</td>
<td>STORCLAS</td>
<td>specify SMS storage class</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>UNIT</td>
<td>specify physical device type</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>VOLSER</td>
<td>specify volume serial number</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 11 shows the commands and subcommands available for execution by the PFPEPR00 batch utility program when used for maintaining and retrieving statistics data sets in a Fast Path/EP statistics repository.
### Table 11 PFPSYSIN commands and subcommands for PFPEPR00 statistics records

<table>
<thead>
<tr>
<th>Command or Subcommand</th>
<th>Keyword</th>
<th>Function</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESS_EPR</td>
<td>REPOSITORY_DSNAME</td>
<td>specify repository catalog data set</td>
<td>EPR</td>
</tr>
<tr>
<td>ADD STATISTICS (subcommand)</td>
<td>DSNAME</td>
<td>specify data set name mask</td>
<td>None</td>
</tr>
<tr>
<td>DELETE STATISTICS (subcommand)</td>
<td>REPOSITORY_RETENTION_PERIOD</td>
<td>specify retention period</td>
<td>RETPD</td>
</tr>
<tr>
<td></td>
<td>SELECT_AREA</td>
<td>specify area name</td>
<td>AREA</td>
</tr>
<tr>
<td></td>
<td>SELECT_DBD</td>
<td>specify database name</td>
<td>DBD</td>
</tr>
<tr>
<td></td>
<td>SELECT_GROUP</td>
<td>specify repository group</td>
<td>GROUP</td>
</tr>
<tr>
<td>LIST STATISTICS (subcommand)</td>
<td>HISTORY_DDNAME</td>
<td>specify history file</td>
<td>HISDD</td>
</tr>
<tr>
<td></td>
<td>SELECT_AREA</td>
<td>specify area name</td>
<td>AREA</td>
</tr>
<tr>
<td></td>
<td>SELECT_DATE</td>
<td>specify date and time range</td>
<td>DATE</td>
</tr>
<tr>
<td></td>
<td>SELECT_DBD</td>
<td>specify database name</td>
<td>DBD</td>
</tr>
<tr>
<td></td>
<td>SELECT_GROUP</td>
<td>specify repository group</td>
<td>GROUP</td>
</tr>
<tr>
<td>REPORT (subcommand)</td>
<td>FREESPACE_ANALYSIS</td>
<td>request reports</td>
<td>FSA</td>
</tr>
<tr>
<td></td>
<td>IOVF_SPACE_ANALYSIS</td>
<td>request reports</td>
<td>ISA</td>
</tr>
<tr>
<td></td>
<td>RECORD_LENGTH_ANALYSIS</td>
<td>request reports</td>
<td>RLA</td>
</tr>
<tr>
<td></td>
<td>RECORD_LENGTH-INCREMENT</td>
<td>control reporting interval</td>
<td>RLI</td>
</tr>
<tr>
<td></td>
<td>RECORD_PLACEMENT_ANALYSIS</td>
<td>request reports</td>
<td>RPLA</td>
</tr>
<tr>
<td></td>
<td>RECORD_PROFILE_ANALYSIS</td>
<td>request reports</td>
<td>RPRA</td>
</tr>
<tr>
<td></td>
<td>REPORT_DDNAME</td>
<td>specify report destination</td>
<td>RPTDD</td>
</tr>
<tr>
<td></td>
<td>REPORT_DEFAULT</td>
<td>set report default</td>
<td>DEFAULT</td>
</tr>
<tr>
<td></td>
<td>REPORT_HEADING</td>
<td>specify report heading</td>
<td>RPTH</td>
</tr>
<tr>
<td></td>
<td>REPORT_LINE_COUNT</td>
<td>specify report page size</td>
<td>RPTLC</td>
</tr>
<tr>
<td></td>
<td>SEGMENT_IO_ANALYSIS</td>
<td>request reports</td>
<td>SIA</td>
</tr>
<tr>
<td></td>
<td>SEGMENT_LENGTH_ANALYSIS</td>
<td>request reports</td>
<td>SLA</td>
</tr>
<tr>
<td></td>
<td>SEGMENT_PLACEMENT_ANALYSIS</td>
<td>request reports</td>
<td>SPLA</td>
</tr>
<tr>
<td></td>
<td>STARTUOW</td>
<td>specify UOW range</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>STOPUOW</td>
<td>specify UOW range</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>SYNONYM_CHAIN_ANALYSIS</td>
<td>request reports</td>
<td>SCA</td>
</tr>
<tr>
<td></td>
<td>SYNONYM_CHAIN_INCREMENT</td>
<td>control reporting interval</td>
<td>SCI</td>
</tr>
<tr>
<td></td>
<td>UOW_DETAILED_ANALYSIS</td>
<td>request reports</td>
<td>UDA</td>
</tr>
<tr>
<td>RETRIEVE (subcommand)</td>
<td>HISTORY_DDNAME</td>
<td>specify history file</td>
<td>HISDD</td>
</tr>
<tr>
<td></td>
<td>SELECT_AREA</td>
<td>specify area name</td>
<td>AREA</td>
</tr>
<tr>
<td></td>
<td>SELECT_DATE</td>
<td>specify date and time range</td>
<td>DATE</td>
</tr>
<tr>
<td></td>
<td>SELECT_DBD</td>
<td>specify database name</td>
<td>DBD</td>
</tr>
<tr>
<td></td>
<td>SELECT_GROUP</td>
<td>specify repository group</td>
<td>GROUP</td>
</tr>
<tr>
<td></td>
<td>SELECT_LIMIT</td>
<td>limit number of entries</td>
<td>LIMIT</td>
</tr>
</tbody>
</table>
Table 12 shows the commands and subcommands available for execution by the PFPEPR00 batch utility program when used for customizing selected product messages and storing the customizations in the Fast Path/EP statistics repository.

Table 12  PFPSYSIN commands and subcommands for PFPEPR00 message customization

<table>
<thead>
<tr>
<th>Command or subcommand</th>
<th>Keyword</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESS_EPR</td>
<td>REPOSITORY_DSNAME</td>
<td>specify repository catalog data set</td>
</tr>
<tr>
<td>LIST (subcommand)</td>
<td>MESSAGE_OVERRIDE</td>
<td>list all message overrides stored in repository</td>
</tr>
<tr>
<td>OVERRIDE (subcommand)</td>
<td>MESSAGE_LEVEL</td>
<td>change severity level for message</td>
</tr>
<tr>
<td></td>
<td>MESSAGE_LIMIT</td>
<td>set threshold limit for suppression of messages</td>
</tr>
<tr>
<td></td>
<td>MESSAGE_NUMBER</td>
<td>specify product ID number of message to be customized</td>
</tr>
<tr>
<td>RESET (subcommand)</td>
<td>MESSAGE_NUMBER</td>
<td>delete customizations and return messages to product defaults</td>
</tr>
</tbody>
</table>

Fast Path/EP operator interface commands

Fast Path/EP Series products, when running in offline mode only, accept commands from the system operator using the MVS MODIFY (F) operator command. These commands are intended for diagnostic purposes. A sample command is shown in Figure 13.

Figure 13  Example operator interface command

```
F JobName,DUMP DUMP_TYPE=ABEND
```

Table 13 shows the operator interface commands, including a brief description of the keywords and their use.

Table 13  Fast Path/EP operator interface commands and keywords (part 1 of 2)

<table>
<thead>
<tr>
<th>Command or subcommand</th>
<th>Keyword</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY</td>
<td>ID</td>
<td>specify task for display</td>
</tr>
<tr>
<td></td>
<td>OBJECT</td>
<td>select display items</td>
</tr>
<tr>
<td>DUMP</td>
<td>DUMP_TYPE</td>
<td>specify dump type</td>
</tr>
<tr>
<td>HELP</td>
<td>None</td>
<td>request command help</td>
</tr>
<tr>
<td>SET</td>
<td>OUTAGE_WINDOW</td>
<td>specify outage window time</td>
</tr>
</tbody>
</table>
Table 13  Fast Path/EP operator interface commands and keywords (part 2 of 2)

<table>
<thead>
<tr>
<th>Command or subcommand</th>
<th>Keyword</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHUTDOWN</td>
<td>FORCE</td>
<td>force a dump</td>
</tr>
<tr>
<td></td>
<td>ID</td>
<td>specify task to stop</td>
</tr>
<tr>
<td></td>
<td>PRODUCT</td>
<td>specify product to stop</td>
</tr>
</tbody>
</table>

Command language description format

The following headings are used on the following command, subcommand, and keyword description pages:

Purpose

Provides a brief description of the purpose of the command, subcommand, or keyword.

Use

Indicates whether the command, subcommand, or keyword is optional or required, and provides qualifications of use.

Available keywords

Lists all keywords that are available for use under a command or a subcommand. This heading will appear only on pages that are describing a command or a subcommand.

Related keywords

Lists any command language keywords that are prerequisite to, affected by, or conditionally affected by the use of the keyword described on the page. This heading will appear only on pages that are describing a keyword. If the keyword has no related keywords, this heading is omitted.

Syntax

Presents example command language syntax for the command language element and its related elements. Certain required punctuation marks are used in the syntax examples:

- Where used, commas (,) and semicolons (;) are required elements of the command syntax.
Where used, parentheses ( ) or sets of parentheses ( (  ) ) are required elements of the keyword syntax only if multiple parameters are specified for the keyword.

Where used, braces { } are required elements of the command syntax.

Other punctuation marks are used in the syntax examples as informational elements only. These punctuation marks help to simplify the examples:

- Brackets ([ ]) are used to indicate that optional additional command language elements can be coded. Brackets are not required syntax elements.

- Ellipses (...) indicate the position within the syntax example where optional additional keyword parameters or additional command language elements can be coded. Ellipses are not required syntax elements.

- A slash character (/) is used to separate explicit parameter options (where only one of the parameters should be specified). Slash characters are not required syntax elements.

Where clarification is needed, appendices are provided at the back of this book that provide syntax diagrams for each Fast Path/EP command language element.

**Parameters**

Defines numbers of placement digits or literal character strings. Parameter values can include character string, integer, and hexadecimal notation.

**Default**

If a keyword has a default parameter, the default is listed next to this heading. A default parameter is a predefined parameter that is supplied by the system automatically when the keyword is not coded on the command or subcommand.

**Alias**

Substitutes an alternate word, usually shortened or initialized, for the full word for a command, subcommand, or keyword.

**ACCESS**

**Purpose**

When restructuring a DEDB online, use the ACCESS keyword to specify the access mode for the area data sets that are processed. You can also use ACCESS to specify the IMS subsystem (imsid) or IMS group (imsgroup) that is used for dynamic allocation of the IMSRESLB, IMSACB, and RECON data sets.
When analyzing an area, use the ACCESS keyword to specify the Database Recovery Control (DBRC) sharing level for the area.

**NOTE**

If you are running in an IFP region, or analyzing an image copy, the ACCESS keyword is not supported.

**Use**

ACCESS is an optional keyword for the following commands:

- ANALYZE
- GLOBAL

ACCESS is a required keyword when used with the following commands:

- BACKOUT
- PREPARE
- RESTART
- RESTRUCTURE
- SHADOW_INIT
- STATUS

**NOTE**

The **imsid** or **imsgroup** is required on the BACKOUT, PREPARE, RESTART, RESTRUCTURE, SHADOW_INIT and STATUS commands.

**Related keyword**

DBRC

**Syntax**

```
ACCESS=parameter
ACCESS=(parameter, imsid | imsgroup)
```

**Parameters**

Specify one of the following parameter values:
If you use the ACCESS keyword with the ANALYZE or SHADOW_INIT command, the default is ACCESS=OFFLINE. When used with the PREPARE, RESTRUCTURE, or RESTART command, the default is ACCESS=ONLINE.

**Value** | **Description**
--- | ---
ONLINE | For DEDB online restructure processing, the area data sets are processed by an IFP region that is associated with the specified **imsid** or **imsgroup**.
OFFLINE | Share level is “read-with-integrity” (RD). Other programs (including IMS) cannot update the area simultaneously by other programs during analysis. The area must be offline to all IMS control regions.
CONCURRENT | Share level is “read-without-integrity” (RO). The area can be updated simultaneously by other programs during analysis. DBRC must be active.

**NOTE**
When you specify ACCESS=CONCURRENT, Fast Path Online Analyzer/EP might report errors for UOWs being updated in-flight. ACCESS=CONCURRENT requires a license for Fast Path Online Analyzer/EP.

**Default**

If you use the ACCESS keyword with the ANALYZE or SHADOW_INIT command, the default is ACCESS=OFFLINE. When used with the PREPARE, RESTRUCTURE, or RESTART command, the default is ACCESS=ONLINE.

**Alias**

None

**ACTIVITY_FILECTL**

**Purpose**

Use the ACTIVITY_FILECTL subcommand to control the allocation of the Restructure Activity data set. The data set is a sequential file that contains a log of actions performed by the online restructure process. Fast Path Online Restructure/EP uses the file as output from the Prepare, Shadow Initialization, Restructure, and Restart functions. In addition, the file is used as input into the Restart, Backout, and Status functions.

**Use**

ACTIVITY_FILECTL is an optional subcommand for the PREPARE command.
**NOTE**

If you do not specify the ACTIVITY_FILECTL subcommand, Fast Path OnlineRestructure/EP automatically allocates the Restructure Activity data set by using the Restructure Plan data set as a model. It uses default parameters (DSNAME, DISP, UNIT, and SPACE) and the data set name you specified for the PLAN_FILECTL by appending “ACT” to the end of the data set name. For example, if the name ends with PLAN, Fast Path replaces it with “.ACT”.

### Available keywords

- AVGREC
- DATACLAS
- DDNAME (required if the DSNAME keyword is not specified)
- DISP
- DSNAME (required if the DDNAME keyword is not specified)
- EXPDT
- LIKE
- MGMTCLAS
- RETPD
- SPACE
- STORCLAS
- UNIT
- VOLCNT
- VOLSER

### Syntax

```
command [keyword=parameter[,...]]
ACTIVITY_FILECTL [keyword=parameter[,...]]
```

### Parameters

None

### Default

None

### Alias

None
ACTUATE

Purpose

Use the ACTUATE keyword to request optional processing related to allocation of the VSAM cluster that is used for the command’s output.

Use

ACTUATE is an optional keyword for the ALLOCATE subcommand.

Syntax

ACTUATE=(parameter,[DSN=parameter],[EROPT=parameter])

Parameters

Specify one of the following values for the first parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELETE</td>
<td>The current VSAM cluster is deleted before the new VSAM cluster is allocated.</td>
</tr>
<tr>
<td>REUSE</td>
<td>The current VSAM cluster is allocated and overwritten.</td>
</tr>
<tr>
<td>RENAME</td>
<td>The current VSAM cluster is renamed before the new VSAM cluster is allocated.</td>
</tr>
<tr>
<td>IDCAMS</td>
<td>A sequential or partitioned data set is passed to Access Method Services as input for allocation of the new VSAM cluster. Due to the design of this feature, each area should have a separate IDCAMS definition.</td>
</tr>
</tbody>
</table>

If you specified RENAME or IDCAMS as the first parameter on the ACTUATE keyword, you must specify a dataset name mask for the DSN parameter. You can use the same substitution variables as defined for the DSNAME keyword on page 133.

The optional EROPT parameter can be specified to indicate whether or not to continue ALLOCATE subcommand processing in the event that an error occurs during processing of the output area allocation criteria that is requested by the ACTUATE keyword. Specify one of the following values for the EROPT parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABORT</td>
<td>Abort the allocate function if an ACTUATE processing error is encountered.</td>
</tr>
<tr>
<td>IGNORE</td>
<td>Continue processing of the ALLOCATE function, even if an ACTUATE processing error is encountered.</td>
</tr>
</tbody>
</table>
Default

The default value for the optional EROPT parameter is EROPT=ABORT. Defaults do not apply to any other parameters.

Alias

None

ADD

Purpose

Use the ADD subcommand to add a new object into the repository catalog data set that is being processed by the PFPEPR00 utility.

Use

ADD is an optional subcommand for the PROCESS_EPR command.

Available keywords

- AREA_KEY
- DATACLAS
- DETAIL_LEVEL
- DBD_KEY
- DSNNAME
- EXPDT
- GROUP_KEY
- MGMTCLAS
- RETPD
- STORCLAS
- UNIT
- VOLSER

Syntax

```
PROCESS_EPR REPOSITORY_DSNNAME= parameter
ADD object-type, [keyword=parameter[...]]
```

Parameters

Specify one of the following values for the object-type parameter:
ADDN

Purpose

Use the ADDN keyword to specify the DDname for the area data set.

Use

ADDN is an optional keyword for the REGISTER subcommand. If you specify the ADDN keyword, the OAREA keyword must specify only one area.

Related keyword

None

Syntax

ADDN=parameter
Parameters

Specify a 1-character to 8-character DDname for the area data set.

Default

The default value is the area name.

Alias

None

**ADSN**

Purpose

Use the ADSN keyword to specify an area data set name.

Use

ADSN is an optional keyword for the REGISTER subcommand.

Related keyword

DSNAME

Syntax

ADSN='parameter'

Parameters

Specify a 1-character to 64-character data set name. Enclose the name in single or double quotation marks. Use the following variables, as necessary, to create a mask for the name:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;ADDN</td>
<td>area ddname</td>
</tr>
<tr>
<td>&amp;AREA</td>
<td>area name</td>
</tr>
<tr>
<td>&amp;AREA#</td>
<td>specify the area number as 3 digits when the area number is less than or equal to 999; specify the area number as 5 digits when the area number is greater than 999</td>
</tr>
<tr>
<td>&amp;AREA4#</td>
<td>4-digit area number</td>
</tr>
</tbody>
</table>
The following table provides examples of the resulting data set name that is
dynamically allocated based on the specified area variable, and the area name or
number of digits in the area number:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;AREA5#</td>
<td>specify the area number as 5 digits</td>
</tr>
<tr>
<td>&amp;DATE</td>
<td>current date (“Dyyddd”) in its default form (local or UTC)</td>
</tr>
<tr>
<td>&amp;DBD</td>
<td>dbdname</td>
</tr>
</tbody>
</table>
| &IMSID        | IMS subsystem ID (actual IMS subsystem ID if online; subsystem ID from
|               | DFSCV000 if offline)                                                      |
| &INDEX        | index name                                                                 |
| &JOBN         | job name                                                                  |
| &LCLDATE      | current date (“Dyyddd”) in local form                                     |
| &LCLTIME      | current time (“Thhmnss”) in local form                                    |
| &PROCSDN      | procedure step name                                                       |
| &STEPN        | step name                                                                 |
| &TIME         | current time (“Thhmnss”) in its default form (local or UTC)              |
| &UTCDATE      | current date (“Dyyddd”) in UTC form                                       |
| &UTCTIME      | current time (“Thhmnss”) in UTC form                                      |

<table>
<thead>
<tr>
<th>Area name/number</th>
<th>Specified area name variable</th>
<th>Dynamically allocated data set name</th>
</tr>
</thead>
<tbody>
<tr>
<td>A123</td>
<td>DSNAMEN='PPFTEST.DB.&amp;AREA'</td>
<td>PFP.TEST.DB.A123</td>
</tr>
<tr>
<td>26</td>
<td>DSNAMEN='PPFTEST.DB.PF&amp;AREA#'</td>
<td>PFP.TEST.DB.PF026</td>
</tr>
<tr>
<td>26</td>
<td>DSNAMEN='PPFTEST.DB.PF&amp;AREA4#'</td>
<td>PFP.TEST.DB.PF0026</td>
</tr>
<tr>
<td>26</td>
<td>DSNAMEN='PPFTEST.DB.PF&amp;AREA5#'</td>
<td>PFP.TEST.DB.PF00026</td>
</tr>
<tr>
<td>1024</td>
<td>DSNAMEN='PPFTEST.DB.PF&amp;AREA#'</td>
<td>PFP.TEST.DB.PF1024</td>
</tr>
<tr>
<td>1024</td>
<td>DSNAMEN='PPFTEST.DB.PF&amp;AREA4#'</td>
<td>PFP.TEST.DB.PF1024</td>
</tr>
<tr>
<td>1024</td>
<td>DSNAMEN='PPFTEST.DB.PF&amp;AREA5#'</td>
<td>PFP.TEST.DB.PF1024</td>
</tr>
</tbody>
</table>

Using the &AREA#4 or &AREA5# variable instead of the &AREA# variable lets you
standardize the length of dynamically allocated data set names when using 1000 or
more areas, while still supporting area numbers 1 through 999.

**Default**

None

**Alias**

None
ALLOCATE

Purpose

Use the ALLOCATE subcommand to dynamically define the VSAM cluster that is used for the command’s output processing.

Use

ALLOCATE is an optional subcommand for the following commands and subcommand:

- CHANGE
- INITIALIZE
- RELOAD
- SHADOW_INIT

Available keywords

- ACTUATE (not valid for the SHADOW_INIT command)
- AVGREC
- CONFIGURE_AREA
- CONFIGURE_IOVF
- CONFIGURE_RAA
- CONFIGURE_SDEP
- DATACLAS
- DATACLAS2
- IDCAMS_OPTION
- MGMTCLAS
- MGMTCLAS2
- OAREA
- REPORT_DDNAME
- SPACE
- STORCLAS
- STORCLAS2
- VOLSER
- VOLSER2

Syntax

command [keyword=parameter[,...]]
ALLOCATE [keyword=parameter[,...]]
**Parameters**

None

**Default**

None

**Alias**

None

---

**ANALYZE**

**Purpose**

Use the ANALYZE command to analyze one or more areas of a DEDB using the area data set or an image copy. The ANALYZE command generates analysis data of DEDB characteristics and performance, validates DEDB physical and logical integrity, generates statistical reports, and provides information to help determine space utilization requirements.

The ANALYZE command acts only in an information-gathering and analysis role and does not perform update functions to a DEDB.

**Use**

ANALYZE is an optional command for the PFPSYSIN DD statement. You must have a license for Fast Path Online Analyzer/EP or Fast Path Analyzer/EP to use this command.
Available keywords
- ACCESS (ignored in online mode)
- DBD (required for offline mode)
- HISTORY_DDNAME
- ICACHE (ignored in online mode)
- INPUT_DSN_MASK (ignored in online mode)
- LARGEST_DATABASE_RECORDS
- MESSAGE_SUPPRESSION
- MODEL_DDNAME
- ORPHANED_SDEP_MSG
- POINTER_VALIDATION
- RAP_VALIDATION
- SDEP_VALIDATION

Syntax
ANALYZE [keyword=parameter[,...]]
   [subcommand [keyword=parameter[,...]]]

Parameters
None

Default
None

Alias
None

AREA_KEY

Purpose
Use the AREA_KEY keyword to specify an area identifier. An area identifier is used as part of the record key for the repository catalog data set. It is used when adding an area level allocation rule.
Use

AREA_KEY is a required keyword for the ADD AREA_ALLOCATION subcommand set.

Related keywords

- DBD_KEY
- GROUP_KEY

Syntax

AREA_KEY=parameter

Parameters

Specify a 1-character to 8-character area name.

Default

None

Alias

None

AVGREC

Purpose

Use the AVGREC keyword to identify the space requirements of the output data set created by dynamic allocation. This keyword is meaningful only when the allocation units (specified using the SPACE keyword) is given as a number of bytes.

Use

AVGREC is an optional keyword for the following subcommands:

- ACTIVITY_FILECTL
- ALLOCATE
- DISCARD_FILECTL
- IC
- OFILECTL
- PLAN_FILECTL
### Related keyword

SPACE

### Syntax

AVGREC=parameter

### Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Specifies that the allocation units specified by the units subparameter of the SPACE keyword represents an average block size.</td>
</tr>
</tbody>
</table>
| U     | Specifies the following:  
  - The allocation units specified by the units subparameter of the SPACE keyword represents an average record size.  
  - The multiplier of the primary and secondary quantities is 1. |
| K     | Specifies the following:  
  - The allocation units specified by the units subparameter of the SPACE keyword represents an average record size.  
  - The multiplier of the primary and secondary quantities is 1024. |
| M     | Specifies the following:  
  - The allocation units specified by the units subparameter of the SPACE keyword represents an average record size.  
  - The multiplier of the primary and secondary quantities is 1048576. |

### Default

AVGREC=B

### Alias

None
**BACKOUT**

**Purpose**

If an error occurs during the online restructure process, you can use the BACKOUT command to restore your database to its original state before the restructure. Use the PLAN_FILECTL subcommand with the BACKOUT command to specify the name of the Restructure Plan data set and Restructure Activity data set that was used when the restructure process failed.

**Use**

BACKOUT is an optional command for the PFPSYSIN DD statement when you execute the PFCMAIN program. You must have a license for Fast Path Online Restructure/EP to use this command.

**Available keywords**

- ACCESS (required)
- DBD (required)

**Syntax**

BACKOUT [keyword=parameter[,...]]  
subcommand[keyword=parameter[,....]]

**Parameters**

None

**Default**

None

**Alias**

None

---

**BREAK**

**Purpose**

Use the BREAK keyword to specify the event that triggers the output record to be written to the extract file.
Use

BREAK is a required keyword for the USER_RECORD subcommand.

Syntax

\texttt{BREAK=(level[,when])}

Parameters

Specify one of the following values for the \textit{level} parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATABASE</td>
<td>The trigger event occurs at the database processing level.</td>
</tr>
<tr>
<td>AREA</td>
<td>The trigger event occurs at the area processing level.</td>
</tr>
</tbody>
</table>

Specify one of the following values for the \textit{when} parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEFORE (default)</td>
<td>The user record is triggered when processing begins for the specified \textit{level}.</td>
</tr>
<tr>
<td>AFTER</td>
<td>The user record is triggered after processing is complete for the specified \textit{level}.</td>
</tr>
</tbody>
</table>

Default

There is no default value for the \textit{level} parameter. The default value for the \textit{when} parameter is BEFORE: \texttt{BREAK=level,BEFORE}).

Alias

None

BUILD

Purpose

Use the BUILD command to create one or more secondary index databases from an associated primary DEDB. You can use the BUILD command with PFX and IBM native indexes if you provide the appropriate password. (For more information about the product password, see the \textit{Fast Path Indexer/EP User Guide}.)
Use

BUILD is an optional command for the PFPSYSIN DD statement. You must have a license for Fast Path Indexer/EP to use this command.

Available keywords

- DBD (required)
- IAREA
- ICACHE
- INDEX_THREADS
- INPUT_DSN_MASK
- INPUT_THREADS
- SORT
- SORT_OPTION

The following keywords are invalid if an XSCAN data set that is created by the XSCAN utility is used for input into the build process instead of using the default scan and sort functions:

- IAREA
- ICACHE
- INPUT_DSN_MASK

Syntax

BUILD [keyword=parameter[,...]]
subcommand [keyword=parameter[,...]]

Parameters

None

Default

None

Alias

None
BYPASS_RECORD

Purpose

Use the BYPASS_RECORD keyword to specify the action to be taken by the command process (terminate or continue) if root segments are randomized to an area of the DEDB that is not being processed by the command set.

Use

BYPASS_RECORD is an optional keyword for the following commands:

- CHANGE
- RELOAD
- UNLOAD

Syntax

BYPASS_RECORD=parameter

Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>The first root segment that is randomized to an area that is not being processed will terminate processing. An error message is generated that contains the key of the bypassed root segment.</td>
</tr>
<tr>
<td>YES</td>
<td>If root segments randomize to an area that is not being processed, processing continues. An informational message is written to the PFPPRINT DD that includes a count of the total number of bypassed records.</td>
</tr>
</tbody>
</table>

Default

BYPASS_RECORD=YES

Alias

None
**CASE**

**Purpose**

Use the CASE keyword to specify whether output messages and reports are generated with mixed-case (uppercase and lowercase) letters or with uppercase letters only.

**Use**

CASE is an optional keyword for the OPTIONS command.

**Syntax**

\[ CASE=\text{parameter} \]

**Parameters**

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIXED</td>
<td>Use uppercase and lowercase letters.</td>
</tr>
<tr>
<td>UPPER</td>
<td>Use uppercase letters only.</td>
</tr>
</tbody>
</table>

**Default**

CASE=MIXED

**Alias**

None

**CHANGE**

**Purpose**

Use the CHANGE command to restructure a DEDB. When restructuring a DEDB, you must have two DD statements in the JCL: the OLDACB DD statement and the IMSACB DD statement. The OLDACB DD statement specifies the library with the input ACB. The IMSACB DD statement specifies the library with the output ACB. The CHANGE command can use image copy data sets or DEDB data sets as input.
CHANGE

**NOTE**

Only one CHANGE command can be executed per job step, and it must be the *only* command specified in the PFPSYSIN input.

**Use**

CHANGE is an optional command for the PFPSYSIN DD statement. You must have a license for Fast Path Online Reorg/EP or Fast Path Reorg/EP to use this command.

**Available keywords**

- BYPASS_RECORD
- DBD (required)
- ERROR_THRESHOLD
- EXPAND
- IAREA
- ICACHE
- INDEX_THREADS
- INPUT_DSN_MASK
- INPUT_THREADS
- IOVF_LOAD_HWM
- LARGEST_DATABASE_RECORDS
- OAREA
- OCACHE
- OUTPUT_DSN_MASK
- OUTPUT_THREADS
- POINTER_VALIDATION
- RAP_VALIDATION
- SDEP_PROCESS
- SDEP_VALIDATION
- SUBSET_POINTERS

**Syntax**

```
CHANGE [keyword=parameter[,...]]
[ [subcommand[keyword=parameter[,...]]]
```

**Parameters**

None
CHECKPOINT

Purpose

Use the CHECKPOINT keyword to specify the number of segment reads that will occur before Fast Path Indexer/EP performs a symbolic checkpoint call.

**NOTE**

The CHECKPOINT keyword functions in BMP mode only. If you specify a value for CHECKPOINT with an offline command, the keyword will be ignored.

Use

CHECKPOINT is an optional keyword for the following commands:

- VERIFY
- RESYNC

Syntax

```
CHECKPOINT=parameter
```

Parameters

Specify a numeric value between 1 and 16777215.

Default

```
CHECKPOINT=10000
```

Alias

None
COMPRESS

Purpose

Use the COMPRESS keyword to compress data segments before the segments are inserted to a DEDB.

You must be using a compression product where compression is controlled by the DBD. If you are using a compression product where compression is controlled by something other than the DBD, this keyword has no effect.

Use

COMPRESS is an optional keyword for the following commands:

- RELOAD
- REORGANIZE
- UNLOAD

**NOTE**
The COMPRESS keyword can be used with the REORGANIZE command *only* if you are running BMC Software’s DATA PACKER®/IMS product. Fast Path Reorg/EP will use the keyword value SELECT_UOW=ALL to ensure that all segments are reorganized and recompressed.

Related keywords

- EXPAND
- SELECT_UOW

Syntax

COMPRESS=parameter
COMPRESS=(parameter1, parameter2, ..., parametern)

Parameters

Specify one or more of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO (default)</td>
<td>Do not invoke compression.</td>
</tr>
<tr>
<td>YES</td>
<td>Compress the data before the segment is processed.</td>
</tr>
<tr>
<td>1-character to 8-character segment name</td>
<td>Compress only the specified segment or segments. This value can be used more than once to specify specific segments using the (parameter1, parameter2, ..., parametern) syntax shown above.</td>
</tr>
</tbody>
</table>
**NOTE**

If you request compression by specifying either COMPRESS=YES or COMPRESS=segment name(s), the compression routine must be present in the appropriate library. Refer to the DD statements IMSRESLB, OLDRESLB, and STEPLIB.

**Default**

COMPRESS=NO

**Alias**

CMP

## COMPRESSION

**Purpose**

Use the COMPRESSION keyword to specify the type of compression to use on the image copy data set that is being created with the IC subcommand.

**Use**

COMPRESSION is an optional keyword for the IC subcommand.

**Syntax**

COMPRESSION=\textit{parameter}

**Parameters**

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONE</td>
<td>No compression is used.</td>
</tr>
<tr>
<td>FSE</td>
<td>Free space element elimination. This method does not write free space elements or free blocks to the output. No other compression is performed.</td>
</tr>
</tbody>
</table>
CONFIGURE_AREA

Purpose

Use the CONFIGURE_AREA keyword to specify the allocation characteristics of the entire area data set that is used for output processing.

Use

CONFIGURE_AREA is an optional keyword for the ALLOCATE subcommand.

Related keyword

SPACE

Syntax

CONFIGURE_AREA=(/[VOLCNT=parameter],[parameter[, ...]])

Parameters

Specify the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCC</td>
<td>Common character compression. This method 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.</td>
</tr>
<tr>
<td>DPE</td>
<td>Full character compression. This method compresses all character types using a subset of the BMC Software’s 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.</td>
</tr>
</tbody>
</table>

Default

COMPRESSION=NONE

Alias

None
### CONFIGURE_IOVF

**Purpose**

Use the CONFIGURE_IOVF keyword to specify the allocation characteristics of the independent overflow portion of the area data set that is used for the command’s output processing.

**Use**

CONFIGURE_IOVF is an optional keyword for the ALLOCATE subcommand.

**Related keyword**

SPACE

**Syntax**

```
CONFIGURE_IOVF=([VOLCNT=parameter],[parameter[, ...]])
```

**Parameters**

Specify the following parameter values:
CONFIGURE_RAA

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLCNT=</td>
<td>Specify a value from 0 to 32 to indicate the maximum number of volumes that</td>
</tr>
<tr>
<td>parameter</td>
<td>will be required for the IOVF of the output area. This parameter is optional.</td>
</tr>
<tr>
<td>parameter</td>
<td>Specify one or more values from 1 to 32767, with each value separated by a</td>
</tr>
<tr>
<td></td>
<td>comma. Each value represents the amount of track or cylinder space to be</td>
</tr>
<tr>
<td></td>
<td>allocated to each extent (as requested by the SPACE keyword parameter or as</td>
</tr>
<tr>
<td></td>
<td>determined by IDCAMS).</td>
</tr>
</tbody>
</table>

Default

CONFIGURE_IOVF=(VOLCNT=0)

Alias

None

CONFIGURE_RAA

Purpose

Use the CONFIGURE_RAA keyword to specify the allocation characteristics of the root-addressable portion of the area data set that is used for the command’s output processing.

Use

CONFIGURE_RAA is an optional keyword for the ALLOCATE subcommand.

Related keyword

SPACE

Syntax

CONFIGURE_RAA=([VOLCNT=parameter],[parameter[, ...]])

Parameters

Specify the following parameter values:
 CONFIGURE_SDEP

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLCNT=</td>
<td>Specify a value from 0 to 32 to indicate the maximum number of volumes that will be required for the RAA of the output area. This parameter is optional.</td>
</tr>
<tr>
<td>parameter</td>
<td>Specify one or more values from 1 to 32767, with each value separated by a comma. Each value represents the amount of track or cylinder space to be allocated to each extent (as requested by the SPACE keyword parameter or as determined by IDCAMS).</td>
</tr>
</tbody>
</table>

Default

CONFIGURE_RAA=(VOLCNT=0)

Alias

None

CONFIGURE_SDEP

Purpose

Use the CONFIGURE_SDEP keyword to specify the allocation characteristics of the sequential dependent portion of the area data set that is used for the command’s output processing.

Use

CONFIGURE_SDEP is an optional keyword for the ALLOCATE subcommand.

Related keyword

SPACE

Syntax

CONFIGURE_SDEP=( [VOLCNT=parameter], [parameter[,] ...] )
Parameters

Specify the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLCNT=</td>
<td>Specify a value from 0 to 32 to indicate the maximum number of volumes that will be required for the SDEP of the output area. This parameter is optional.</td>
</tr>
<tr>
<td>parameter</td>
<td>Specify one or more values from 1 to 32767. Each value represents the amount of track or cylinder space to be allocated to each extent (as requested by the SPACE keyword parameter or as determined by IDCAMS).</td>
</tr>
</tbody>
</table>

Default

CONFIGURE_SDEP=(VOLCNT=0)

Alias

None

**CORRECTIONS_FILECTL**

Purpose

Use the CORRECTIONS_FILECTL subcommand (in either online or offline mode) to detect and remove invalid pointers. When pointer errors are detected, this subcommand can be used to generate an output file that contains pointer correction control cards that can be used to remove the invalid pointers.

The output file can be created within your JCL (refer to the DDNAME keyword), or by dynamic allocation (refer to the DSNNAME keyword).

Use

CORRECTIONS_FILECTL is an optional subcommand for the ANALYZE command.
NOTE

To use the CORRECTIONS_FILECTL subcommand successfully, you must specify either POINTER_VALIDATION=FULL or SDEP_VALIDATION=FULL with the ANALYZE command:

- Specifying POINTER_VALIDATION=FULL generates error corrections control cards for direct pointers.
- Specifying SDEP_VALIDATION=FULL generates error corrections control cards for sequential pointers.

Available keywords

- DDNAME (required if the DSNAME keyword is not specified)
- DSNAME (required if the DDNAME keyword is not specified)

Syntax

```
command [keyword=parameter[,...]]
CORRECTIONS_FILECTL [keyword=parameter[,...]]
```

Parameters

None

Default

None

Alias

None

DATACLAS

Purpose

Use the DATACLAS keyword to specify the SMS data class of an output data set created by dynamic allocation.
Use

DATACLAS is an optional keyword for the following subcommands:

- ACTIVITY_FILECTL
- ADD
- ALLOCATE
- DISCARD_FILECTL
- IC
- MODIFY
- OFILECTL
- PLAN_FILECTL

Related keyword

DSNAME

Syntax

DATACLAS=parameter

Parameters

Specify a 1-character to 8-character data class name.

Default

None

Alias

None

DATAKLAS2

Purpose

Use the DATAKLAS2 keyword to specify the SMS data class of a secondary shadow output data set created by dynamic allocation.

Use

DATAKLAS2 is an optional keyword for the ALLOCATE subcommand.
## DATE_TIME_FORMAT

**Purpose**

Use the DATE_TIME_FORMAT keyword to specify the default format for date and time data.

**Use**

DATE_TIME_FORMAT is an optional keyword for the OPTIONS command.

**Syntax**

```plaintext
DATE_TIME_FORMAT=parameter
```

**Parameters**

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCAL</td>
<td>Date and time values display in local form.</td>
</tr>
<tr>
<td>UTC</td>
<td>Date and time values display in Coordinated Universal Time (UTC) form.</td>
</tr>
</tbody>
</table>
**Default**

DATE_TIME_FORMAT=LOCAL

**Alias**

None

---

**DBD**

**Purpose**

Use the DBD keyword to identify the database definition (DBD) containing the areas to be processed. The DBD name must correspond to a member name in the IMSACB library.

**Use**

DBD is a *required* keyword for the following commands when running in offline mode:

- ANALYZE
- BUILD
- CHANGE
- DMAC_PRINT
- EXTEND
- EXTRACT
- INITIALIZE
- PFPSORT
- PROCESS_AREA
- RELOAD
- REORGANIZE
- UNLOAD
- VERIFY
- XSCAN
DBD is an *optional* keyword for the following commands when running in online mode. If the DBD keyword is specified, the DBD name must match the value specified in the PARM keyword on the JCL EXEC statement.

- ANALYZE
- DMAC_PRINT
- EXTEND
- EXTRACT
- IMAGECOPY
- PROCESS_AREA
- REORGANIZE

DBD is a *required* keyword for the following commands when running in BMP mode. The DBD name must be included in the PSB specified in the PARM keyword on the JCL EXEC statement.

- RESYNC
- VERIFY

DBD is a *required* keyword for the XSCAN command when running in IFP mode.

DBD is a *required* keyword for the following commands when running Fast Path Online Restructure/EP:

- BACKOUT
- PREPARE
- RESTART
- RESTRUCTURE
- SHADOW_INIT
- STATUS

**Related keywords**

- IAREA
- OAREA
- REQUIRE_AREA

**Syntax**

```
DBD=parameter
```

**Parameters**

Specify a 1-character to 8-character database name.
DBD_KEY

**Purpose**

Use the DBD_KEY keyword to specify a DBD identifier. A DBD identifier is used as part of the record key for the repository catalog data set. It is used when adding a DBD level and an area level allocation rule.

**Use**

DBD_KEY is a required keyword for the following subcommand sets:

- ADD DBD_ALLOCATION
- ADD AREA_ALLOCATION

**Related keywords**

- AREA_KEY
- DBD_KEY

**Syntax**

```
DBD_KEY=parameter
```

**Parameters**

Specify a 1-character to 8-character database name.

**Default**

None

**Alias**

None
DBRC

Purpose

Use the DBRC keyword in offline mode to request or bypass database recovery control (DBRC) facility processing of the database. Processing includes DBRC signon and signoff database authorization, and event notification. This keyword has no effect when running in online mode.

Use

DBRC is an optional keyword for the GLOBAL command.

Related keyword

ACCESS

Syntax

DBRC=parameter
DBRC=(YES[,gsgname])

Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Bypass DBRC processing of the database.</td>
</tr>
<tr>
<td>YES</td>
<td>Request DBRC processing of the database.</td>
</tr>
<tr>
<td>gsgname</td>
<td>If running in a Remote Site Recovery (RSR) environment, specify Global Service Group (GSG) name.</td>
</tr>
<tr>
<td>FORCEOFF</td>
<td>Bypass DBRC processing of the database if your system has been defined to force the use of DBRC.</td>
</tr>
</tbody>
</table>

**NOTE**

If DBRC=FORCE is set for IMS, specifying DBRC=NO does not bypass DBRC. You must specify DBRC=FORCEOFF instead.

DBRC=YES is required with ACCESS=CONCURRENT.
DDNAME

Default

In offline mode, DBRC runs under IMS control and is specified for IMS in the RESLIB. The Fast Path/EP default for the DBRC keyword is the parameter specified for IMS. In online mode, DBRC is always active.

Alias

None

DDNAME

Purpose

Use the DDNAME keyword to specify the ddname(s) for the data set(s) to be processed by the subcommand. If the JCL does not contain a DD statement with the referenced ddname(s), an error message is issued. One copy of the output data set is written to each ddname specified.

NOTE

The subcommand should contain either the DDNAME or DSNAME keyword. Do not specify both keywords; they are mutually exclusive.

Use

DDNAME is an optional keyword for the following subcommands:

- ACTIVITY_FILECTL
- CORRECTIONS_FILECTL
- DISCARD_FILECTL
- IC
- IX
- OFILECTL
- PLAN_FILECTL

Syntax

```
DDNAME=parameter
DDNAME=(parameter[, parameter[, ...]])
```
Parameters

Specify a 1-character to 8-character data set name.

Default

None

Alias

None

DEFLTJCL

Purpose

Use the DEFLTJCL keyword to specify an implicit skeletal JCL default member for the DBDS. The GENJCL.IC and GENJCL.RECOV DBRC commands use the member to resolve keywords.

Use

DEFLTJCL is an optional keyword for the REGISTER subcommand.

Related keywords

None

Syntax

DEFLTJCL=parameter

Parameters

Specify a 1-character to 8-character member name.

Default

None

Alias

None
DELETE

Purpose

Use the DELETE subcommand to delete an existing object from the repository catalog data set being processed by the PFPEPR00 utility.

Use

DELETE is an optional subcommand for the PROCESS_EPR command.

Related keywords

- REPOSITORY_RETENTION_PERIOD
- SELECT_AREA
- SELECT_DATE
- SELECT_DBD
- SELECT_GROUP

Syntax

PROCESS_EPR [keyword=parameter[,...]]
   DELETE object-type. [keyword=parameter[,...]]

Parameters

Specify one of the following values for the object-type parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOBAL_ALLOCATION</td>
<td>Delete the global level allocation rule.</td>
</tr>
<tr>
<td>GROUP_ALLOCATION</td>
<td>Delete all group level allocation rules that match the criteria specified using the SELECT_GROUP keyword.</td>
</tr>
<tr>
<td>DBD_ALLOCATION</td>
<td>Delete all DBD level allocation rules that match the criteria specified using the SELECT_GROUP and SELECT_DBD keywords.</td>
</tr>
<tr>
<td>AREA_ALLOCATION</td>
<td>Delete all area level allocation rules that match the criteria specified using the SELECT_GROUP, SELECT_DBD and SELECT_AREA keywords.</td>
</tr>
<tr>
<td>ALLOCATION</td>
<td>Delete all allocation rules (regardless of type) that match the selection criteria specified using the SELECT_GROUP, SELECT_DBD and SELECT_AREA keywords.</td>
</tr>
<tr>
<td>STATISTICS</td>
<td>Delete all statistics catalog entries that match the selection criteria specified using the SELECT_GROUP, SELECT_DBD, SELECT_AREA, SELECT_DATE and REPOSITORY_RETENTION_PERIOD keywords.</td>
</tr>
</tbody>
</table>
DESC

Purpose

Use the DESC keyword to specify the write-to-operator (WTO) message descriptor codes for messages sent to the system operator.

Use

DESC is an optional keyword for the OPTIONS command.

Related keyword

ROUTCDE

Syntax

DESC=parameter
DESC=(parameter1, parameter2[, ...])

Parameters

Specify a value from 1 to 13.

NOTE

The DESC keyword has the same format and follows the same coding practices and rules as the DESC keyword on the WTO macro. The values 1 to 6, 11, and 12 are mutually exclusive.

Default

DESC=6

Alias

None
**DETAIL**

**Purpose**

Use the DETAIL keyword to specify items that require additional details to be generated on the Restructure Activity Report.

**Use**

DETAIL is an optional keyword for the following commands:

- RESTART
- RESTRUCTURE
- STATUS

**Syntax**

DETAIL=parameter

**Parameters**

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONE</td>
<td>No additional details are generated on the Restructure Activity Report.</td>
</tr>
<tr>
<td>ALL</td>
<td>Additional details about DBRC registration and history, component names, and IDCAMS definitions are generated on the Restructure Activity Report</td>
</tr>
<tr>
<td>DBRC</td>
<td>Additional details about DBRC registration and history are generated on the Restructure Activity Report.</td>
</tr>
<tr>
<td>COMPONENT_NAME</td>
<td>Additional details about component names for data sets are generated on the Restructure Activity Report.</td>
</tr>
<tr>
<td>IDCAMS</td>
<td>Additional details about IDCAMS definitions are generated on the Restructure Activity Report.</td>
</tr>
</tbody>
</table>

**Default**

DETAIL=None

**Alias**

None
**DETAIL_LEVEL**

**Purpose**

Use the optional DETAIL_LEVEL keyword to specify the level of detail to write into the repository statistics data set. The statistics written to the repository statistics data set can provide details for each individual UOW or can be summarized into area details. The data set that is used to save summarized area details is significantly smaller than the data set that is used to save individual UOW details.

When statistics are retrieved from the repository by using the RETRIEVE command (see “RETRIEVE” on page 279), the level of detail (UOW or area) written affects the reports that can be generated. When UOW details are written, reports can be generated for a a range of UOWs. When area details are written, only area reports can be generated. As a result, the STARTUOW and STOPUOW keywords cannot be used to generate reports for individual UOWs when area details are written to the repository statistics data set.

**Use**

DETAIL_LEVEL is an optional keyword for the following subcommand set:

- ADD ALLOCATION
- ADD AREA_ALLOCATION
- ADD DBD_ALLOCATION
- ADD GLOBAL_ALLOCATION
- ADD GROUP_ALLOCATION
- MODIFY ALLOCATION
- MODIFY AREA_ALLOCATION
- MODIFY DBD_ALLOCATION
- MODIFY GLOBAL_ALLOCATION
- MODIFY GROUP_ALLOCATION

**Syntax**

```
DETAIL_LEVEL=parameter
```

**Parameters**

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOW</td>
<td>store repository statistics with all UOW details</td>
</tr>
<tr>
<td>AREA</td>
<td>store repository statistics with area details only</td>
</tr>
</tbody>
</table>
**DISCARD_FILECTL**

**Default**

DETAIL_LEVEL=UOW

**Alias**

None

**Purpose**

Use the DISCARD_FILECTL subcommand to handle discarded data (data that could not be reloaded). This subcommand is used to request that a discard file be created during command processing.

As long as the number of exceptions that are encountered in an input area does not exceed the value specified by using the EXCEPTION_LIMIT keyword, the command process will discard the segment containing the error, along with all of its dependent segments. The discarded segments are written to an output file only if the DISCARD_FILECTL subcommand and the EXCEPTION_LIMIT keyword are specified. For information about the record layout of the discard file, see Appendix D, “Discard file record layout.”

---

**NOTE**

All discarded segments are written to one output file, regardless of how many input areas are processed.

The discarded output file can be created within your JCL (refer to the DDNAME keyword), or by dynamic allocation (refer to the DSNAME keyword).

**Use**

DISCARD_FILECTL is an optional subcommand for the RELOAD command.

---

**NOTE**

Only one DISCARD_FILECTL subcommand is allowed in the RELOAD command set.
Available keywords

- AVGREC
- DATACLAS
- DDNAME (required if DSNAME keyword is not specified)
- DISP
- DSNAME (required if DDNAME keyword is not specified)
- EXPDT
- LIKE
- MGMTCLAS
- RETPD
- SPACE
- STORCLAS
- UNIT
- VOLCNT
- VOLSER

Syntax

```
command [keyword=parameter[,...]]
DISCARD_FILECTL [keyword=parameter[,...]]
```

Parameters

- None

Default

- None

Alias

- None

**DISP**

**Purpose**

Use the DISP keyword to specify the status and disposition of the data set accessed by dynamic allocation.
Use

DISP is an optional keyword for the following subcommands:

- ACTIVITY_FILECTL
- DISCARD_FILECTL
- IC
- IX
- OFILECTL
- PLAN_FILECTL

Related keyword

DSNAME

Syntax

```plaintext
DISP=(status[,normal[,conditional]])
```

Parameters

Specify one of the following values for the `status` parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEW</td>
<td>Create a new data set.</td>
</tr>
<tr>
<td>OLD</td>
<td>Use an existing data set exclusively.</td>
</tr>
<tr>
<td>USE</td>
<td>Use conditional allocation. If the data set does not exist, it is created (as if NEW had been specified). If the data set already exists, it is re-allocated (as if OLD had been specified). The normal and conditional parameters are also changed from CATLG (if specified) to KEEP.</td>
</tr>
</tbody>
</table>

Specify one of the following values for the `normal` parameter to request the action to occur if the data set is created successfully:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELETE</td>
<td>Release the space allocated for the data set.</td>
</tr>
<tr>
<td>KEEP</td>
<td>Keep the data set on the volume.</td>
</tr>
<tr>
<td>CATLG</td>
<td>Place an entry pointing to the data set in the catalog.</td>
</tr>
<tr>
<td>UNCATLG</td>
<td>Remove the catalog entry to the data set, but retain the data set.</td>
</tr>
</tbody>
</table>
Specify one of the following values for the *conditional* parameter to request the action to occur if the if any errors occur while creating the data set:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELETE</td>
<td>Release the space allocated for the data set.</td>
</tr>
<tr>
<td>KEEP</td>
<td>Keep the data set on the volume.</td>
</tr>
<tr>
<td>CATLG</td>
<td>Place an entry pointing to the data set in the catalog.</td>
</tr>
<tr>
<td>UNCATLG</td>
<td>Remove the catalog entry to the data set, but retain the data set.</td>
</tr>
</tbody>
</table>

**Default**

Default settings for the DISP keyword are as follows:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>DISP=NEW</td>
</tr>
<tr>
<td>normal</td>
<td>If status is NEW, then DISP=(, DELETE). If status is OLD, then DISP=(, KEEP).</td>
</tr>
<tr>
<td>conditional</td>
<td>If status is NEW, then DISP=(, DELETE). If status is OLD, then DISP=(, KEEP).</td>
</tr>
</tbody>
</table>

**Alias**

None

**DISPLAY**

**Purpose**

Use the DISPLAY command to request information about resources specified with the OBJECT keyword or tasks identified with the ID keyword.

**Use**

DISPLAY is an optional command for the Fast Path/EP operator interface.

**Related keywords**

- ID
- OBJECT
DMAC_CLEANUP

Syntax

DISPLAY [keyword=parameter[, ...]]

Parameters

None

Default

None

Alias

None

Purpose

Use the DMAC_CLEANUP command to update the DMAC for a specified database if a restructure fails to complete successfully, and you want to run the IBM SDEP Scan and SDEP Delete utilities. The command can be used in offline mode or as an IFP.

Use

DMAC_CLEANUP is an optional command for the PFPSYSIN DD statement that is used with PFPMAIN.

Related keywords

- DBD (required)
- IAREA
- MESSAGE_SUPPRESSION

Syntax

DMAC_CLEANUP [keyword=parameter[, ...]]

Parameters

None
**DMAC_PRINT**

**Purpose**

Use the DMAC_PRINT command to print contents of the DMAC block for specified areas or all areas within a DEDB. The function can be used in either offline or online mode.

When DMAC_PRINT executes in offline mode, the input to the function can be either the area data set or an image copy. The utility prints the contents of the DMAC constructed by merging the values from the DMB (from the ACB Library) with the DMAC block from the area data set or image copy.

When DMAC_PRINT executes in online mode, the input to the function is the online area data set. The utility prints the contents of the DMAC in use by the IMS control region.

**Use**

DMAC_PRINT is an optional command for the PFPSYSIN DD statement.

**Related keywords**

- DBD (required for offline mode)
- IAREA
- INPUT_DSN_MASK (ignored in online mode)
- MESSAGE_SUPPRESSION

**Syntax**

```
DMAC_PRINT [keyword=parameter[, ...]]
```

**Parameters**

None
DOVF_FREESPACE_PERCENT

Default

None

Alias

None

DOVF_FREESPACE_PERCENT

Purpose

Use the DOVF_FREESPACE_PERCENT keyword to specify a threshold setting when analyzing or monitoring DEDB activity. The percentage of free space within DOVF blocks is computed and compared with the user-specified percentage threshold setting. If the computed value is less than the user-specified setting, a warning message is issued.

NOTE

If you do not specify the DOVF_FREESPACE_PERCENT keyword, this threshold test is not performed.

Use

DOVF_FREESPACE_PERCENT is an optional keyword for the THRESHOLD subcommand.

Syntax

DOVF_FREESPACE_PERCENT=parameter

Parameters

Specify a value from 0 to 100.

Default

None

Alias

DFP
Purpose

Use the DSNAME keyword to specify a mask for a data set name to use for dynamic allocation of the data set to be processed by the subcommand. The data set name produced by the mask (after variable substitutions are performed) must conform to standard data set naming rules. It cannot contain a reference to a partitioned data set (PDS) member name. It can contain a reference to a generation data set group (GDG), with or without a relative generation number such as “(+1)”.

If DISP=OLD or DISP=SHR is specified, the data set name must refer to an existing catalogued data set. If DISP=NEW is specified, the data set name must not refer to an existing catalogued data set.

If the data set name matches an existing GDG base name, the product will automatically append the appropriate generation information to the base name equivalent to relative generation “(+1)”, if not specified explicitly.

DISP=(NEW,CATLG) is required when a new generation data set is being created.

**NOTE**
The subcommand should contain either the DSNAME or DDNAME keyword. Do not specify both keywords; they are mutually exclusive.

Use

DSNAME is an optional keyword for the following subcommands:

- ACTIVITY_FILECTL
- CORRECTIONS_FILECTL
- DISCARD_FILECTL
- IC
- IX
- MODIFY
- OFILECTL
- PLAN_FILECTL

DSNAME is a required keyword for the ADD subcommand.
Related keywords
- DATACLAS
- DDNAME
- DISP
- EXPDT
- LIKE
- MGMTCLAS
- RETPD
- SPACE
- STORCLAS
- UNIT
- VOLENT
- VOLSER

Syntax

DSNAME='parameter'

Parameters

Specify a 1-character to 64-character data set name mask. Enclose the mask in single or double quotation marks. Use the following variables, as necessary, to create the mask:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;AREA</td>
<td>area name</td>
</tr>
<tr>
<td>&amp;AREA#</td>
<td>specify the area number as 3 digits when the area number is less than or equal to 999; specify the area number as 5 digits when the area number is greater than 999</td>
</tr>
<tr>
<td>&amp;AREA4#</td>
<td>4-digit area number</td>
</tr>
<tr>
<td>&amp;AREA5#</td>
<td>specify the area number as 5 digits</td>
</tr>
<tr>
<td>&amp;DATE</td>
<td>current date (&quot;Dyyddd&quot;) in its default form (local or UTC)</td>
</tr>
<tr>
<td>&amp;DBD</td>
<td>dbdname</td>
</tr>
<tr>
<td>&amp;IMSID</td>
<td>IMS subsystem ID (actual IMS subsystem ID if online; subsystem ID from DFSCV000 if offline)</td>
</tr>
<tr>
<td>&amp;INDEX</td>
<td>index name</td>
</tr>
<tr>
<td>&amp;JOBN</td>
<td>job name</td>
</tr>
<tr>
<td>&amp;PROCSN</td>
<td>procedure step name</td>
</tr>
<tr>
<td>&amp;LCLDATE</td>
<td>current date (&quot;Dyyddd&quot;) in local form</td>
</tr>
<tr>
<td>&amp;LCLTIME</td>
<td>current time (&quot;Thhmmss&quot;) in local form</td>
</tr>
<tr>
<td>&amp;STEPN</td>
<td>step name</td>
</tr>
<tr>
<td>&amp;TIME</td>
<td>current time (&quot;Thhmmss&quot;) in its default form (local or UTC)</td>
</tr>
</tbody>
</table>
Chapter 2 Command language

The following table provides examples of the resulting data set name that is dynamically allocated based on the specified area variable, and the area name or number of digits in the area number:

<table>
<thead>
<tr>
<th>Area name/number</th>
<th>Specified area name variable</th>
<th>Dynamically allocated data set name</th>
</tr>
</thead>
<tbody>
<tr>
<td>A123</td>
<td>DSNAME='PFPTEST.DB.&amp;AREA'</td>
<td>PFP.TEST.DB.A123</td>
</tr>
<tr>
<td>26</td>
<td>DSNAME='PFPTEST.DB.PF&amp;AREA#'</td>
<td>PFP.TEST.DB.PF026</td>
</tr>
<tr>
<td>26</td>
<td>DSNAME='PFPTEST.DB.PF&amp;AREA4#'</td>
<td>PFP.TEST.DB.PF0026</td>
</tr>
<tr>
<td>26</td>
<td>DSNAME='PFPTEST.DB.PF&amp;AREA5#'</td>
<td>PFP.TEST.DB.PF00026</td>
</tr>
<tr>
<td>1024</td>
<td>DSNAME='PFPTEST.DB.PF&amp;AREA#'</td>
<td>PFP.TEST.DB.PF1024</td>
</tr>
<tr>
<td>1024</td>
<td>DSNAME='PFPTEST.DB.PF&amp;AREA4#'</td>
<td>PFP.TEST.DB.PF1024</td>
</tr>
<tr>
<td>1024</td>
<td>DSNAME='PFPTEST.DB.PF&amp;AREA5#'</td>
<td>PFP.TEST.DB.PF1024</td>
</tr>
</tbody>
</table>

Using the &AREA#4 or &AREA5# variable instead of the &AREA# variable lets you standardize the length of dynamically allocated data set names when using 1000 or more areas, while still supporting area numbers 1 through 999.

**NOTE**

When using the UNLOAD command, the substituted value for &AREA is OAREAxxx or OARxxxxxx when the mask of &AREA is used with the DSNAME keyword on the OFILECTL subcommand.

When using the BUILD, VERIFY, RESYNC, or XSCAN command, the substituted value for &INDEX is the name of the index being processed by Fast Path Indexer/EP when the mask of &INDEX is used with the DSNAME keyword on the IX subcommand.

The following table provides examples of the resulting data set name that is dynamically allocated based on the specified area variable, and the area name or number of digits in the area number:

<table>
<thead>
<tr>
<th>Area name/number</th>
<th>Specified area name variable</th>
<th>Dynamically allocated data set name</th>
</tr>
</thead>
<tbody>
<tr>
<td>A123</td>
<td>DSNAME='PFPTEST.DB.&amp;AREA'</td>
<td>PFP.TEST.DB.A123</td>
</tr>
<tr>
<td>26</td>
<td>DSNAME='PFPTEST.DB.PF&amp;AREA#'</td>
<td>PFP.TEST.DB.PF026</td>
</tr>
<tr>
<td>26</td>
<td>DSNAME='PFPTEST.DB.PF&amp;AREA4#'</td>
<td>PFP.TEST.DB.PF0026</td>
</tr>
<tr>
<td>26</td>
<td>DSNAME='PFPTEST.DB.PF&amp;AREA5#'</td>
<td>PFP.TEST.DB.PF00026</td>
</tr>
<tr>
<td>1024</td>
<td>DSNAME='PFPTEST.DB.PF&amp;AREA#'</td>
<td>PFP.TEST.DB.PF1024</td>
</tr>
<tr>
<td>1024</td>
<td>DSNAME='PFPTEST.DB.PF&amp;AREA4#'</td>
<td>PFP.TEST.DB.PF1024</td>
</tr>
<tr>
<td>1024</td>
<td>DSNAME='PFPTEST.DB.PF&amp;AREA5#'</td>
<td>PFP.TEST.DB.PF1024</td>
</tr>
</tbody>
</table>

Using the &AREA#4 or &AREA5# variable instead of the &AREA# variable lets you standardize the length of dynamically allocated data set names when using 1000 or more areas, while still supporting area numbers 1 through 999.

**Default**

None

**Alias**

None
**DUMP**

**Purpose**

Use the DUMP command to request generation of a dump.

---

**NOTE**

Use this command only when instructed by BMC Customer Support.

---

**Use**

DUMP is an optional command for the Fast Path/EP operator interface.

**Available keyword**

DUMP_TYPE

**Syntax**

DUMP [keyword=parameter[,...]]

**Parameters**

None

**Default**

None

**Alias**

None

**DUMP_TYPE**

**Purpose**

Use the DUMP_TYPE keyword to specify the type of dump to be generated.
Use

DUMP_TYPE is an optional keyword for the DUMP command:

Syntax

DUMP_TYPE=parameter

Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABEND</td>
<td>Terminate the job, and generate a dump to the SYSUDUMP or SYSMDUMP data set.</td>
</tr>
<tr>
<td>SNAP</td>
<td>Generate a snap dump of storage areas to the data set indicated in the PFPSNAP DD statement.</td>
</tr>
<tr>
<td>SVC</td>
<td>Generate a system dump of the address space to the SYS1.DUMP data set.</td>
</tr>
</tbody>
</table>

Default

DUMP_TYPE=SVC

Alias

None

EARLY_TERMINATION

Purpose

Use the EARLY_TERMINATION keyword to specify whether post-processing tasks for the DEDB online restructure process should terminate before completion.

Use

EARLY_TERMINATION is an optional keyword for the following commands:

- GLOBAL
- PREPARE
- RESTART
- RESTRUCTURE
Related keyword

None

Syntax

EARLY_TERMINATION=parameter

Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONE</td>
<td>Post-processing runs until completion. Completion of post-processing activities will bring the database online.</td>
</tr>
<tr>
<td>BEFORE, RENAME_ADS</td>
<td>Post-processing terminates before the original and shadow area data sets are renamed, and after the CHANGE.ADS command is issued to DBRC.</td>
</tr>
<tr>
<td></td>
<td>If the DELTA IMS product is not active in your IMS system, the BEFORE,RENAME_ADS parameter is the only value allowed on the EARLY_TERMINATION keyword.</td>
</tr>
<tr>
<td>BEFORE, NOTIFY_IC</td>
<td>Post-processing terminates before the NOTIFY.UIC command is issued to DBRC.</td>
</tr>
<tr>
<td></td>
<td>For each restructured area, you can perform a batch image copy, and issue a NOTIFY.IC command to DBRC. Execute the RESTART command to continue post-processing with the /STA command.</td>
</tr>
<tr>
<td></td>
<td>The BEFORE,NOTIFY_IC parameter is not allowed on the EARLY_TERMINATION keyword under the following conditions:</td>
</tr>
<tr>
<td></td>
<td>■ The DELTA IMS product is not active in your IMS system.</td>
</tr>
<tr>
<td></td>
<td>■ Secondary shadow area data sets are defined on the PREPARE command.</td>
</tr>
<tr>
<td></td>
<td>For more information, see the Fast Path Online Restructure/EP User Guide.</td>
</tr>
<tr>
<td>BEFORE, START_DATABASE</td>
<td>Post-processing terminates before the restructured database is brought online.</td>
</tr>
<tr>
<td></td>
<td>If the DELTA IMS product is not active in your IMS system, the BEFORE, START_DATABASE parameter is not allowed on the EARLY_TERMINATION keyword.</td>
</tr>
<tr>
<td></td>
<td>For more information, see the Fast Path Online Restructure/EP User Guide.</td>
</tr>
</tbody>
</table>

Default

EARLY_TERMINATION=NONE
**END**

**Purpose**

Use the END command to signify the completion of a command set. The END command can be used at the end of a CTL card statement, or as a separator between command sets to improve readability.

**Use**

END is an optional command for all input sources. The END command is no longer required, but is valid for downward compatibility purposes.

**Related keywords**

None

**Syntax**

```
command [keyword=parameter[,...]]
[subcommand [keyword=parameter[,...]]]
[END]
```

**Parameters**

None

**Default**

None

**Alias**

None
**ERROR_THRESHOLD**

**Purpose**

Most pointer errors that are encountered during a reorganization or unload process will prevent processing of the area and will terminate the process. Certain pointer errors can be bypassed and allow processing to continue.

The ERROR_THRESHOLD keyword can be used to enable the primary command process to encounter and bypass a specified number of pointer errors without terminating. The ERROR_THRESHOLD keyword is used to specify the number of pointer errors that can be bypassed in the processing of an area before terminating the process.

When ERROR_THRESHOLD is used with the UNLOAD command, the database record is written up to the point of the segment in error, and processing continues with the next accessible segment. The segment containing the pointer error (and its dependent segments) is bypassed and is not written to the unload file. When the number of errors encountered in all database records that have been read exceeds the value specified on the ERROR_THRESHOLD keyword, processing of the area terminates.

When ERROR_THRESHOLD is used with the REORGANIZE command, any UOW where a pointer error is detected is bypassed (not reorganized), and 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.

**Use**

ERROR_THRESHOLD is an optional keyword for the following commands:

- CHANGE
- RELOAD
- REORGANIZE
- UNLOAD

**Related keyword**

SDEP_PROCESS
WARNING

When you specify SDEP_PROCESS=PHYSICAL on the UNLOAD command, the process does not read SDEPs by following pointers. Consequently, pointer errors in the SDEP portion of the database will not be detected. If you want the ERROR_THRESHOLD keyword to detect and bypass any pointer errors detected in the SDEP portion of the DEDB, then you must specify SDEP_PROCESS=LOGICAL.

Syntax

ERROR_THRESHOLD=parameter

Parameters

Specify a value from 0 to 99999.

Default

ERROR_THRESHOLD=0

Alias

ERT

EXCEPTION_LIMIT

Purpose

Use the EXCEPTION_LIMIT keyword to control exception toleration during command processing.

The EXCEPTION_LIMIT keyword is used to enable the primary command process to bypass a specified number of segment errors on an input area without terminating. The segment that is in error, and all of its dependent segments are bypassed, and processing of the input area continues. Once the number of segment errors exceeds the EXCEPTION_LIMIT, the command process terminates for that input area.

The bypassed segments are written to the discard file (for user processing at a later time) only if the EXCEPTION_LIMIT keyword and DISCARD_FILECTL subcommand are specified. For information about the record layout of the discard file, see Appendix D, “Discard file record layout.”

Use

EXCEPTION_LIMIT is an optional keyword for the RELOAD command.
Related keyword

BYPASS_RECORD

Syntax

EXCEPTION_LIMIT=parameter

Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Exceptions are not tolerated for this value. The primary command will terminate if an exception is encountered.</td>
</tr>
<tr>
<td>1 to 2,147,483,647</td>
<td>Exceptions are tolerated for this range of values. The primary command will continue to process the input file within this range.</td>
</tr>
<tr>
<td>INFINITE</td>
<td>This value is the same as 2,147,483,647. There is not a limit to the number of exceptions that will be tolerated. The primary command will continue to process the input file.</td>
</tr>
</tbody>
</table>

Default

EXCEPTION_LIMIT=0

Alias

None

EXCLUDE

Purpose

Use the EXCLUDE subcommand to specify segment exclusion criteria for processing this segment and its dependents by the associated command.
Use

EXCLUDE is an optional subcommand for the following commands:

- CHANGE
- EXTRACT
- RELOAD
- UNLOAD

Available keywords

- SAMPLE_INTERVAL
- SAMPLE_LIMIT
- SEGMENT (required)
- WHERE

Syntax

```
cmd [keyword=parameter[...]]
EXCLUDE SEGMENT=parameter,[keyword=parameter[,...]]
```

Parameters

None

Default

If the EXCLUDE subcommand is not specified, all segments are selected.

Alias

None

**EXPAND**

Purpose

Use the EXPAND keyword to force expansion of compressed segments before the segments are passed to the output process. The optional EXPAND keyword can be used with several commands to perform the following functions:

- EXPAND on the CHANGE or RELOAD command expands the segment before sending it to output, where it will be recompressed if a compression routine is defined for the segment in the DBD.
- EXPAND on the EXTRACT or UNLOAD command expands the segment before writing it to the output file. The output file is then in expanded format.

- EXPAND on the CHANGE or UNLOAD command can be used to change the compression routine.

The EXPAND keyword is effective only with compression products that use compression defined in the DBD. If you are using a compression product that controls compression by something other than the DBD, the EXPAND keyword has no effect.

**Use**

EXPAND is an optional keyword for the following commands:

- CHANGE
- EXTRACT
- RELOAD
- UNLOAD

**Related keywords**

- COMPRESS
- FIELDS

**Syntax**

```
EXPAND=parameter
EXPAND=(parameter1, parameter2, ... , parametern)
```

**Parameters**

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not expand the data before the segment is processed.</td>
</tr>
<tr>
<td>YES</td>
<td>Expand the data before the segment is processed.</td>
</tr>
<tr>
<td>1-character to 8-character segment name</td>
<td>Expand only the specified segment or segments. This value can be used more than once to specify specific segments using the (parameter1, parameter2, ..., parametern) syntax shown above.</td>
</tr>
</tbody>
</table>
### Purpose

Use the EXPDT keyword to specify the expiration date an output data set created by dynamic allocation.

**NOTE**

Do not specify both the EXPDT keyword and the RETPD keyword; these keywords are mutually exclusive.

The EXPDT keyword can also be used with the PFPEPR00 batch utility program to specify or modify an expiration date for an allocation rule stored in the repository catalog associated with the Fast Path Analyzer/EP repository facility.

### Use

EXPDT is an optional keyword for the following subcommands:

- ACTIVITY_FILECTL
- ADD
- DISCARD_FILECTL
- IC
- MODIFY
- OFILECTL
- PLAN_FILECTL
**Syntax**

`EXPDT=date`

**Parameters**

Specify a date in either of the following formats:

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>yyddd</code></td>
<td><code>yy</code> is the 2-digit year. <code>ddd</code> is the 3-digit Julian day of the year.</td>
</tr>
<tr>
<td><code>yyyy/ddd</code></td>
<td><code>yyyy</code> is the 4-digit year <code>ddd</code> is the 3-digit Julian day of the year.</td>
</tr>
</tbody>
</table>

**Default**

None

**Alias**

None

**EXTEND**

**Purpose**

Use the EXTEND command to increase the size of IOVF and SDEP portions of a DEDB without reorganizing the database.

The EXTEND command is available for both offline and online processing.

**NOTE**

For online processing, the EXTEND command can be used to extend IOVF with SDEPs defined.

**Use**

EXTEND is an optional command for the PFPSYSIN DD statement. You must have a license for Fast Path Online Reorg/EP or Fast Path Reorg/EP to use this command.
Available keywords

- DBD
- EXTEND_IOVF
- EXTEND_SDEP
- IAREA
- INPUT_DSN_MASK
- LARGEST_DATABASE_RECORDS
- MESSAGE_SUPPRESSION
- ORPHANED_SDEP_MSG
- POINTER_VALIDATION
- RAP_VALIDATION
- SDEP_VALIDATION
- TYPE_RUN

**NOTE**
At least one of the keywords, EXTEND_IOVF or EXTEND_SDEP, is required.

Syntax

```
EXTEND [keyword=parameter...]
      [subcommand [keyword=parameter[...]]]
```

Parameters

None

Default

None

Alias

None

**EXTEND_IOVF**

**Purpose**

Use the EXTEND_IOVF keyword to specify the type of allocation unit and the number of allocation units to add to the IOVF storage portion of the DEDB.
If SDEPs are defined in the DBD, you can use the EXTEND_IOVF keyword to extend the IOVF area online or offline.

**Use**

EXTEND_IOVF is an optional keyword for the EXTEND command.

---

**NOTE**

At least one of the keywords, EXTEND_IOVF or EXTEND_SDEP, is required for the EXTEND command.

**Related keyword**

EXTEND_SDEP

**Syntax**

```plaintext
EXTEND_IOVF=(units, quantity)
```

**Parameters**

For the `units` parameter, specify one of the following allocation unit values for the extension:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTROL_INTERVALS or (CIS,CI)</td>
<td>extend IOVF by using control intervals</td>
</tr>
<tr>
<td>CONTROL_AREAS or (CAS,CA)</td>
<td>extend IOVF by using control areas</td>
</tr>
<tr>
<td>UNITS_OF_WORK or (UOWS, UOW)</td>
<td>extend IOVF by using units of work</td>
</tr>
<tr>
<td>CYLINDERS or (CYLS,CYL)</td>
<td>extend IOVF by using cylinders</td>
</tr>
</tbody>
</table>

For the `quantity` parameter, specify the quantity (number) of allocation units to add during the extension.

**Default**

```plaintext
EXTEND_IOVF=(UOWS,0)
```

**Alias**

IOVF
EXTEND_IOVF_='UOWS

Purpose

Use the EXTEND_IOVF_='UOWS keyword to specify the number of UOWs to add to the IOVF storage portion of the DEDB.

If SDEPs are defined in the DBD, use of the EXTEND_IOVF_='UOWS keyword will differ depending on whether you are performing online or offline:

- If SDEPs are defined in the DBD and you specify the EXTEND_IOVF_='UOWS keyword on a REORGANIZE command set, Fast Path/EP will automatically apply the SELECT_UOW=ALL keyword value to ensure that all SDEP pointers are updated.

- If SDEPs are defined in the DBD, you cannot use the EXTEND_IOVF_='UOWS keyword with an online REORGANIZE command set.

Use

EXTEND_IOVF_='UOWS is an optional keyword for the REORGANIZE and EXTEND commands.

Related keywords

- FRAGMENTATION_PERCENT
- EXTEND_SDEP_='CIS
- SELECT_UOW

Syntax

EXTEND_IOVF_='UOWS=parameter

Parameters

Specify a value from 0 to 32765.

Default

EXTEND_IOVF_='UOWS=0

Alias

XIOVF
EXTEND_SDEP

Purpose

Use the EXTEND_SDEP keyword to specify the type of allocation unit and the number of allocation units to add to the SDEP storage portion of the DEDB.

Use

EXTEND_SDEP is an optional keyword for the EXTEND command.

NOTE

At least one of the keywords, EXTEND_SDEP or EXTEND_IOVF, is required for the EXTEND command.

Related keyword

EXTEND_IOVF

Syntax

EXTEND_IOVF=(units, quantity)

Parameters

For the units parameter, specify one of the following allocation unit values for the extension:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTROL_INTERVALS or (CIS,Ci)</td>
<td>extend SDEP by using control intervals</td>
</tr>
<tr>
<td>CONTROL AREAS or (CAS,CA)</td>
<td>extend SDEP by using control areas.</td>
</tr>
<tr>
<td>UNITS_OF_WORK or (UOWS, UOW)</td>
<td>extend SDEP by using units of work</td>
</tr>
<tr>
<td>CYLINDERS or (CYLS,CYL)</td>
<td>extend SDEP by using cylinders</td>
</tr>
</tbody>
</table>

For the quantity parameter, specify the quantity (number) of allocation units to add during the extension.

Default

EXTEND_SDEP=(CIS,0)
**EXTEND_SDEP_#CIS**

**Purpose**

Use the EXTEND_SDEP_#CIS keyword to specify the number of CIs to add to the SDEP storage portion of the DEDB.

Fast Path Reorg (offline or online) automatically moves the SDEP segments between the SDEP Logical Begin and the original end of area, to the new (extended) area, if all of the following conditions exist:

- You specify the EXTEND_SDEP_#CIS keyword with an offline or online EXTEND command set.
- The cycle count portion of the SDEP Logical End is greater than the cycle count portion of the SDEP Logical Begin.
- The number of SDEP CIs that are added to the area exceeds the number of SDEP CIs that are between the SDEP Logical Begin and the original end of area.

Otherwise, the existing SDEP segments remain in their original position.

**Use**

EXTEND_SDEP_#CIS is an optional keyword for the REORGANIZE and EXTEND commands.

**Related keyword**

- EXTEND_IOVF_#UOWS
- FRAGMENTATION_PERCENT

**Syntax**

```
EXTEND_SDEP_#CIS=parameter
```

**Parameters**

Specify a value from 0 to 8388601.
Default

EXTEND_SDEP_#CIS=0

Alias

XSDEP

**EXTRACT**

**Purpose**

Use the EXTRACT command to extract data from a DEDB into a sequential file. The command can be used in online or offline mode.

**Use**

EXTRACT is an optional command for the PFPSYSIN DD statement. You must have a license for Fast Path Online Analyzer/EP or Fast Path Analyzer/EP to use this command.

**Available keywords**

- DBD
- EXPAND
- EXTRACT_FORMAT
- IAREA
- ICACHE (offline mode only)
- INPUT_DSN_MASK (offline mode only)
- OUTPUT_DSN_MASK
- SEGMENT_RECORD_PREFIX
- SEGMENT_RECORD SUFFIX
- SORT
- SORT_OPTION

**Syntax**

```
EXTRACT [keyword=parameter[,...]]
    [subcommand [keyword=parameter[,...]]]
```

**Parameters**

None
Default

None

Alias

None

**EXTRACT_FORMAT**

**Purpose**

Use the EXTRACT_FORMAT keyword to specify the format of the extract file created by the EXTRACT command.

**Use**

EXTRACT_FORMAT is an optional keyword for the EXTRACT command.

**Related keywords**

- SEGMENT_RECORD_PREFIX
- SEGMENT_RECORD_SUFIX

**Syntax**

`EXTRACT_FORMAT=parameter`

**Parameters**

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTRACT</td>
<td>The file is in the standard format for the Fast Path/EP offline or online DEDB data extract process. This format is defined in Appendix B, “DEDB Data Extract record layout.”</td>
</tr>
<tr>
<td>HDUNLOAD</td>
<td>The file is in a format compatible with the IBM HD Reorganization Unload utility.</td>
</tr>
<tr>
<td>USER</td>
<td>The file is in a format that is specified by using the SEGMENT_RECORD_PREFIX and SEGMENT_RECORD_SUFIX keywords, and the OUTPUT and USER_RECORD subcommands.</td>
</tr>
</tbody>
</table>
FIELDS

Default

EXTRACT_FORMAT=EXTRACT

Alias

FORMAT

FIELDS

Purpose

Use the FIELDS keyword to specify the contents the output record written by the command function. Extracted fields are in the output record in the order specified by the parameter that is coded on the FIELDS keyword.

When used on an OUTPUT subcommand within an EXTRACT command, this keyword specifies the content of the segment data records written to the extract file. When used on an OUTPUT subcommand within a CHANGE, UNLOAD, or RELOAD command, this keyword specifies the content of the database segments.

**NOTE**

When the FIELDS keyword is used with the OUTPUT subcommand to refer to the content of a compressed segment, you must specify EXPAND=YES on the primary command. This will ensure that the operands specified on the FIELDS keyword correspond to the proper segment columns.

Use

FIELDS is an optional keyword for the OUTPUT subcommand.

FIELDS is a required keyword (when used within a PREPARE command) on each of the OUTPUT subcommands for a segment if you modify the position or length of the symbolic key field.

FIELDS is a required keyword for the USER_RECORD subcommand.

Related keyword

EXPAND

Syntax

FIELDS=(expression[,...][,VL=YES/NO])
Parameters

The parameter consists of a list of one or more expressions, each of which defines a value to be placed in the output record. For details, see Chapter 3, “Expression syntax.”

The values resulting from each expression in the parameter list are placed in the output area in the order specified. Unless otherwise directed by use of an explicit conversion operator, the format, size and precision of the value(s) is derived from the operands and operators used in the expression. For details, see Chapter 3, “Expression syntax.”

The entire list of values resulting from the expression or expressions can be optionally prefixed with a 2-byte length indicator. Specify VL=YES as a subparameter on the FIELDS keyword to request a length prefix; specify VL=NO to suppress the length prefix.

Default

VL=NO will be used by default when running an EXTRACT command, unless EXTRACT_FORMAT=HDUNLOAD is selected.

When running a CHANGE, UNLOAD, or RELOAD command, or when running an EXTRACT command with EXTRACT_FORMAT=HDUNLOAD selected, VL=YES will be used by default for segments defined as variable length. VL=NO will be used by default for segments defined as fixed length.

Alias

None

FLOWER_BOX

Purpose

Use the FLOWER_BOX keyword to turn on or off the flower box borders around messages in the JES2 processing log.

NOTE

Turning off the flower box borders also suppresses messages in the JES2 processing log.
**Use**

FLOWER_BOX is an optional keyword for the GLOBAL command.

**Syntax**

FLOWER_BOX=\textit{parameter}

**Parameters**

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES (turn on)</td>
<td>Print flower box borders around messages.</td>
</tr>
<tr>
<td>NO (turn off)</td>
<td>Do not print flower box borders around messages.</td>
</tr>
</tbody>
</table>

**NOTE**

Specifying NO also suppresses messages in the JES2 processing log.

**Default**

FLOWER_BOX=YES

**Alias**

None

**FORCE**

**Purpose**

Use the FORCE keyword to initiate the immediate shutdown of a product or task without waiting for an end-of-task.

**NOTE**

You must attempt a shutdown without the FORCE keyword before attempting a shutdown with the FORCE keyword.
Use

FORCE is an optional keyword for the SHUTDOWN command.

Related keywords

- ID
- PRODUCT

Syntax

FORCE=parameter

Parameters

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not force product or task shutdown.</td>
</tr>
<tr>
<td>YES</td>
<td>Force product or task shutdown.</td>
</tr>
</tbody>
</table>

Default

FORCE=NO

Alias

None

FORMAT

Purpose

Use the FORMAT keyword to specify the format of the unload file.

Use

FORMAT is an optional keyword for the UNLOAD command.

Syntax

FORMAT=parameter
FRAGMENTATION_PERCENT

Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDUNLOAD</td>
<td>File is in format that is compatible with the IBM Reorganization Unload utility.</td>
</tr>
<tr>
<td>MSDBINIT</td>
<td>File is in format that is compatible with the MSDB Initialization utility.</td>
</tr>
<tr>
<td>TFMT</td>
<td>File is in format that is compatible with the TRIMAR FAST PATH UNLOAD/RELOAD product.</td>
</tr>
<tr>
<td>DBT</td>
<td>File is in format that is compatible with the DEDB Reload Utility component of the IMS Fast Path Basic Tools for OS/390.</td>
</tr>
</tbody>
</table>

Default

FORMAT=HDUNLOAD

Alias

None

Purpose

Use the FRAGMENTATION_PERCENT keyword to specify the allowable percentage of disorganization within a UOW. The reorganization function uses this percentage in the selection of UOWs for reorganization. Factors that contribute to disorganization in a UOW are the number of free space elements (FSEs) and the number of segments stored out of sequence.

In selecting UOWs for reorganization, the function first checks the parameter of the SELECT_UOW keyword. If the UOW meets the criteria specified by the SELECT_UOW keyword, the reorganization function then determines (using an algorithm) if the UOW meets or exceeds the value specified for FRAGMENTATION_PERCENT. If the UOW meets or exceeds this parameter, the UOW is selected and reorganized.

Use

FRAGMENTATION_PERCENT is an optional keyword for the REORGANIZE command.
Related keyword

SELECT_UOW

Syntax

FRAGMENTATION_PERCENT=parameter

Parameters

Specify a value from 1 to 100. Reasonable values range from 1 to 5 percent. Fragmentation of greater than 5 percent is unlikely under normal circumstances.

Default

None

NOTE

If you do not specify the FRAGMENTATION_PERCENT keyword, the fragmentation test is not performed.

Alias

None

FREESPACE_ANALYSIS

Purpose

Use the FREESPACE_ANALYSIS keyword to request the Free Space Analysis report. This report shows the available and usable percentage of free space of each DEDB area section, including RAA base, DOVF, IOVF, and SDEP.

Use

FREESPACE_ANALYSIS is an optional keyword for the REPORT subcommand.

Related keyword

REPORT_DEFAULT
FREESPACE_DOVF_IOVF

Syntax

FREESPACE_ANALYSIS=\textit{parameter}

Parameters

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not produce the report.</td>
</tr>
<tr>
<td>YES</td>
<td>Produce the report.</td>
</tr>
</tbody>
</table>

Default

The default parameter is set by the REPORT_DEFAULT keyword.

Alias

FSA

FREESPACE_DOVF_IOVF

Purpose

Use the FREESPACE_DOVF_IOVF keyword to specify a threshold setting when analyzing or monitoring DEDB activity. The percentage of free space within DOVF and IOVF blocks is computed and compared with user-specified thresholds. If the computed DOVF free space is greater than the user-specified setting and the computed IOVF free space is less than the user-specified setting, a warning message is issued.

\textbf{NOTE}

If you do not specify the FREESPACE_DOVF_IOVF keyword, this threshold test is not performed.

Use

FREESPACE_DOVF_IOVF is an optional keyword for the THRESHOLD subcommand.
FREESPACERAA_DOVF

Syntax

FREESPACERAA_DOVF = (parameter1, parameter2)

Parameters

Specify a value from 0 to 100 for each parameter, where \textit{parameter1} is the percentage of DOVF and \textit{parameter2} is the percentage of IOVF.

Default

None

Alias

FDI

FREESPACERAA_DOVF

Purpose

Use the FREESPACERAA_DOVF keyword to specify a threshold setting when analyzing or monitoring DEDB activity. The percentage of free space within RAA and DOVF blocks is computed and compared with user-specified thresholds. If the computed RAA free space is greater than the user-specified setting and the computed DOVF free space is less than the user-specified setting, a warning message is issued.

\textbf{NOTE}

If you do not specify the FREESPACERAA_DOVF keyword, this threshold test is not performed.

Use

FREESPACERAA_DOVF is an optional keyword for the THRESHOLD subcommand.

Syntax

FREESPACERAA_DOVF = (parameter1, parameter2)
**FREESPACE_RAA_IOVF**

**Parameters**

Specify a value from 0 to 100 for each parameter, where `parameter1` is the percentage of RAA and `parameter2` is the percentage of DOVF.

**Default**

None

**Alias**

FRD

**FREESPACE_RAA_IOVF**

**Purpose**

Use the FREESPACE_RAA_IOVF keyword to specify a threshold setting when analyzing or monitoring DEDB activity. The percentage of free space within RAA and IOVF blocks is computed and compared with user-specified thresholds. If the computed RAA free space is greater than the user-specified setting and the computed IOVF free space is less than the user-specified setting, a warning message is issued.

**NOTE**

If you do not specify the FREESPACE_RAA_IOVF keyword, this threshold test is not performed.

**Use**

FREESPACE_RAA_IOVF is an optional keyword for the THRESHOLD subcommand.

**Syntax**

FREESPACE_RAA_IOVF=(parameter1,parameter2)

**Parameters**

Specify a value from 0 to 100 for each parameter, where `parameter1` is the percentage of RAA and `parameter2` is the percentage of IOVF.
FULLSEG

Purpose

Use the FULLSEG keyword to specify whether the full segment image is logged in the X'5950' log record when the segment is updated by a Replace (REPL) call.

Use

FULLSEG is an optional keyword for the REGISTER subcommand.

Syntax

\[
\text{FULLSEG} = \text{NO} \\
\text{FULLSEG} = \text{YES}
\]

Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>indicates that only the updated portion of the segment, not the full segment, is logged in the X'5950' log record when the segment is updated</td>
</tr>
<tr>
<td>YES</td>
<td>indicates the full segment image is logged in the X'5950' log record when the segment is updated</td>
</tr>
</tbody>
</table>

Default

The default value is NO.

Alias

None
**GENMAX**

**Purpose**

Use the GENMAX keyword to specify the maximum number of image copies that DBRC should maintain for the specified DBDS.

**Use**

GENMAX is an optional keyword for the REGISTER subcommand.

**Related keywords**

None

**Syntax**

```
GENMAX=parameter
```

**Parameters**

The parameter specifies the number of image copies.

**Default**

The default value is 2.

**Alias**

None

---

**GSGNAME**

**Purpose**

Use the GSGNAME keyword to specify the global service group to which a database is assigned.

**Use**

GSGNAME is an optional keyword for the REGISTER subcommand.
Related keywords

None

Syntax

GSGNAME=parameter

Parameters

The parameter specifies the name of the global service group.

Default

None

Alias

None

GLOBAL

Purpose

Use the GLOBAL command to establish implicit job step keyword values. The implicit keyword values set with the GLOBAL command are in effect unless overridden. To override an implicitly set keyword value for a particular command, specify the keyword explicitly with the command.

You can specify the GLOBAL command in the PFPSYSIN DD input stream or from the GLOBAL module from the PFPDFLTS DD statement. If specified for the PFPSYSIN DD input stream, GLOBAL must be the first command in the input stream.

NOTE

You can specify multiple GLOBAL commands in the PFPSYSIN DD input stream; however, they must precede all other commands.

If both PFPDFLTS and PFPSYSIN GLOBAL commands are specified in the JCL, then a keyword specified in the input stream for PFPSYSIN DD GLOBAL command will override the same keyword specified in the GLOBAL module for the PFPDFLTS DD statement.
GLOBAL

Use

GLOBAL is an optional command for the PFPSYSIN DD statement or the GLOBAL module from the PFPDFLTS DD statement.

Available keywords

- ACCESS
- DBRC
- EARLY_TERMINATION
- FLOWER_BOX
- HISTORY_DDNAME
- LARGEST_DATABASE_RECORDS
- MESSAGE_SUPPRESSION
- ORPHANED_SDEP_MSG
- OUTAGE_WINDOW
- POINTER_VALIDATION
- RAP_VALIDATION
- SCAN
- SDEP_VALIDATION
- TYPE_RUN

Syntax

GLOBAL [keyword=parameter[,...]]
   [subcommand [keyword=parameter[,...]]]

Parameters

None

Default

None

Alias

None
GROUP_KEY

Purpose

Use the GROUP_KEY keyword to specify a group identifier. A group identifier is used as part of the record key for the repository catalog data set. It is used when adding a group level, a DBD level, and an area level allocation rule.

Use

GROUP_KEY is a required keyword for the following subcommands.

- ADD GROUP_ALLOCATION
- ADD DBD_ALLOCATION
- ADD AREA_ALLOCATION

Related keywords

- AREA_KEY
- DBD_KEY

Syntax

GROUP_KEY=parameter

Parameters

Specify a 1-character to 4-character group name.

Default

None

Alias

None

HELP

Purpose

Use the HELP command to request brief descriptions of the Fast Path/EP console commands (DISPLAY, DUMP, SHUTDOWN, and TRACE).
Use

HELP is an optional command for the Fast Path/EP operator interface.

Syntax

HELP
[END]

NOTE
There are no keywords for the HELP command.

Parameters

None

Default

None

Alias

None

HISTORY_DDNAME

Purpose

Use the HISTORY_DDNAME keyword to specify the ddname(s) of a DD statement(s) specified in the JCL to which a statistical summary record is written. If the JCL does not contain a DD statement(s) with the referenced ddname(s), an error message is generated. If the ddname(s) refers to a data set that contains data, the summary record is appended to the existing data (as if you had specified DISP=MOD). For information about the record layout, see Appendix A, “History file record layout.”

NOTE
All fields are written to the History File only when POINTER_VALIDATION=FULL is specified. If any other value is specified for the POINTER_VALIDATION keyword, certain fields in the History File will be set to zero.
The data set referenced by this ddname(s) must be a physical sequential data set. It cannot be a member of a partitioned data set. The record format (RECFM) supports fixed or variable length records that can be blocked or unblocked. The statistics summary records written to the data set contain 200 bytes of data. The logical record length (LRECL) should be 200 for RECFM=FB and 204 for RECFM=VB. The block size (BLKSIZE) must be appropriate for the RECFM and LRECL.

**NOTE**

If critical errors are detected during analysis, a statistics summary record is not generated for the area.

**Use**

HISTORY_DDNAME is an optional keyword for the following commands and subcommands:

- ANALYZE
- LIST
- PREPARE
- RETRIEVE
- GLOBAL

**Related keywords**

- POINTER_VALIDATION
- RAP_VALIDATION

**Related subcommand**

REPORT

**Syntax**

```
HISTORY_DDNAME=parameter
HISTORY_DDNAME=(parameter[,parameter[, ...]])
```

**Parameters**

Specify a 1-character to 8-character ddname.

**Default**

None
Alias

The following are aliases of the HISTORY_DDNAME keyword:

- EXTDD
- EXTRACT_DDNAME
- HISDD

**IAREA**

**Purpose**

Use the IAREA keyword to specify one or more input areas, a range of input areas, or a combination of input areas and input area ranges to the process.

**Use**

IAREA is an optional keyword for the following commands:

- ANALYZE
- BUILD
- CHANGE
- DMAC_PRINT
- EXTEND
- EXTRACT
- IMAGECOPY
- INITIALIZE
- PROCESS_AREA
- RELOAD
- REORGANIZE
- UNLOAD
- VERIFY (not valid in BMP mode)
- XSCAN

**Related keyword**

OAREA

**Syntax**

- IAREA=ALL or IAREA=*  
- IAREA=parameter  
- IAREA='parameter'
IAREA=(parameter1, parameter2, ..., parametern)
IAREA=(RANGE=(parameter1, parameter2))
IAREA=(RANGE=(parameter1, parameter2), RANGE=(parameter3, parameter4), ...)
IAREA=(RANGE=(*, parametern))
IAREA=(RANGE=(parameter1, *))

**NOTE**

IAREA='parameter' includes wildcard masks. For example, the syntax for areas DPAS7A0, DPAS7B0, and DPAS7C0 can be specified as follows:

- IAREA='DP????C0'
- IAREA='DPAS7*'
- IAREA='DPA7??'
- IAREA='DPAS7??'

**Parameters**

Specify either IAREA=ALL or IAREA=*, or any combination of the following parameter values:

**NOTE**

IAREA=ALL and IAREA=* can be used to specify all areas of the DEDB.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>areaname</td>
<td>Specify one or more areas by using the 1-character to 8-character area name for each area specified.</td>
</tr>
<tr>
<td>areaname mask</td>
<td>You can also specify one or more areas by using the 1-character to 8-character area name mask for each area specified. Wildcard masks using the asterisk (*) and question mark (?) can be included. When using wildcard masks, the value of the keyword must be enclosed in quotation marks. The asterisk wildcard matches all characters to the right of the asterisk. The question mark wildcard matches the characters in the string in the place the question mark holds.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>areanumber</td>
<td>Specify one or more areas by using the 1-character to 5-character area number for each area specified.</td>
</tr>
<tr>
<td>RANGE=(startarea,endarea)</td>
<td>Use this syntax to specify a consecutive range of areas using either areaname or areanumber parameters. The area number associated with startarea must be less than (and not equal to) the area number associated with endarea.</td>
</tr>
<tr>
<td>RANGE=(*,endarea)</td>
<td></td>
</tr>
<tr>
<td>RANGE=(startarea,*)</td>
<td></td>
</tr>
</tbody>
</table>

| Note: Wildcard masks cannot be used with the RANGE keyword. |

**Default**

IAREA=ALL

**Alias**

AREA is an alias for IAREA only under the following commands:

- ANALYZE
- DMAC_PRINT
- EXTEND
- IMAGECOPY
- PROCESS_AREA
- REORGANIZE
- INITIALIZE

**IC**

**Purpose**

Use the IC subcommand to request that one or more image copies be created during command processing. The image copies requested are created for each area processed.

The image copy output file can be created within your JCL (refer to the DDNAME keyword), or by dynamic allocation (refer to the DSNAME keyword). If you specify the IC subcommand without specifying either of these keywords, no image copy will be produced. This feature can be used to bypass the creation an image copy if the IC subcommand was specified on the GLOBAL command.
Any IC subcommand or subcommands that are specified under the GLOBAL command are inherited by all other commands within the command stream unless those commands contain their own IC subcommand or subcommands.

**Use**

IC is a required subcommand for the IMAGECOPY command. IC is an optional subcommand for the following commands:

- ANALYZE
- CHANGE
- EXTEND
- GLOBAL
- INITIALIZE
- REORGANIZE
- UNLOAD
- RELOAD

---

**NOTE**

To execute the IC subcommand in online mode, you must have a license for Fast Path Online Image Copy/EP.

**Available keywords**

- AVGREC
- COMPRESSION
- DATACLAS
- DDNAME (required if DSNAME keyword is not specified)
- DISP
- DSNAME (required if DDNAME keyword is not specified)
- EXPDST
- LIKE
- MGMTCLAS
- NOTIFY
- RETPD
- SPACE
- STACK_NAME
- STORCLAS
- UNIT
- VOLCNT
- VOLSER
ICACHE

Syntax

command [keyword=parameter[...]]
IC [keyword=parameter[...]]

Parameters

None

Default

None

Alias

None

ICACHE

Purpose

Use the ICACHE keyword to specify the types of control intervals that are to be preloaded into storage for processing. A dataspace is used to implement the input cache.

Use

ICACHE is an optional keyword for the following commands:

- ANALYZE
- BUILD
- CHANGE
- EXTRACT
- PROCESS_AREA
- UNLOAD
- VERIFY
- XSCAN

NOTE

The ICACHE keyword applies to processing in offline mode only. It is ignored during online and BMP processing.
Chapter 2 Command language

Related keyword

OCACHE

Syntax

ICACHE=parameter
ICACHE=(parameter1, parameter2)

Parameters

Specify one or more of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOVF</td>
<td>Preload all non-empty IOVF and space map control intervals.</td>
</tr>
<tr>
<td>SDEP</td>
<td>Preload all non-empty SDEP control intervals.</td>
</tr>
</tbody>
</table>

Default

If the ICACHE keyword is not specified, no input caching is performed. When an image copy is used as input to the CHANGE or UNLOAD processes, however, both IOVF and SDEP control intervals are preloaded (ICACHE=IOVF,SDEP).

Alias

None

ICJCL

Purpose

Use the ICJCL keyword to specify the name of a partitioned data set member that contains skeletal JCL. When you issue the GENJCL.IC command, DBRC uses the member that you specify to generate the JCL that runs the Database Image Copy utility for the identified DBDS or DEDB.

Use

ICJCL is an optional keyword for the REGISTER subcommand.

Related keywords

None
### Syntax

```
ICJCL=parameter
```

### Parameters

The parameter specifies the member name.

### Default

The default value is ICJCL.

### Alias

None

---

### ID

#### Purpose

Use the ID keyword to request a specific task for a DISPLAY or a SHUTDOWN command. You can use either of the following methods to obtain the value for the ID keyword: 1) Use the DISPLAY command with the OBJECT=PRODUCT keyword. 2) Refer to the number in parentheses after the Fast Path/EP command that is issued in the text of message BMC110011I. In the following example, the ID=6.

```
BMC110011I Command REORGANIZE (TASK 6) is in progress
```

#### Use

ID is a required keyword for the SHUTDOWN command. ID is an optional keyword for the DISPLAY command:

#### Related keywords

- FORCE
- OBJECT
- PRODUCT

### Syntax

```
ID=parameter
```
Parameters

Specify a value from 1 to 250.

Default

None

Alias

None

IDCAMS_OPTION

Purpose

Use the IDCAMS_OPTION keyword to pass optional parameters to Access Method Services (AMS) for the VSAM cluster definition.

Use

IDCAMS_OPTION is an optional keyword for the ALLOCATE subcommand.

Syntax

IDCAMS_OPTION=parameter
IDCAMS_OPTION=(parameter[,parameter...])

Parameters

Numerous parameters are available for use with this keyword to enable you to control the values used by AMS for the VSAM cluster definition. For more information, see the IBM Access Method Services Reference Manual.

NOTE

If a parameter contains blanks or other special characters, it must be enclosed in single or double quotation marks.

Default

IDCAMS_OPTION='SHAREOPTIONS(3 3)'

Chapter 2 Command language 177
Alias

AMSOPT

**IFP_ACCOUNT**

**Purpose**

Use the IFP_ACCOUNT keyword to specify whether to display accounting information when starting an IFP region.

**Use**

IFP_ACCOUNT is an optional keyword for the OPTIONS command.

**Syntax**

IFP_ACCOUNT=parameter

**Parameters**

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>Display accounting information when starting an IFP region.</td>
</tr>
<tr>
<td>NO</td>
<td>Do not display accounting information when starting an IFP region.</td>
</tr>
</tbody>
</table>

**Default**

None

**Alias**

None

**IFP_JOBNAME**

**Purpose**

Use the IFP_JOBNAME keyword to specify the job name for the IFP region.
Use

IFP_JOBNAME is an optional keyword for the OPTIONS command.

Syntax

IFP_JOBNAME=parameter

Parameters

Specify a 1-character to 8-character job name. The following rules apply to the job name:

- The first character must be alphabetic or national ("$", ",", ",", or ",@").
- Subsequent characters can be alphabetic, numeric, or national.
- The last character can be an asterisk (*). If an * is used, the parameter must be enclosed in single or double quotation marks.

Default

When you use IFP_JOBNAME with the Fast Path Online Restructure/EP product, the default is the job name that you used with the PREPARE and RESTRUCTURE command.

Otherwise, IFP_JOBNAME does not have a default.

Alias

None

IFP_LIMIT

Purpose

Use the IFP_LIMIT keyword to specify the maximum number of concurrent IFP regions created during the area copy tasks that are generated by the Restructure function.

Use

IFP_LIMIT is an optional keyword for the OPTIONS command.
Syntax

IFP_LIMIT=parameter

Parameters

Specify a value from 1 to 256.

Default

IFP_LIMIT=4

Alias

None

IFP_PROCNAME

Purpose

Use the IFP_PROCNAME keyword to specify the name of the cataloged procedure that initiates the started task for the IFP regions.

Use

IFP_PROCNAME is an optional keyword for the OPTIONS command.

Syntax

IFP_PROCNAME=parameter

Parameters

Specify a 1-character to 8-character procedure name. The following rules apply to the procedure name:

- The first character must be alphabetic or national ("$", "_", "#", or "@").
- Subsequent characters can be alphabetic, numeric, or national.
- The last character can be an asterisk (*). If an * is used, the parameter must be enclosed in single or double quotation marks.

Default

IFP_PROCNAME=IEESYSAS
IMAGECOPY

Purpose

Use the IMAGECOPY command to create one or more image copies of an online DEDB. Use the IC subcommand to control the characteristics of the image copy data sets. The IMAGECOPY command can be used in online mode only.

Use

IMAGECOPY is an optional command for the PFPSYSIN DD statement. You must have a license for Fast Path Online Image Copy/EP to use this command.

Available keywords

- DBD
- IAREA
- LARGEST_DATABASE_RECORDS
- POINTER_VALIDATION
- RAP_VALIDATION
- SDEP_VALIDATION

Syntax

IMAGECOPY [keyword=parameter[,...]]
IC [keyword=parameter[,...]]

Parameters

None

Default

None

Alias

None
INCLUDE

Purpose

Use the INCLUDE subcommand to specify segment selection criteria for processing this segment and its dependents by the associated command.

Use

INCLUDE is an optional subcommand for the following commands:

- CHANGE
- EXTRACT
- RELOAD
- UNLOAD

Available keywords

- SAMPLE_INTERVAL
- SAMPLE_LIMIT
- SEGMENT (required)
- WHERE

Syntax

```
command [keyword=parameter[,....]]  
   INCLUDE SEGMENT=parameter,[keyword=parameter[,....]]
```

Parameters

None

Default

If the INCLUDE subcommand is not specified, all segments are selected.

Alias

None
INDEX

Purpose

Use the INDEX keyword to identify the database definition (DBD) of the index to be processed. The DBD must be registered to the source database that is specified on the primary command. Also, the DBD name must correspond to a member name in the IMSACB and PFXLIB libraries.

Use

INDEX is an optional keyword for the IX subcommand.

Syntax

INDEX=parameter

Parameters

Specify a 1-character to 8-character index name.

Default

If the INDEX keyword is omitted, all of the indexes associated with the source database are processed.

Alias

None

INDEX_THREADS

Purpose

Use the INDEX_THREADS keyword to specify the maximum number of index threads that are to be used by Fast Path Indexer/EP during offline index maintenance.
Use

INDEX_THREADS is an optional keyword for following commands:

- BUILD
- CHANGE
- RELOAD
- VERIFY

Related subcommand

IX

NOTE

The INDEX_THREADS keyword is an optional keyword for the CHANGE or RELOAD commands only when used with the IX subcommand.

Related keyword

INDEX

Syntax

INDEX_THREADS=parameter

Parameters

Specify a value from 1 to 16. This value must be less than the number of indexes defined.

Default

None

Alias

XTHREADS
INITIALIZE

Purpose

Use the INITIALIZE command to format (initialize) in offline mode a VSAM cluster that will be used for a DEDB. This command is to perform a fast initialization of the new area data set, which must be empty and have space allocated for it.

Use

INITIALIZE is an optional command for the PFPSYSIN DD statement. You must have a license for Fast Path Online Reorg/EP or Fast Path Reorg/EP to use this command.

Available keywords

- DBD (required)
- IAREA
- INPUT_DSN_MASK

Syntax

INITIALIZE [keyword=parameter[, ...]]

Parameters

None

Default

None

Alias

None

INPUT_DSN_MASK

Purpose

Use the INPUT_DSN_MASK keyword to specify a data set name or mask for use when dynamically allocating *pre-existing* input data sets.
To allocate the VSAM area data set or image copy data set to be used as input, use INPUT_DSN_MASK with one of the following commands:

- ANALYZE
- BUILD
- CHANGE
- DMAC_PRINT
- EXTRACT
- INITIALIZE
- PFPSORT
- PROCESS_AREA
- REORGANIZE
- UNLOAD
- VERIFY

To allocate the input (UNLOAD) data set, use INPUT_DSN_MASK with the RELOAD or PFPSORT command. You can refer to a relative generation of a GDG by including it in the mask, such as in ‘gdg-name(-2)’. The relative generation number that you specify must be 0 or less. If you refer to a GDG name without specifying a relative generation number, the product automatically appends a relative generation of 0.

**Use**

INPUT_DSN_MASK is an optional keyword for the following commands and subcommand:

- ANALYZE
- BUILD
- CHANGE
- DMAC_PRINT
- EXTEND
- EXTRACT
- INITIALIZE
- PFPSORT
- PROCESS_AREA
- RELOAD
- REORGANIZE
- UNLOAD
- VERIFY
- XSCAN
**NOTE**
This keyword applies to processing in offline mode only. It is ignored during online and BMP processing.

**Syntax**

```
INPUT_DSN_MASK='parameter'
INPUT_DSN_MASK='LATEST_BATCH_IMAGECOPY'
```

**Parameters**

Specify a data set name. If the data set name contains special characters, enclose the name in single quotes.

To use the most recent image copy data set that is available in DBRC, specify LATEST_BATCH_IMAGECOPY. The following requirements must be met to use this parameter:

- DBRC must be active.
- The image copy data set must be batch and standard (use the sequential access method). (Concurrent image copies are ignored.)
- The image copy must be marked as valid in DBRC.

**Masks for data set names**

Use the following variables, as necessary, to create the mask.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;AREA</td>
<td>use the area name</td>
</tr>
<tr>
<td>&amp;AREA#</td>
<td>specify the area number as 3 digits when the area number is less than or equal to 999; specify the area number as 5 digits when the area number is greater than 999</td>
</tr>
<tr>
<td>&amp;AREA4#</td>
<td>use the 4-digit area number</td>
</tr>
<tr>
<td>&amp;AREA5#</td>
<td>specify the area number as 5 digits</td>
</tr>
<tr>
<td>&amp;DATE</td>
<td>current date (&quot;Dyyddd&quot;) in its default form (local or UTC)</td>
</tr>
<tr>
<td>&amp;DBD</td>
<td>use the DBD name</td>
</tr>
<tr>
<td>&amp;IMSID</td>
<td>IMS subsystem ID (actual IMS subsystem ID if online; subsystem ID from DFSCV000 if offline)</td>
</tr>
<tr>
<td>&amp;JOBN</td>
<td>job name</td>
</tr>
<tr>
<td>&amp;LCLDATE</td>
<td>current date (&quot;Dyyddd&quot;) in local form</td>
</tr>
<tr>
<td>&amp;LCLTIME</td>
<td>current time (&quot;Thhmmss&quot;) in local form</td>
</tr>
</tbody>
</table>
The following table provides examples of the resulting data set name that is dynamically allocated based on the specified area variable, and the area name or number of digits in the area number:

<table>
<thead>
<tr>
<th>Area name/number</th>
<th>Specified area name variable</th>
<th>Dynamically allocated data set name</th>
</tr>
</thead>
<tbody>
<tr>
<td>A123</td>
<td><code>INPUT_DSN_MASK='PFPTEST.DB.&amp;AREA'</code></td>
<td><code>PFP.TEST.DB.A123</code></td>
</tr>
<tr>
<td>26</td>
<td><code>INPUT_DSN_MASK='PFPTEST.DB.PF&amp;AREA#'</code></td>
<td><code>PFP.TEST.DB.PF026</code></td>
</tr>
<tr>
<td>26</td>
<td><code>INPUT_DSN_MASK='PFPTEST.DB.PF&amp;AREA4#'</code></td>
<td><code>PFP.TEST.DB.PF0026</code></td>
</tr>
<tr>
<td>26</td>
<td><code>INPUT_DSN_MASK='PFPTEST.DB.PF&amp;AREA5#'</code></td>
<td><code>PFP.TEST.DB.PF00026</code></td>
</tr>
<tr>
<td>1024</td>
<td><code>INPUT_DSN_MASK='PFPTEST.DB.PF&amp;AREA#'</code></td>
<td><code>PFP.TEST.DB.PF1024</code></td>
</tr>
<tr>
<td>1024</td>
<td><code>INPUT_DSN_MASK='PFPTEST.DB.PF&amp;AREA4#'</code></td>
<td><code>PFP.TEST.DB.PF01024</code></td>
</tr>
<tr>
<td>1024</td>
<td><code>INPUT_DSN_MASK='PFPTEST.DB.PF&amp;AREA5#'</code></td>
<td><code>PFP.TEST.DB.PF01024</code></td>
</tr>
</tbody>
</table>

Using the `&AREA#4` or `&AREA5#` variable instead of the `&AREA#` variable lets you standardize the length of dynamically allocated data set names when using 1000 or more areas, while still supporting area numbers 1 through 999.

**Default**

None

**Alias**

IDM

## INPUT_THREADS

**Purpose**

Use the INPUT_THREADS keyword to specify the maximum number of input areas to be processed concurrently.
Use

INPUT_THREADS is an optional keyword for the following commands:

- BUILD
- CHANGE
- RELOAD
- UNLOAD
- VERIFY

**NOTE**
The INPUT_THREADS keyword applies to processing in offline mode only. It is ignored during online and BMP processing.

Related keyword

OUTPUT_THREADS

Syntax

INPUT_THREADS=\textit{parameter}

Parameters

Specify a value from 1 to 2048. Specify a value less than or equal to the number of input areas.

**NOTE**
When specifying a value, keep in mind that the more INPUT_THREADS you specify, the more resources you will use.

Default

The default value is determined by the product based on the number of input areas, CPU processors, and other system resources.

Alias

ITHREADS
**INSERT_LIMIT_COUNT**

**Purpose**

Use the INSERT_LIMIT_COUNT keyword to specify the number of segment occurrences for each parent segment to be placed in the normal storage location before attempting to place segments in the storage location that is specified by the LOCATION keyword.

**Use**

INSERT_LIMIT_COUNT is an optional keyword for the LOADCTL subcommand.

**Syntax**

```
INSERT_LIMIT_COUNT=parameter
```

**Parameters**

Specify a value from 0 to 32766.

**Default**

INSERT_LIMIT_COUNT=0 (All segments are placed in the storage location that is specified by the LOCATION keyword.)

**Alias**

ILC

**IOVF_FREESPACE_PERCENT**

**Purpose**

Use the IOVF_FREESPACE_PERCENT keyword to specify a threshold setting when analyzing or monitoring DEDB activity. The percentage of free space within IOVF blocks is computed and compared with the user-specified percentage threshold setting. If the percentage is less than the user-specified parameter, a warning message is issued.
**NOTE**
If you do not specify the IOVF_FREESPACE_PERCENT keyword, this threshold test is not performed.

**Use**

IOVF_FREESPACE_PERCENT is an optional keyword for the THRESHOLD subcommand.

**Syntax**

IOVF_FREESPACE_PERCENT=parameter

**Parameters**

Specify a value from 0 to 100.

**Default**

None

**Alias**

IFP

**IOVF_LOAD_HWM**

**Purpose**

Use the IOVF_LOAD_HWM keyword to specify the percentage of each IOVF block that is utilized by the command process before it attempts to place segments in other storage locations. If the user requests a value less than 100 and the IOVF is not large enough to store all data at the requested fill level, Fast Path Reorg/EP will make a second pass to load data until all data is stored or the IOVF portion is full. An error message will be issued when IOVF becomes full.

**Use**

IOVF_HOLD_HWM is an optional keyword for the following commands:

- CHANGE
- RELOAD
IOVF_SAVE_THRESHOLD

Syntax

IOVF_LOAD_HWM=parameter

Parameters

Specify a value from 50 to 100.

Default

100

Alias

None

IOVF_SAVE_THRESHOLD

Purpose

Use the IOVF_SAVE_THRESHOLD keyword for setting Intelligent Reorg reorganization criteria. After reorganization is performed in main storage, Intelligent Reorg determines the amount of IOVF that has been saved by the reorganization. The IOVF_SAVE_THRESHOLD keyword specifies the minimum amount or percentage of IOVF that must be saved. If sufficient IOVF is not saved, the reorganized UOW is not rewritten to DASD.

Use

IOVF_SAVE_THRESHOLD is an optional keyword for the REORGANIZE command.

Syntax

IOVF_SAVE_THRESHOLD=count
IOVF_SAVE_THRESHOLD=percentage%

Parameters

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>Specify a value from 1 to 32767. If a number is specified without the percent sign, the number represents the minimum number of IOVF CIs to be saved for a UOW before that UOW will be reorganized.</td>
</tr>
<tr>
<td>percentage</td>
<td>Specify a value from 1 to 100 followed by the percent sign (%). If a number is specified with the percent sign, the number represents the percentage of IOVF CIs to be saved for a UOW before that UOW will be reorganized.</td>
</tr>
</tbody>
</table>

Default

IOVF_SAVE_THRESHOLD=1

Alias

ISAVE

IOVF_SPACE_ANALYSIS

Purpose

Use the IOVF_SPACE_ANALYSIS keyword to request the IOVF Space Analysis report. This report shows the percentage of IOVF blocks used for each IOVF space map.

Use

IOVF_SPACE_ANALYSIS is an optional keyword for the REPORT subcommand.

Related keyword

REPORT_DEFAULT

Syntax

IOVF_SPACE_ANALYSIS=parameter
Parameters

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not produce the report.</td>
</tr>
<tr>
<td>YES</td>
<td>Produce the report.</td>
</tr>
</tbody>
</table>

Syntax

The default parameter is set by the REPORT_DEFAULT keyword.

Alias

ISA

**IOVF_USED_PERCENT**

Purpose

Use the IOVF_USED_PERCENT keyword to specify a threshold setting when analyzing or monitoring DEDB activity. The percentage of IOVF blocks used is computed and compared with the user-specified percentage threshold setting. If the computed percentage is greater than the user-specified setting, a warning message is issued.

**NOTE**

If you do not specify the IOVF_USED_PERCENT keyword, this threshold test is not performed.

Use

IOVF_USED_PERCENT is an optional keyword for the THRESHOLD subcommand.

Syntax

```
IOVF_USED_PERCENT=parameter
```

Parameters

Specify a value from 0 to 100.
**IX**

**Default**

None

**Alias**

IUP

**IX**

**Purpose**

Use the IX subcommand to specify the index or indexes to be processed by the primary command.

**Use**

IX is a required subcommand for the following commands:

- BUILD
- VERIFY
- RESYNC
- XSCAN

IX is an optional subcommand for the following commands:

- CHANGE
- INITIALIZE
- RELOAD

**Available keywords**

- DDNAME
- DSNAME
- INDEX
- INPUT_DSN_MASK
- OUTPUT_DSN_MASK
- SORT
- SORT_OPTION
Syntax

```plaintext
command [keyword=parameter[,...]]
IX [keyword=parameter[,...]]
```

Parameters

None

Default

None

Alias

None

LANGUAGE

Purpose

Use the LANGUAGE keyword to specify the national language to be used for messages and reports.

Use

LANGUAGE is an optional keyword for the OPTIONS command.

Syntax

```plaintext
LANGUAGE=parameter
```

Parameters

US – United States English

Default

LANGUAGE=US

Alias

None
LARGEST_DATABASE_RECORDS

Purpose

Use the LARGEST_DATABASE_RECORDS keyword to specify a value different from the default for the number of the largest database records to be tracked by the analysis process, and to be reported on the Record Length Analysis Report. When you specify the LARGEST_DATABASE_RECORDS keyword under a valid primary command, the analysis process must be active by specifying POINTER_VALIDATION=FULL. If POINTER_VALIDATION=FULL is not specified, the LARGEST_DATABASE_RECORDS keyword will have no effect.

If you have activated the Fast Path Analyzer/EP statistics repository feature, the information tracked by the LARGEST_DATABASE_RECORDS keyword will be stored in the Fast Path Analyzer/EP statistics repository facility. You can retrieve collected information on the specified number of largest database records by requesting the Record Length Analysis Report using any of the available retrieval methods. For more information, see the Fast Path Offline Suite User Guide.

If you specify a value of 0 on the LARGEST_DATABASE_RECORDS keyword, no information will be tracked for the largest database records. Consequently, no information will be included on the Record Length Analysis Report for the largest database records in the database or area.

Use

LARGEST_DATABASE_RECORDS is an optional keyword for the following commands:

- ANALYZE
- CHANGE
- EXTEND
- GLOBAL
- IMAGECOPY
- PREPARE
- RELOAD
- REORGANIZE

Related keywords

- POINTER_VALIDATION
- RECORD_LENGTH_ANALYSIS

Syntax

LARGEST_DATABASE_RECORDS=parameter
LIKE

Parameters

Specify a value from 0 to 37167.

Default

LARGEST_DATABASE_RECORDS=10

Aliases

■ LDR
■ KEYS

LIKE

Purpose

Use the LIKE keyword to specify the SMS allocation attributes of an output data set created by dynamic allocation. The model data set specified must be an existing SMS data set.

Use

LIKE is an optional keyword for the following subcommands:

■ ACTIVITY_FILECTL
■ DISCARD_FILECTL
■ IC
■ OFILECTL
■ PLAN_FILECTL

Related keyword

DSNAME

Syntax

LIKE='parameter'

Parameters

Specify a 1-character to 64-character data set name or data set mask. Enclose the mask in single or double quotes. Use the following variables, as necessary, to create the mask:
If you specify DSNAME='PFP.TEST.DB.&AREA' and the area name is A123, the resulting data set name will be PFP.TEST.DB.A123. If you specify DSNAME='PFP.TEST.DB.PF&AREA#' and the area number is 026, the resulting data set will be PFP.TEST.DB.PF026. This data set will have the same SMS attributes as the data set specified on the LIKE keyword. If you specify DSNAME='PFP.TEST.DB.PF&AREA5#' and the area number is greater than 999 (1024 for example), a 5-digit area number will be substituted for the &AREA5# variable. The resulting data set name will be PFP.TEST.DB.PF01024.

Using the &AREA4# or &AREA5# variable in place of the &AREA# variable allows you to standardize the length of the generated data set names when using 1000 or more areas while still supporting area numbers 1 through 999.

**Default**

None

**Alias**

None
**LINE_COUNT**

**Purpose**

Use the LINE_COUNT keyword to specify the number of lines on a report page.

**Use**

LINE_COUNT is an optional keyword for the OPTIONS command.

**Syntax**

```
LINE_COUNT=parameter
```

**Parameters**

Specify a value from 16 to 32767.

**Default**

LINE_COUNT=60

**Alias**

None

---

**LIST**

**Purpose**

Use the LIST subcommand to list objects from the repository catalog data set being processed by the PFPEPR00 utility.

**Use**

LIST is an optional subcommand for the PROCESS_EPR command.

**Available keywords**

- SELECT_AREA
- SELECT_DATE
- SELECT_DBD
- SELECT_GROUP
Syntax

`PROCESS_EPR REPOSITORY_DSNAME=parameter
    LIST object-type,[keyword=parameter[,...]]`

Parameters

Specify one of the following values for the `object-type` parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOBAL_ALLOCATION</td>
<td>List the global level allocation rule.</td>
</tr>
<tr>
<td>GROUP_ALLOCATION</td>
<td>List all group level allocation rules that match the criteria specified using the SELECT_GROUP keyword.</td>
</tr>
<tr>
<td>DBD_ALLOCATION</td>
<td>List all DBD level allocation rules that match the criteria specified using the SELECT_GROUP and SELECT_DBD keywords.</td>
</tr>
<tr>
<td>AREA_ALLOCATION</td>
<td>List all area level allocation rules that match the criteria specified using the SELECT_GROUP, SELECT_DBD and SELECT_AREA keywords.</td>
</tr>
<tr>
<td>ALLOCATION</td>
<td>List all allocation rules (regardless of type) that match the selection criteria specified using the SELECT_GROUP, SELECT_DBD and SELECT_AREA keywords.</td>
</tr>
<tr>
<td>STATISTICS</td>
<td>List all statistics catalog entries that match the selection criteria specified using the SELECT_GROUP, SELECT_DBD, SELECT_AREA, and SELECT_DATE keywords.</td>
</tr>
<tr>
<td>MESSAGE_OVERRIDE</td>
<td>List all message customizations that are stored in the repository catalog</td>
</tr>
</tbody>
</table>

Default

None

Alias

None

**LIST_OPTIONS**

Purpose

Use the LIST_OPTIONS keyword to indicate whether the global default parameters (those specified for the keywords on the OPTIONS command or those defaulted by the product if no OPTIONS command is included) are printed in the data set that is referenced by the PFPPRINT DD statement when the product initializes.
Use

LIST_OPTIONS is an optional keyword for the OPTIONS command.

Syntax

LIST_OPTIONS=parameter

Parameters

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>Print the global default parameters.</td>
</tr>
<tr>
<td>NO</td>
<td>Do not print the global default parameters.</td>
</tr>
</tbody>
</table>

Default

LIST_OPTIONS=YES

Alias

None

LOADCTL

Purpose

Use the LOADCTL subcommand to specify segment placement criteria during reorganization. If a UOW is selected for reorganization, the LOADCTL subcommand lets you set the criteria for placing segments outside of root addressable storage.

Use

LOADCTL is an optional subcommand for the following commands:

- RELOAD
- REORGANIZE
- CHANGE
LOCATION

Available keywords
- INSERT_LIMIT_COUNT
- LOCATION
- SEGMENT (required)
- WHERE

Syntax

```
command [keyword=parameter[,...]]
LOADCTL [keyword=parameter[,...]]
```

Parameters

None

Default

None

Alias

LCT

**LOCATION**

**Purpose**

Use the LOCATION keyword to specify where segments should be placed according to the parameters established with the SEGMENT, INSERT_LIMIT_COUNT, and WHERE keywords.

**Use**

LOCATION is an optional keyword for the LOADCTL subcommand.

**Syntax**

```
LOCATION=parameter
```

**Parameters**

Specify one of the following parameter values:
MESSAGE_LEVEL

**Value** | **Description**
--- | ---
IOVF | Place segments in IOVF storage after the number of segment occurrences for each parent segment placed in root addressable and DOVF storage for a UOW exceeds the user-specified limit. Use the INSERT_LIMIT_COUNT keyword to specify the limit.

DOVF | Place segments in DOVF storage after the number of segments occurrences for each parent segment placed in root addressable storage for a UOW exceeds the user-specified limit. When DOVF storage is full, IOVF will be used. Use the INSERT_LIMIT_COUNT keyword to specify the limit.

---

**NOTE**

When all IOVF storage is full, a normal out-of-space condition occurs.

Default

LOCATION=IOVF

Alias

None

**MESSAGE_LEVEL**

**Purpose**

Use the MESSAGE_LEVEL keyword to change the default severity level (suffix code) of an eligible message.

**Use**

MESSAGE_LEVEL is an optional keyword for the OVERRIDE subcommand.

**Related keyword**

MESSAGE_NUMBER

**Syntax**

MESSAGE_LEVEL=parameter
Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INFORMATIONAL</td>
<td>Generate an informational message; the severity level is 0 (zero). The suffix code will be changed to ‘I’.</td>
</tr>
<tr>
<td>WARNING</td>
<td>Generate a warning message; the severity level is 4. The suffix code will be changed to ‘W’.</td>
</tr>
<tr>
<td>ERROR</td>
<td>Generate an error message; the severity level is 8. The suffix code will be changed to ‘E’.</td>
</tr>
<tr>
<td>CRITICAL</td>
<td>Generate a critical message; the severity level is 12. The suffix code will be changed to ‘C’.</td>
</tr>
</tbody>
</table>

Default

None

Aliases

- MSGLVL
- LEVEL

MESSAGE_LIMIT

Purpose

Use the MESSAGE_LIMIT keyword to suppress the issuance of eligible repetitious messages detected by Fast Path/EP primary command functions. The keyword specifies the maximum number of occurrences of a repetitious message to be issued before suppression. After the maximum has been reached, additional occurrences of that message are not issued, but are listed in a summary message.

Use

MESSAGE_LIMIT is an optional keyword for the OVERRIDE subcommand.

Related keyword

MESSAGE_NUMBER

Syntax

MESSAGE_LIMIT=number
MESSAGE_NUMBER

Parameters

Specify a value from 1 to 32767 to specify the number of occurrences of a repetitious message that are issued prior to suppression. Specify the value NONE to specify that repetitious messages are not to be suppressed.

Default

MESSAGE_LIMIT=100

Aliases

- MSGLIM
- LIMIT

MESSAGE_NUMBER

Purpose

Use the MESSAGE_NUMBER keyword to specify the ID number of an eligible message (numeric portion only) for which you want to perform customization (either change its default severity level (suffix code), specify a suppression threshold, or restore a previously specified override to the product default).

Use

MESSAGE_NUMBER is a required keyword for the following subcommands:

- OVERRIDE
- RESET

Related keywords

- MESSAGE_LEVEL
- MESSAGE_LIMIT

Syntax

MESSAGE_NUMBER=number

Parameters

Specify the 6-digit message number, such as 111162. Do not specify the BMC prefix or the severity level (suffix code). For the RESET subcommand only, you can also specify ALL.
MESSAGE_OVERRIDE

Purpose

Use the MESSAGE_OVERRIDE keyword to produce a report that lists all message customizations that were stored in the repository catalog by a previous execution of the PFPEPR00 utility.

Use

MESSAGE_OVERRIDE is an optional keyword for the LIST subcommand.

Syntax

LIST MESSAGE_OVERRIDE

Parameters

None

Default

None

Alias

None
MESSAGE_SUPPRESSION

Purpose

Use the MESSAGE_SUPPRESSION keyword to suppress the issuance of repetitious messages detected by the primary command function. The keyword specifies the maximum number of occurrences of a repetitious message to be issued before suppression. After the maximum has been reached, additional occurrences of that message are not issued individually. A summary message at the end of the process lists by message number the total suppressions for each suppressed message. Another message at the end of the process lists the total of all messages (both issued messages and suppressed messages).

Use

MESSAGE_SUPPRESSION is an optional keyword for the following commands:

- ANALYZE
- BUILD
- CHANGE
- DMAC_CLEANUP
- DMAC_PRINT
- EXTEND
- GLOBAL
- IMAGECOPY
- INITIALIZE
- PREPARE
- RELOAD
- REORGANIZE
- RESTRUCTURE
- RESYNC
- UNLOAD
- VERIFY

Syntax

MESSAGE_SUPPRESSION=(INFORMATIONAL=parameter,
WARNING=parameter,ERROR=parameter,CRITICAL=parameter)

Parameters

Specify one or more of the parameters as follows:
Default

MESSAGE_SUPPRESSION=(INFORMATIONAL=100, WARNING=100, ERROR=100, CRITICAL=100)

Alias

MSGSUP

**MGMTCLAS**

**Purpose**

Use the MGMTCLAS keyword to specify the SMS management class of an output data set created by dynamic allocation.

**Use**

MGMTCLAS is an optional keyword for the following subcommands:

- ACTIVITY_FILECTL
- ADD
- ALLOCATE
- DISCARD_FILECTL
- IC
- MODIFY
- OFILECTL
- PLAN_FILECTL
Related keyword

DSNAME

Syntax

MGMTCLAS=parameter

Parameters

Specify a 1-character to 8-character management class name.

Default

None

Alias

None

**MGMTCLAS2**

Purpose

Use the MGMTCLAS2 keyword to specify the SMS management class of a secondary shadow output data set created by dynamic allocation.

Use

MGMTCLAS2 is an optional keyword for the ALLOCATE subcommand.

Related keyword

- SHADOW2_DSNNAME
- SHADOW2_SUFFIX

Syntax

MGMTCLAS2=parameter

Parameters

Specify a 1-character to 8-character management class name.
**MODEL_DDNAME**

**Purpose**

Use the MODEL_DDNAME keyword to specify the ddname(s) of a DD statement(s) specified in the JCL to which information is written for subsequent processing by the Area Change Modeling Utility. If the DD statement refers to an existing data set, the new records are appended to the data set (as if you specified DISP=MOD). If the JCL does not contain a DD statement(s) with the referenced ddname, an error message is issued.

The data set referenced by this ddname(s) must be a physical sequential data set. It cannot be a member of a partitioned data set. The DCB characteristics are determined by the analysis function and should not be specified in the JCL. Space requirements vary depending on the number of segments in the area; approximately 24 bytes per segment are required.

**Use**

MODEL_DDNAME is an optional keyword for the ANALYZE command.

**Related keyword**

POINTER_VALIDATION

---

**NOTE**

If you specify the MODEL_DDNAME keyword, you must also specify POINTER_VALIDATION=FULL. If critical errors are detected during analysis, modeling records are not generated.

**Syntax**

MODEL_DDNAME=parameter
MODEL_DDNAME=(parameter[,parameter[, ...]])
Parameters

Specify a 1-character to 8-character ddname.

Default

None

Alias

MDLDD

MODIFY

Purpose

Use the MODIFY subcommand to update existing objects in the repository catalog data set being processed by the PFPEPR00 utility.

Use

MODIFY is an optional subcommand for the PROCESS_EPR command.

Available keywords

- DATACLAS
- DETAIL_LEVEL
- DSNNAME
- EXPDT
- MGMTCLAS
- RETPD
- SELECT_AREA
- SELECT,DBD
- SELECT,GROUP
- STORCLAS
- UNIT
- VOLSER

Syntax

PROCESS_EPR REPOSITORY_DSNNAME=parameter
MODIFY object-type,[keyword=parameter[,...]]
Parameters

Specify one of the following values for the `object-type` parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOBAL_ALLOCATION</td>
<td>Modify the global level allocation rule.</td>
</tr>
<tr>
<td>GROUP_ALLOCATION</td>
<td>Modify all group level allocation rules that match the criteria specified using the SELECT_GROUP keyword.</td>
</tr>
<tr>
<td>DBD_ALLOCATION</td>
<td>Modify all DBD level allocation rules that match the criteria specified using the SELECT_GROUP and SELECT_DBD keywords.</td>
</tr>
<tr>
<td>AREA_ALLOCATION</td>
<td>Modify all area level allocation rules that match the criteria specified using the SELECT_GROUP, SELECT_DBD and SELECT_AREA keywords.</td>
</tr>
<tr>
<td>ALLOCATION</td>
<td>Modify all allocation rules (regardless of type) that match the selection criteria specified using the SELECT_GROUP, SELECT_DBD and SELECT_AREA keywords.</td>
</tr>
</tbody>
</table>

Default

None

Alias

None

**MONITOR**

Purpose

Use the MONITOR keyword to periodically initiate the operator console DISPLAY command. Product resource usage will be reported at the minute frequency specified by the MONITOR keyword parameter value.

Use

MONITOR is an optional keyword for the OPTIONS command.

Syntax

```
MONITOR=parameter
```
NOTIFY

Parameters

Specify a value from 1 to 1440.

Default

None

Alias

None

NOTIFY

Purpose

Use the NOTIFY keyword to specify whether the image copy data set created with the IC subcommand is to be recorded within DBRC. The keyword is meaningful only if DBRC is active, the area is registered, and the image copy data set is successfully created without error. No more than two image copy data sets can be recorded within DBRC.

Use

NOTIFY is an optional keyword for the IC subcommand.

Syntax

NOTIFY=parameter

Parameters

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>Record this image copy data set within DBRC.</td>
</tr>
<tr>
<td>NO</td>
<td>Do not record this image copy data set.</td>
</tr>
</tbody>
</table>

Default

NOTIFY=YES
**OAREA**

**Purpose**

Use the OAREA keyword to specify one or more output areas, a range of output areas, or a combination of output areas and output area ranges to the process.

**Use**

OAREA is an optional keyword for the following commands and subcommands:

- ALLOCATE
- CHANGE
- OFILECTL
- PFPSORT
- REGISTER
- RELOAD
- UNLOAD

**Related keyword**

IAREA

**Syntax**

OAREA=ALL or OAREA=*  
OAREA=parameter  
OAREA='parameter'  
OAREA=(parameter1, parameter2, ..., parameterN)  
OAREA=(RANGE=(parameter1, parameterN))  
OAREA=(RANGE=(parameter1, parameter2), RANGE=(parameter3, parameter4))  
OAREA=(RANGE=(*, parameterN))  
OAREA=(RANGE=(parameter1, *))  
OAREA=(area-list)
OAREA

**NOTE**

OAREA='\texttt{parameter}' includes wildcard masks. For example, the syntax for areas DPAS7A0, DPAS7B0, and DPA7C0 can be specified as follows:

- OAREA='DP????C0'
- OAREA='DPAS7*'
- OAREA='DPA7???'
- OAREA='DPAS7??'

**Parameters**

Specify either OAREA=ALL or OAREA=*, or any combination of the following parameter values:

**NOTE**

OAREA=ALL and OAREA=* can be used to specify all areas of the DEDB.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>areaname</td>
<td>Specify one or more areas by using the 1-character to 8-character area name for each area specified.</td>
</tr>
<tr>
<td>areaname mask</td>
<td>You can also specify one or more areas by using the 1-character to 8-character area name mask for each area specified. Wildcard masks</td>
</tr>
<tr>
<td></td>
<td>using the asterisk (*) and question mark (?) can be included. When using wildcard masks, the value of the keyword must be enclosed in</td>
</tr>
<tr>
<td></td>
<td>quotation marks. The asterisk wildcard matches all characters to the right of the asterisk. The question mark wildcard matches the</td>
</tr>
<tr>
<td></td>
<td>characters in the string in the place the question mark holds.</td>
</tr>
<tr>
<td>areanumber</td>
<td>Specify one or more areas by using the 1-character to 5-character area number for each area specified.</td>
</tr>
<tr>
<td>area-list</td>
<td>Specify a list of areas.</td>
</tr>
<tr>
<td>RANGE=(startarea,endarea)</td>
<td>Use this syntax to specify a consecutive range of areas using either areaname or areanumber parameters. The area number associated with</td>
</tr>
<tr>
<td>RANGE=(*,endarea)</td>
<td>startarea must be less than (and not equal to) the area number associated with endarea.</td>
</tr>
<tr>
<td>RANGE=(startarea,*)</td>
<td>When the asterisk is used with the RANGE keyword, it can be used to specify the beginning or ending range for specific areas of the DEDB.</td>
</tr>
</tbody>
</table>

**Note:** Wildcard masks cannot be used with the RANGE keyword.
Default

OAREA=ALL

Alias

None

OBJECT

Purpose

Use the OBJECT keyword to specify the types of resource statistics to display.

Use

OBJECT is an optional keyword for the DISPLAY command.

Related keyword

ID

Syntax

OBJECT=parameter

Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB</td>
<td>Generate a summary of statistics for the executing job</td>
</tr>
<tr>
<td>PRODUCT</td>
<td>Generate a summary of statistics for the executing products, including the</td>
</tr>
<tr>
<td></td>
<td>task ID</td>
</tr>
<tr>
<td>VSMSUMM</td>
<td>Generate a summary of storage usage. This summary is at a more detailed</td>
</tr>
<tr>
<td></td>
<td>level than would be provided by the JOB parameter.</td>
</tr>
<tr>
<td>TASK</td>
<td>Generate a list of tasks and statistics about the tasks. This list includes</td>
</tr>
<tr>
<td></td>
<td>task status (active or waiting) and task level.</td>
</tr>
<tr>
<td>MMGR</td>
<td>Generate a summary of subpool usage from the perspective of the Fast Path/EP</td>
</tr>
<tr>
<td></td>
<td>Series memory manager.</td>
</tr>
<tr>
<td>TRACE</td>
<td>Generate a summary of trace statistics. This summary includes a listing of</td>
</tr>
<tr>
<td></td>
<td>active traces and the number of internal trace tables.</td>
</tr>
</tbody>
</table>
OCACHE

Default

OBJECT=JOB

Alias

None

OCACHE

Purpose

Use the OCACHE keyword to specify the types of control intervals that are held in storage until all processing has completed. All control intervals held in storage in this way are written to DASD when the data set is closed. A dataspace is used to implement the output cache.

Use

OCACHE is an optional keyword for the following commands:

- CHANGE
- RELOAD

Related keyword

ICACHE

Syntax

OCACHE=parameter
OCACHE=(parameter1, parameter2)

Parameters

Specify one or more of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOVF</td>
<td>Keep IOVF and space map control intervals in storage.</td>
</tr>
<tr>
<td>SDEP</td>
<td>Keep SDEP control intervals in storage.</td>
</tr>
</tbody>
</table>
OFILECTL

Purpose

Use the OFILECTL subcommand to define the output files to be created during command processing.

Use

OFILECTL is an optional subcommand for the following commands:

- EXTRACT
- UNLOAD

Available keywords

- AVGREC
- DATACLAS
- DDNAME
- DISP
- DSNAME
- EXPDT
- LIKE
- MGMTCLAS
- OAREA
- RETPD
- SPACE
- STORCLAS
- UNIT
- VOLCNT
- VOLSER
OPTIONS

Syntax

```
command [keyword=parameter[,....]]
OFILECTL=[keyword=parameter[,....]]
```

Parameters

None

Default

None

Alias

FCT

OPTIONS

Purpose

Use the OPTIONS command to provide default parameters for the Fast Path/EP Series products. Use its subcommands (OVERRIDE and RESET) to set or revert temporary overrides of product message defaults.

NOTE

You can specify OPTIONS command settings in the PFPOPTS DD statement or the OPTIONS module from the PFPDFLTS DD statement. Use the DISPLAY command to view the OPTIONS settings.

If both PFPDFLTS and PFPOPTS DD statements are specified in the JCL, then a keyword specified in the PFPOPTS DD will override the same keyword specified in the OPTIONS module for the PFPDFLTS DD.

Use

OPTIONS is an optional command for the PFPOPTS DD statement or the OPTIONS module from the PFPDFLTS DD statement.
Available keywords
- CASE
- DATE_TIME_FORMAT
- DESC
- IFP_ACCOUNT
- IFP_JOBNAME
- IFP_LIMIT
- IFP_PROCNAME
- LANGUAGE
- LINE_COUNT
- LIST_OPTIONS
- MONITOR
- PRODUCT_LIMIT
- REPOSITORY_DSNAME
- REPOSITORY_GROUP
- REPOSITORY_OVERWRITE
- REPOSITORY_RETENTION_COUNT
- REPOSITORY_RETENTION_PERIOD
- ROUTCDE
- SORT_NAME
- TIMESTAMP
- WARNING
- WORK_DATASET

Syntax

OPTIONS [keyword=parameter[,...]]

Parameters

None

Default

None

Alias

None
ORPHANED_SDEP_MSG

Purpose

Use the ORPHANED_SDEP_MSG keyword to specify the error message severity level to be set when an orphaned SDEP segment is detected. Individual segments can be identified as orphans only when SDEP_VALIDATION=FULL is specified. This keyword has no effect if SDEP_VALIDATION=QUICK or SDEP_VALIDATION=NONE is specified since SDEP segments are not analyzed.

Use

ORPHANED_SDEP_MSG is an optional keyword for the following commands:

- ANALYZE
- CHANGE
- EXTEND
- GLOBAL
- IMAGECOPY
- PREPARE
- RELOAD
- REORGANIZE

Related keywords

- ACCESS
- SDEP_VALIDATION

Syntax

ORPHANED_SDEP_MSG=parameter

Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR</td>
<td>Generate an error message; severity level is 8.</td>
</tr>
<tr>
<td>INFORMATIONAL</td>
<td>Generate an informational message; severity level is 0.</td>
</tr>
<tr>
<td>NOMSG</td>
<td>Do not generate a message; severity level is not affected.</td>
</tr>
<tr>
<td>WARNING</td>
<td>Generate a warning message; severity level is 4.</td>
</tr>
</tbody>
</table>
Default

ORPHANED_SDEP_MSG=ERROR is the default if the area is being analyzed by using ACCESS=OFFLINE, or if an image copy created in offline mode is being analyzed.

ORPHANED_SDEP_MSG=INFORMATIONAL is the default if one of the following conditions exists:

- The area is being analyzed by using ACCESS=CONCURRENT.
- An image copy created in concurrent mode is being analyzed.
- The area is being analyzed online.

Alias

OSM

OUTAGE_WINDOW

Purpose

Use the OUTAGE_WINDOW keyword to specify a window of time that post-processing can start and end during restructure processing. During this outage time, the affected areas are taken offline to swap the online area data sets and primary shadow data sets.

**NOTE**

Specifying an OUTAGE_WINDOW value during restructure processing overrides any value that was written to the Restructure Plan data set during prepare processing.

If you want to modify the outage window time after the restructure job has started, you can dynamically modify the outage time by using the MVS MODIFY (F) operator command. For more information, see the *Fast Path Online Restructure/EP User Guide*. 
OUTAGE_WINDOW

Use

OUTAGE_WINDOW is a *required* keyword for the SET command.
OUTAGE_WINDOW is an *optional* keyword for the following commands:

- GLOBAL
- PREPARE
- RESTART
- RESTRUCTURE

Syntax

OUTAGE_WINDOW=(start-date-time, end-date-time)
OUTAGE_WINDOW=(start-date-time)
OUTAGE_WINDOW=(, end-date-time)

Parameters

The *date-time* parameters define a window of time.

The following considerations apply to the *date-time* parameters:

- If you omit the *start-date-time*, Fast Path Online Restructure/EP assumes that the window opened at the beginning of the year 0000.
- If you omit the *end-date-time*, Fast Path Online Restructure/EP assumes that the window closes at the end of year 9999.
- The *start-date-time* must be less than the *end-date-time*.
- The *end-date-time* must be greater than the current time.

You can specify the *date-time* parameters in different combinations, as a string of digits separated by decimals:

```
```

The values are interpreted as follows:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cc</td>
<td>defines the century (with 20 as the default)</td>
</tr>
<tr>
<td>yy</td>
<td>defines the year (00 through 99).</td>
</tr>
<tr>
<td>ddd</td>
<td>defines the day of the year (000 through 366, where 366 is valid only for leap years)</td>
</tr>
<tr>
<td>hh</td>
<td>defines the hour by using the 24-hour military format (00 through 23)</td>
</tr>
</tbody>
</table>
The following example illustrates how to specify an outage window that opens at 2:00 A.M. on December 1, 2009, and closes two hours later:

```
OUTAGE_WINDOW=( 2009.335.02 , 2009.335.04 )
```

**Default**

None

**Alias**

None

### OUTPUT

**Purpose**

Use the OUTPUT subcommand to control output selection and to modify the layout or content of selected segment data.

**NOTE**

When SDEP_PROCESS=PHYSICAL has been selected on a CHANGE, UNLOAD, or RELOAD command, you cannot specify an OUTPUT subcommand for the SDEP segment.

**Use**

OUTPUT is an optional subcommand for the following commands:

- CHANGE
- EXTRACT
- PREPARE
- RELOAD
- UNLOAD
Available keywords

- FIELDS
- SEGMENT (required)
- WHERE

**NOTE**

When used with the PREPARE command, the FIELDS keyword is required on each of the OUTPUT subcommands for a segment if you modify the position or length of the symbolic key field; otherwise, it is optional.

Syntax

```
command [keyword=parameter[,...]]
OUTPUT [keyword=parameter[,...]]
```

Parameters

None

Default

None

Alias

None

**OUTPUT_DSN_MASK**

Purpose

Use the OUTPUT_DSN_MASK keyword to specify a data set name or mask for use when dynamically allocating *pre-existing* output data sets.

When used with the CHANGE, or RELOAD command, OUTPUT_DSN_MASK is used to allocate the output area being created.

When used with the IX subcommand, OUTPUT_DSN_MASK is used to allocate the index data set being built.
When used with the EXTRACT, UNLOAD or PFPSORT command, OUTPUT_DSN_MASK is used to allocate the output file. You can refer to a relative generation of a GDG by including it in the mask, such as in ‘gdg-name(-2)’. The relative generation number you specify must be 0 or less. If you refer to a GDG name without specifying a relative generation number, the product will automatically append a relative generation of 0.

Because the data set must pre-exist, using a GDG is cumbersome when the OUTPUT_DSN_MASK keyword is used with EXTRACT or RELOAD. As an alternate method for specifying an output data set with EXTRACT or UNLOAD, you can use the OFILECTL subcommand and its associated DSNAME keyword.

Use

OUTPUT_DSN_MASK is an optional keyword for the following commands and subcommand:

- CHANGE
- EXTRACT
- IX
- PFPSORT
- RELOAD
- UNLOAD

Syntax

OUTPUT_DSN_MASK=’parameter’

Parameters

Specify a data set name. Enclose the data set name in single quotes. Use the following parameters, as necessary, to create the mask:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;AREA</td>
<td>use the area name</td>
</tr>
<tr>
<td>&amp;AREA#</td>
<td>specify the area number as 3 digits when the area number is less than or equal to 999; specify the area number as 5 digits when the area number is greater than 999</td>
</tr>
<tr>
<td>&amp;AREA4#</td>
<td>use the 4-digit area number</td>
</tr>
<tr>
<td>&amp;AREA5#</td>
<td>specify the area number as 5 digits</td>
</tr>
</tbody>
</table>

NOTE

The OUTPUT_DSN_MASK keyword applies to index processing in offline mode only. It is ignored during BMP processing.
The following table provides examples of the resulting data set name that is dynamically allocated based on the specified area variable, and the area name or number of digits in the area number:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;DATE</td>
<td>current date (&quot;Dyyddd&quot;) in its default form (local or UTC)</td>
</tr>
<tr>
<td>&amp;DBD</td>
<td>use the DBD name</td>
</tr>
<tr>
<td>&amp;IMSID</td>
<td>IMS subsystem ID (actual IMS subsystem ID if online; subsystem ID from DFSCV000 if offline)</td>
</tr>
<tr>
<td>&amp;JOBN</td>
<td>job name</td>
</tr>
<tr>
<td>&amp;LCLDATE</td>
<td>current date (&quot;Dyyddd&quot;) in local form</td>
</tr>
<tr>
<td>&amp;LCLTIME</td>
<td>current time (&quot;Thhmmss&quot;) in local form</td>
</tr>
<tr>
<td>&amp;PROCSN</td>
<td>procedure step name</td>
</tr>
<tr>
<td>&amp;STEPN</td>
<td>step name</td>
</tr>
<tr>
<td>&amp;TIME</td>
<td>current time (&quot;Thhmmss&quot;) in its default form (local or UTC)</td>
</tr>
<tr>
<td>&amp;UTCDATE</td>
<td>current date (&quot;Dyyddd&quot;) in UTC form</td>
</tr>
<tr>
<td>&amp;UTCTIME</td>
<td>current time (&quot;Thhmmss&quot;) in UTC form</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Area name/number</th>
<th>Specified area name variable</th>
<th>Dynamically allocated data set name</th>
</tr>
</thead>
<tbody>
<tr>
<td>A123</td>
<td>OUTPUT_DSN_MASK='PFPTEST.DB.&amp;AREA'</td>
<td>PFP.TEST.DB.A123</td>
</tr>
<tr>
<td>26</td>
<td>OUTPUT_DSN_MASK='PFPTEST.DB.PF&amp;AREA#'</td>
<td>PFP.TEST.DB.PF026</td>
</tr>
<tr>
<td>26</td>
<td>OUTPUT_DSN_MASK='PFPTEST.DB.PF&amp;AREA4#'</td>
<td>PFP.TEST.DB.PF0026</td>
</tr>
<tr>
<td>26</td>
<td>OUTPUT_DSN_MASK='PFPTEST.DB.PF&amp;AREA5#'</td>
<td>PFP.TEST.DB.PF00026</td>
</tr>
<tr>
<td>1024</td>
<td>OUTPUT_DSN_MASK='PFPTEST.DB.PF&amp;AREA#'</td>
<td>PFP.TEST.DB.PF01024</td>
</tr>
<tr>
<td>1024</td>
<td>OUTPUT_DSN_MASK='PFPTEST.DB.PF&amp;AREA4#'</td>
<td>PFP.TEST.DB.PF01024</td>
</tr>
<tr>
<td>1024</td>
<td>OUTPUT_DSN_MASK='PFPTEST.DB.PF&amp;AREA5#'</td>
<td>PFP.TEST.DB.PF01024</td>
</tr>
</tbody>
</table>

Using the &AREA#4 or &AREA5# variable instead of the &AREA# variable lets you standardize the length of dynamically allocated data set names when using 1000 or more areas, while still supporting area numbers 1 through 999.

**Default**

None

**Alias**

ODM
OUTPUT_THREADS

Purpose

Use the OUTPUT_THREADS keyword to specify the maximum number of output threads that are to be used by Fast Path Reorg/EP. If you determine that a database change is taking longer than expected, adjusting the number of output threads might alleviate the problem.

Use

OUTPUT_THREADS is an optional keyword for the following commands:

- CHANGE
- RELOAD
- UNLOAD

Related keyword

INPUT_THREADS

Syntax

OUTPUT_THREADS=parameter

Parameters

Specify a value from 1 to 2048. Specify a value less than or equal to the number of output areas.

NOTE

When specifying a value, keep in mind that the more INPUT_THREADS you specify, the more resources you will use.

Default

The default value is determined by the product based on the number of output areas, CPU processors, and other system resources.

Alias

OTHREADS
### OVERRIDE

**Purpose**

Use the OVERRIDE subcommand to permanently suppress the issuance of selected messages after a specified number of occurrences, or to permanently change the severity level of the suffix for selected messages.

The OVERRIDE subcommand is executed under the PFPEPR00 utility to permanently store changes to messages in the Fast Path/EP repository catalog.

The OVERRIDE subcommand can also be specified on the PFPOPTS DD statement to temporarily set changes to messages for the processes specified on the PFPSYSIN DD statement.

**Use**

OVERRIDE is an optional subcommand for the following commands:

- PROCESS_EPR
- OPTIONS

**Available keywords**

- MESSAGE_LEVEL
- MESSAGE_LIMIT
- MESSAGE_NUMBER (required)

**Syntax**

```
command [keyword=parameter[,...]]
OVERRIDE [keyword=parameter[,...]]
```

**Parameters**

None

**Default**

None

**Alias**

None
PERFORM

Purpose
Use the PERFORM subcommand to specify a script to be executed. A script is a complete procedural program written in an internal script language.

Use
PERFORM is an optional subcommand that can be used with the PROCESS_AREA command. You can specify multiple PERFORM subcommands on a single PROCESS_AREA command.

Available keyword
SCRIPT (required)

Syntax
PROCESS_AREA [keyword=parameter[,...]]
[PERFORM [SCRIPT= { script }]]...

Parameters
For detailed information on coding scripts, see Chapter 3, “Expression syntax.”

Default
If no PERFORM subcommand is specified, then the PROCESS_AREA command performs the same function as the DMAC_PRINT command.

Alias
None

PFPSORT

Purpose
Use the PFPSORT command to execute the Fast Path Reorg/EP File Sort Utility. This utility can be used to customize the sorting of the root anchor points (RAPs), root keys, and logical SDEPs defined in the file.
Use

PFPSORT is an optional command for the PFPSYSIN DD statement. You must have a license for Fast Path Online Reorg/EP or Fast Path Reorg/EP to use this command.

Available keywords

- DBD
- INPUT_DSN_MASK
- OAREA
- OUTPUT_DSN_MASK
- SORT_OPTION
- SORT_SEQUENCE

Syntax

PFPSORT [keyword=parameter[,...]]

Parameters

None

Default

None

Alias

None

PLAN_FILECTL

Purpose

Use the PLAN_FILECTL subcommand to control the allocation of the Restructure Plan data set. The Prepare function produces the Restructure Plan data set as an output data set. This data set becomes input for the Shadow Initialization and Restructure functions when restructuring a DEDB online. The Restructure Plan data set is used to

- identify unsupported changes by comparing the IMS control blocks that define the database (to identify what changes are being requested)
- determine minimum processing requirements by determining which area or areas within the original database must be processed
**Use**

PLAN_FILECTL is a required subcommand for the following commands:

- BACKOUT
- PREPARE
- RESTART
- RESTRUCTURE
- SHADOW_INIT
- STATUS

**Available keywords**

- AVGREC
- DDNAME (required if the DSNAME keyword is not specified)
- DISP
- DSNAME (required if the DDNAME keyword is not specified)
- EXPDT
- LIKE
- MGMTCLAS
- RETPD
- SPACE
- STORCLAS
- UNIT
- VOLCNT
- VOLSER

**Syntax**

```
command [keyword=parameter[,...]]
PLAN_FILECTL [keyword=parameter[,...]]
```

**Parameters**

None

**Default**

None

**Alias**

None
POINTER_ANALYSIS

Purpose

Use the POINTER_ANALYSIS keyword to request the Pointer Analysis Report. For each prefix pointer type occurring within each segment, this report provides a count and percentage of null pointers, pointers that point into the same CI, and pointers that point into a different CI.

Use

POINTER_ANALYSIS is an optional keyword for the REPORT subcommand.

Related keyword

REPORT_DEFAULT

Syntax

POINTER_ANALYSIS=parameter

Parameters

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not produce the report.</td>
</tr>
<tr>
<td>YES</td>
<td>Produce the report.</td>
</tr>
</tbody>
</table>

Default

The default parameter is set by the REPORT_DEFAULT keyword.

Aliases

- PA
- PTR
POINTER_VALIDATION

Purpose

Use the POINTER_VALIDATION keyword to select the method to be used to perform logical validation of pointers to the root and direct dependent segments.

Use

POINTER_VALIDATION is an optional keyword for the following commands:

- ANALYZE
- CHANGE
- EXTEND
- GLOBAL
- IMAGECOPY
- PREPARE
- REORGANIZE
- RELOAD

Related keywords

- LARGEST_DATABASE_RECORDS
- SDEP_VALIDATION
- RAP_VALIDATION

Syntax

POINTER_VALIDATION=parameter

Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO</td>
<td>For the Fast Path Online Restructure/EP product, this value is the default and is only valid on the PREPARE command. If pointer analysis is performed, it will only gather statistics (same as POINTER_VALIDATION=OFF).</td>
</tr>
</tbody>
</table>
POINTER_VALIDATION

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FULL</td>
<td>This value performs a complete cross-reference validation of every pointer and segment occurrence. All pointers are checked to ensure that a segment occurrence of the proper type appears at the referenced relative byte address (RBA). The complete chain of dependent segment occurrences associated with their parent segment (the PCF/PTF pointers) is identified. PCL and SSPTR pointers are checked to verify that the segments referenced appear in the chain and that the PCL pointer references the last segment in the chain. FULL validation also ensures that every segment occurrence is referenced by only one PCF/PTF pointer.</td>
</tr>
<tr>
<td>QUICK</td>
<td>This value generates a checksum for each segment type within each 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 only one PCF/PTF pointer to each segment, the resultant checksum is zero if no errors are present. A second checksum value consists of the sum of the RBAs for each segment with a null PTF pointer minus the RBAs of all PCL pointers to the segment type. SSPTR pointers are not validated in QUICK validation mode. Because many of the statistics needed cannot be collected in QUICK validation mode, certain reports are not produced, certain thresholds are not checked, and modeler output is not allowed.</td>
</tr>
<tr>
<td>OFF</td>
<td>This value specifies that pointers will not be validated by either of the methods discussed above. The analysis process performs physical validation of the area and gathers statistics for reporting. Because many of the statistics needed cannot be collected in this mode, certain reports are not produced, certain thresholds are not checked, and modeler output is not allowed.</td>
</tr>
<tr>
<td>NONE</td>
<td>This value is valid only on GLOBAL, EXTEND, REORGANIZE, RELOAD, CHANGE, and IMAGECOPY commands. The analysis function is not executed.</td>
</tr>
</tbody>
</table>

NOTE
You must have a license for Fast Path Analyzer/EP to use the FULL, QUICK, and OFF parameters on the PREPARE command.

Default

POINTER_VALIDATION=QUICK
POINTER_VALIDATION=AUTO (when used with the PREPARE command)

Aliases

- PTR
- FPA
**PREOPEN**

**Purpose**

Use the PREOPEN keyword to specify whether to pre-open an area at either of the following points:

- after the first checkpoint following the next initialization of the control region
- when the next /START AREA command is processed

**Use**

PREOPEN is an optional keyword for the REGISTER subcommand.

**Related keywords**

VSO

**Syntax**

PREOPEN=parameter

**Parameters**

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Indicates that the area should not be pre-opened the next time the control region is started or a /START AREA command is processed.</td>
</tr>
<tr>
<td>YES</td>
<td>Indicates that the area should be opened the next time the control region is started or a /START AREA command is processed.</td>
</tr>
</tbody>
</table>

**Default**

The default value is NO.

**Alias**

None
PREPARE

Purpose

Use the PREPARE command to create a Restructure Plan data set, which can be used for input into the online restructure process. Use the PLAN_FILECTL subcommand with the PREPARE command to specify the name of the output Restructure Plan data set.

The Restructure Plan data set is used to

- identify unsupported changes by providing a detailed comparison of the IMS control blocks that defines the database to identify what changes are being requested.

- determine minimum processing requirements by determining which area or areas within the original database must be processed.

NOTE

You cannot specify multiple PREPARE commands in the same command stream; only one PREPARE command can be specified in a command stream.

PREPARE commands cannot be used in the same command stream with the SHADOW_INIT, RESTRUCTURE, or RESTART commands.

Use

PREPARE is an optional command for the PFPSYSIN DD statement when you execute the PFCMAIN program. You must have a license for Fast Path Online Restructure/EP to use this command.

Available keywords

- ACCESS (required)
- DBD (required)
- HISTORY_DDNAME
- INPUT_DSN_MASK
- LARGEST_DATABASE_RECORDS
- MESSAGE_SUPPRESSION
- ORPHANED_SDEP_MSG
- OUTAGE_WINDOW
- REQUIRE_AREA
- POINTER_VALIDATION
- RAP_VALIDATION
- RETAINED_SUFFIX
PROCESS_AREA

- SDEP_VALIDATION
- SHADOW_SUFFIX
- SHADOW2_DSNAME
- SHADOW2_SUFFIX

Syntax

PREPARE [keyword=parameter[,...]]
   [subcommand[keyword=parameter[,...]]]

Parameters

None

Subcommands

- ACTIVITY_FILECTL
- OUTPUT
- PLAN_FILECTL (required)
- REGISTER
- REPORT
- THRESHOLD

Default

None

Alias

None

PROCESS_AREA

Purpose

Use the PROCESS_AREA command to execute the Control Interval Dump and Modification Utility. The Control Interval Dump and Modification Utility will execute the script or scripts that are specified using one or more PERFORM subcommands.
Use

PROCESS_AREA is an optional command for the PFPSYSIN DD statement. You must have a license for Fast Path Online Analyzer/EP or Fast Path Analyzer/EP to use this command.

Available keywords

- DBD
- IAREA
- INPUT_DSN_MASK

Syntax

PROCESS_AREA [keyword=parameter[,...]]
[PERFORM [SCRIPT={script}]]...

NOTE

When you specify the PROCESS_AREA command without any PERFORM subcommand, it will execute the same function as the DMAC_PRINT command (it will print the contents of the DMAC block for an area).

Parameters

None

Default

None

Alias

None

PROCESS_EPR

Purpose

Use the PROCESS_EPR command to specify the repository catalog data set to be processed by the utility. Use its subcommands (ADD, LIST, MODIFY, DELETE, RETRIEVE, OVERRIDE, and RESET) to specify the actions to be performed by the utility against the selected repository catalog data set.
Use

PROCESS_EPR is an optional command for the PFPSYSIN DD statement when executing the PFPEPR00 utility program.

Available keyword

REPOSITORY_DSNAME (required)

Syntax

PROCESS_EPR REPOSITORY_DSNAME=parameter
   [subcommand [keyword=parameter[,...]] ...]

Parameters

None

Default

None

Alias

None

PRODUCT

Purpose

Use the PRODUCT keyword to specify the products to halt (shut down) when a message is routed to the operator. When used with the ID keyword, the PRODUCT keyword helps to ensure that accidental task shutdown does not occur. The task ID specified with the ID keyword must be a task for the specified product.

Use

PRODUCT is a required keyword for the SHUTDOWN command.

Related keywords

- FORCE
- ID
Syntax

PRODUCT=parameter

Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>Shut down all functions that are listed in this table.</td>
</tr>
<tr>
<td>ANALYZER</td>
<td>Shut down Fast Path Analyzer/EP analysis function.</td>
</tr>
<tr>
<td>BUILD</td>
<td>Shut down Fast Path Indexer/EP index build function.</td>
</tr>
<tr>
<td>CHANGE</td>
<td>Shut down Fast Path Reorg/EP change function.</td>
</tr>
<tr>
<td>DMAC_PRINT</td>
<td>Shut down Fast Path/EP DMAC print function.</td>
</tr>
<tr>
<td>EXTRACT</td>
<td>Shut down Fast Path Analyzer/EP extract function</td>
</tr>
<tr>
<td>IMAGECOPY</td>
<td>Shut down Fast Path Online Image Copy/EP image copy function.</td>
</tr>
<tr>
<td>INITIALIZE</td>
<td>Shut down Fast Path Reorg/EP initialization function.</td>
</tr>
<tr>
<td>PREPARE</td>
<td>Shut down Fast Path Online Restructure/EP Prepare function.</td>
</tr>
<tr>
<td>RELOAD</td>
<td>Shut down Fast Path Reorg/EP reload function.</td>
</tr>
<tr>
<td>REORGANIZE</td>
<td>Shut down Fast Path Reorg/EP reorganization function.</td>
</tr>
<tr>
<td>RESTRUCTURE</td>
<td>Shut down Fast Path Online Restructure/EP Restructure function.</td>
</tr>
<tr>
<td>RESTART</td>
<td>Shut down Fast Path Online Restructure/EP Restart function.</td>
</tr>
<tr>
<td>RESYNC</td>
<td>Shut down Fast Path Indexer/EP index resynchronize function.</td>
</tr>
<tr>
<td>RETRIEVE</td>
<td>Shut down Fast Path Analyzer/EP retrieve function.</td>
</tr>
<tr>
<td>SHADOW_INIT</td>
<td>Shut down Fast Path Online Restructure/EP Shadow Initialization function.</td>
</tr>
<tr>
<td>VERIFY</td>
<td>Shut down Fast Path Indexer/EP index verify function.</td>
</tr>
</tbody>
</table>

**NOTE**

If you specify PRODUCT=ALL, you also must specify ID=1.

Default

None
**PRODUCT_LIMIT**

**Purpose**

Use the PRODUCT_LIMIT keyword to specify the maximum number of product command tasks to be executed concurrently when multiple commands are being processed in the same job. If processors are limited at your site, you can use this keyword to single-thread command execution by specifying PRODUCT_LIMIT=1. To take advantage of a multiple-processor environment, you can use this keyword to specify a larger number to increase command execution parallelism.

**Use**

PRODUCT_LIMIT is an optional keyword for the OPTIONS command.

**Syntax**

PRODUCT_LIMIT=parameter

**Parameters**

Specify a value from 1 to 15.

**Default**

PRODUCT_LIMIT=3

**Alias**

None
Purpose

Use the RAA_FREESPACE_PERCENT keyword to specify a threshold setting when analyzing or monitoring DEDB activity. The percentage of total amount of free space in the RAA base is computed and compared with the user-specified percentage threshold setting. If the computed RAA free space percentage is less than the user-specified setting, a warning message is issued.

**NOTE**

If you do not specify the RAA_FREESPACE_PERCENT keyword, this threshold test is not performed.

Use

RAA_FREESPACE_PERCENT is an optional keyword for the THRESHOLD subcommand.

Syntax

```
RAA_FREESPACE_PERCENT=parameter
```

Parameters

Specify a value from 0 to 100.

Default

None

Alias

RFP
RAPOVERFLOW_PERCENT

Purpose

Use the RAPOVERFLOW_PERCENT keyword to specify a threshold setting when analyzing or monitoring overflow free space statistics. The percentage of RAP blocks that have pointers to DOVF or IOVF blocks is computed and compared with a user-specified threshold. If the computed RAP overflow percentage is greater than the user-specified setting, a warning message is issued.

NOTE

If you do not specify the RAPOVERFLOW_PERCENT keyword, this threshold test is not performed.

Use

RAPOVERFLOW_PERCENT is an optional keyword for the THRESHOLD subcommand.

Syntax

RAPOVERFLOW_PERCENT=parameter

Parameters

Specify a value from 0 to 100.

Default

None

Alias

ROP
RAP_VALIDATION

Purpose

Use the RAP_VALIDATION keyword to specify logical validation of Root Anchor Points (RAPs). You can cause the analysis function to build a complete cross-reference of every RAP and root segment occurrence. You can also cause the analysis function to call the randomizer module to validate the following:

- that each root segment is located in the proper area
- that each root segment is located in the proper RAP

Use

RAP VALIDATION is an optional keyword for the following commands:

- ANALYZE
- CHANGE
- EXTEND
- GLOBAL
- IMAGECOPY
- PREPARE
- RELOAD
- REORGANIZE

Related keywords

- POINTER_VALIDATION
- SDEP_VALIDATION

Syntax

Use either of the following syntaxes with the ANALYZE, CHANGE, EXTEND, GLOBAL, IMAGECOPY, RELOAD, and REORGANIZE commands:

- RAP_VALIDATION=xref
- RAP_VALIDATION=(xref, placement)

Use either of the following syntaxes with the PREPARE command:

- RAP_VALIDATION=(xref,placement,rerandomize)
- RAP_VALIDATION=(xref,placement,)
- RAP_VALIDATION=(xref,,rerandomize)
- RAP_VALIDATION=(xref,,)
- RAP_VALIDATION=(,placement,rerandomize)
Parameters

Specify one of the following values for the xref parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XREF</td>
<td>Perform cross-reference validation.</td>
</tr>
<tr>
<td>NOXREF</td>
<td>Do not perform cross-reference validation.</td>
</tr>
</tbody>
</table>

Specify one of the following values for the placement parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLACEMENT</td>
<td>Call the randomizer to validate root segment placement.</td>
</tr>
<tr>
<td>NOPLACEMENT</td>
<td>Do not call randomizer to validate root segment placement.</td>
</tr>
</tbody>
</table>

RAP_VALIDATION=(XREF, PLACEMENT, RERANDOMIZE) indicates that the randomizer for the output is to be called to perform root randomization analysis for the PREPARE command.

**NOTE**

You cannot modify these parameters.

Default

RAP_VALIDATION=(NOXREF, NOPLACEMENT)

RAP_VALIDATION=(XREF, PLACEMENT, RERANDOMIZE) is the default when used with the PREPARE command.

Alias

None
RECORD IO AVERAGE

Purpose

Use the RECORD_IO_AVERAGE keyword to specify a threshold setting when analyzing or monitoring DEDB activity. The average number of physical I/Os necessary to read a database record is computed and compared with a user-specified setting. If the computed average is greater than the user-specified setting, a warning message is issued.

NOTE
If you do not specify the RECORD_IO_AVERAGE keyword, this threshold test is not performed.

Use

RECORD_IO_AVERAGE is an optional keyword for the THRESHOLD subcommand.

Related keyword

POINTER_VALIDATION (required)

NOTE
This threshold test is performed only if you also specify POINTER_VALIDATION=FULL.

Syntax

RECORD_IO_AVERAGE=parameter

Parameters

Specify a value from 0 to 32767.

Default

None

Alias

RECIOA
RECORD_IO_MAXIMUM

Purpose

Use the RECORD_IO_MAXIMUM keyword to specify a threshold setting when analyzing or monitoring DEDB activity. The maximum number of physical I/Os necessary to read a database record is computed and compared with a user-specified setting. If the computed maximum I/O is greater than the user-specified setting, a warning message is issued.

NOTE
If you do not specify the RECORD_IO_MAXIMUM keyword, this threshold test is not performed.

Use

RECORD_IO_MAXIMUM is an optional keyword for the THRESHOLD subcommand.

Related keyword

POINTER_VALIDATION (required)

NOTE
This threshold test is performed only if you also specify POINTER_VALIDATION=FULL.

Syntax

RECORD_IO_MAXIMUM=parameter

Parameters

Specify a value from 0 to 32767.

Default

None

Alias

RECIOM
RECORD_IOVF_PERCENT

Purpose

Use the RECORD_IOVF_PERCENT keyword to specify a threshold setting when analyzing or monitoring DEDB activity. The percentage of records that use one or more IOVF blocks is computed and compared with the user-specified percentage threshold setting. If the percentage is greater than the user-specified setting, a warning message is issued.

NOTE

If you do not specify the RECORD_IOVF_PERCENT keyword, this threshold test is not performed.

Use

RECORD_IOVF_PERCENT is an optional keyword for the THRESHOLD subcommand.

Related keyword

POINTER_VALIDATION (required)

NOTE

This threshold test is performed only if you also specify POINTER_VALIDATION=FULL.

Syntax

RECORD_IOVF_PERCENT=parameter

Parameters

Specify a value from 0 to 100.

Default

None

Alias

RIP
RECORD_LENGTH_ANALYSIS

Purpose

Use the RECORD_LENGTH_ANALYSIS keyword to request the Record Length Analysis report. This report shows the database record lengths, including statistics on average, minimum, and maximum record lengths in each of 21 user-specified reporting intervals.

NOTE

This report is produced only if you also specify POINTER_VALIDATION=FULL.

Use

RECORD_LENGTH_ANALYSIS is an optional keyword for the REPORT subcommand.

Related keywords

- LARGEST_DATABASE_RECORDS
- POINTER_VALIDATION (required)
- RECORD_LENGTH_INCREMENT
- REPORT_DEFAULT

Syntax

RECORD_LENGTH_ANALYSIS=parameter

Parameters

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not produce the report.</td>
</tr>
<tr>
<td>YES</td>
<td>Produce the report.</td>
</tr>
</tbody>
</table>

Default

The default parameter is set by the REPORT_DEFAULT keyword.

Alias

RLA
RECORD_LENGTH_INCREMENT

Purpose

Use the RECORD_LENGTH_INCREMENT keyword to specify the record length interval for the Record Length Analysis report.

Use

RECORD_LENGTH_INCREMENT is an optional keyword for the REPORT subcommand.

Related keyword

RECORD_LENGTH_ANALYSIS

Syntax

RECORD_LENGTH_INCREMENT=parameter

Parameters

Specify a value from 10 to 200,000,000.

Default

RECORD_LENGTH_INCREMENT=250

Alias

RLI

RECORD_PLACEMENT_ANALYSIS

Purpose

Use the RECORD_PLACEMENT_ANALYSIS keyword to request the Record Placement Analysis report. This report shows a profile of record placement within an area, including statistics on the number of DOVF and IOVF blocks used.
Use

RECORD_PLACEMENT_ANALYSIS is an optional keyword for the REPORT subcommand.

Related keywords

- POINTER_VALIDATION (required)
- REPORT_DEFAULT

**NOTE**

This report is produced only if you also specify POINTER_VALIDATION=FULL.

Syntax

RECORD_PLACEMENT_ANALYSIS=parameter

Parameters

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not produce the report.</td>
</tr>
<tr>
<td>YES</td>
<td>Produce the report.</td>
</tr>
</tbody>
</table>

Default

The default parameter is set by the REPORT_DEFAULT keyword.

Alias

RPLA

**RECORD_PROFILE_ANALYSIS**

Purpose

Use the RECORD_PROFILE_ANALYSIS keyword to request the Record Profile Analysis report. This report shows the number of database records; average, maximum, and minimum lengths; segment frequencies; and segment lengths.
Use

RECORD_PROFILE_ANALYSIS is an optional keyword for the REPORT subcommand.

Related keyword

REPORT_DEFAULT

Syntax

RECORD_PROFILE_ANALYSIS=parameter

Parameters

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not produce the report.</td>
</tr>
<tr>
<td>YES</td>
<td>Produce the report.</td>
</tr>
</tbody>
</table>

Default

The default parameter is set by the REPORT_DEFAULT keyword.

Alias

RPRA

RECOVJCL

Purpose

Use the RECOVJCL keyword to specify the partitioned data set member that contains skeletal JCL. When you issue the GENJCL.RECOV command, DBRC uses the member that you specify to generate the JCL that runs the Database Recovery utility for the identified DBDS or area.

Use

RECOVJCL is an optional keyword for the REGISTER subcommand.
**RECOVPD**

**Purpose**
Use the RECOVPD keyword to specify the recovery period for a specified DBDS or DEDB area.

**Use**
RECOVPD is an optional keyword for the REGISTER subcommand.

**Related keywords**
None

**Syntax**

```
RECOVPD=parameter
```

**Parameters**
The parameter specifies a numeric value for the recovery period.
Default

The default value is 0.

Alias

None

RECVJCL

Purpose

Use the RECVJCL keyword to specify the partitioned data set member that contains skeletal JCL. The GENJCL.RECEIVE command uses the member.

Use

RECVJCL is an optional keyword for the REGISTER subcommand.

Related keywords

None

Syntax

RECVJCL=parameter

Parameters

The parameter specifies the member name.

Default

The default value is RECVJCL.

Alias

None
REGISTER

Purpose

Use the REGISTER subcommand to register with DBRC a new area that is inserted into or appended to an area list. The area is registered during the restructure process.

Use

REGISTER is an optional subcommand for the PREPARE command.

Available keywords

- ADDN
- ADSN
- CFSTR1
- CFSTR2
- DEFLTJCL
- FULLSEG
- GENMAX
- GSGNAME
- ICJCL
- LKASID
- MAS
- OAREA
- PRELOAD
- PREOPEN
- RECOVJCL
- RECOVPD
- RECVJCL
- REUSE
- TRACK
- VSO

Syntax

PREPARE [keyword=parameter[,...]]
REGISTER [keyword=parameter[,...]]

Parameters

None
Default

None

Alias

None

RELOAD

Purpose

Use the RELOAD command to load in offline mode one or more areas of a DEDB from input compatible with the IBM HD Reorganization Reload utility. The ACB of the database being reloaded is used to describe the areas being loaded and is referenced by the IMSACB DD statement.

NOTE

Only one RELOAD command can be executed per job step, and it must be the only command specified in the PFPSYSIN input.

Use

RELOAD is an optional command for the PFPSYSIN DD statement. You must have a license for Fast Path Online Reorg/EP or Fast Path Reorg/EP to use this command.

Available keywords

- BYPASS_RECORD
- COMPRESS
- DBD (required)
- ERROR_THRESHOLD
- EXPAND
- EXCEPTION_LIMIT
- IAREA
- INDEX_THREADS
- INPUT_DSN_MASK
- INPUT_THREADS
- IOVF_LOAD_HWM
- LARGEST_DATABASE_RECORDS
- OAREA
- OCACHE
REORGANIZE

- OUTPUT_DSN_MASK
- OUTPUT_THREADS
- POINTER_VALIDATION
- SDEP_PROCESS
- SDEP_VALIDATION
- SORT
- SORT_OPTION
- SUBSET_POINTERS

Syntax

RELOAD [keyword=parameter[,...]]
    [subcommand [keyword=parameter[,...]]]

Parameters

None

Default

None

Alias

None

REORGANIZE

Purpose

Use the REORGANIZE command to reorganize a DEDB in place. Each UOW is read in order to determine the need for reorganization. If the reorganization function determines by the settings of the SELECT_UOW and FRAGMENTATION_PERCENT keywords that a UOW needs reorganization, then it reorganizes the UOW.

Use

REORGANIZE is an optional command for the PFPSYSIN DD statement. You must have a license for Fast Path Online Reorg/EP or Fast Path Reorg/EP to use this command.
Available keywords

- COMPRESS
- DBD (required for offline mode)
- ERROR_THRESHOLD
- EXTEND_IOVF_#UOWS (restricted in online mode)
- EXTEND_SDEP_#CIS
- FRAGMENTATION_PERCENT
- IAREA
- INPUT_DSN_MASK (ignored in online mode)
- IOVF_SAVE_THRESHOLD
- LARGEST_DATABASE_RECORDS
- POINTER_VALIDATION
- RAP_VALIDATION
- SDEP_VALIDATION
- SELECT_UOW

Syntax

REORGANIZE [keyword=parameter...]
   [subcommand [keyword=parameter[...]]]

Parameters

None

Default

None

Alias

None

REPORT

Purpose

Use the REPORT subcommand to request that the analysis function produce reports for display or printing.
Use

REPORT is an optional subcommand for the following commands:

- ANALYZE
- CHANGE
- EXTEND
- GLOBAL
- IMAGECOPY
- PREPARE
- RELOAD
- REORGANIZE
- RETRIEVE

Available keywords

- FREESPACE_ANALYSIS
- IOVF_SPACE_ANALYSIS
- POINTER_ANALYSIS
- RECORD_LENGTH_ANALYSIS
- RECORD_LENGTH_INCREMENT
- RECORD_PLACEMENT_ANALYSIS
- RECORD_PROFILE_ANALYSIS
- REPORT_DDNAME
- REPORT_DEFAULT
- REPORT_HEADING
- REPORT_LINE_COUNT
- REPORT_IO_ANALYSIS
- SEGMENT_LENGTH_ANALYSIS
- SEGMENT_PLACEMENT_ANALYSIS
- STARTUOW
- STOPUOW
- SYNONYM_CHAIN_ANALYSIS
- SYNONYM_CHAIN_INCREMENT
- UOW_DETAILED_ANALYSIS

Syntax

```
command [keyword=parameter[,...]]
REPORT [keyword=parameter[,...]]
```
REPORT_DDNAME

Parameters

None

Default

None

Alias

None

REPORT_DDNAME

Purpose

Use the REPORT_DDNAME keyword to specify the destination ddname(s) for one or more reports requested with the associated subcommand. You can use the JCL parameters on the referenced DD statement(s) to place report(s) on tape, disk, or another storage medium. If the JCL does not contain a DD statement(s) with the referenced ddname(s), an error message is issued.

WARNING

Multiple report sets will be written to the same DD statement if the ddname is used on more than one command or if multiple areas are being reported. The DD statement referenced will be OPENed and CLOSEd for each report set that is written to it. If more than one report set will be written to the DD statement, DISP=MOD must be used to ensure that all reports are available.

The output data set must be a sequential data set. The record format (RECFM) can specify fixed or variable length records, blocked or unblocked, and can include ANSI carriage control. The logical record length (LRECL) can be any length. If an output record exceeds the LRECL, the record is truncated. If an output record is shorter than the LRECL, the record is padded with trailing blanks. The block size (BLKSIZE) can be any value that is appropriate for the LRECL and RECFM.

If you specify ANSI carriage control, a control character is generated for each logical record. If you do not specify ANSI carriage control, the Fast Path/EP product generates blank lines to simulate any carriage control function requested.

The following DCB characteristics are recommended:

RECFM=VBA,LRECL=137,BLKSIZE=4096
Use

REPORT_DDNAME is an optional keyword for the following subcommands:

- ALLOCATE
- REPORT

Syntax

REPORT_DDNAME=parameter
REPORT_DDNAME=(parameter[,parameter ...])

Parameters

Specify a 1-character to 8-character ddname of a JCL statement, or an asterisk (*) to request dynamic allocation of a SYSOUT=* data set.

Default

None

Alias

RPTDD

REPORT_DEFAULT

Purpose

Use the REPORT_DEFAULT keyword to specify the default parameter for the keywords on the REPORT subcommand. All report keywords (see Related Keywords) are set with a default of YES or NO based on the parameter of the REPORT_DEFAULT keyword.

Use

REPORT_DEFAULT is an optional keyword for the REPORT subcommand.
Related keywords

- FREESPACE_ANALYSIS
- IOVF_SPACE_ANALYSIS
- RECORD_LENGTH_ANALYSIS
- RECORD_PLACEMENT_ANALYSIS
- RECORD_PROFILE_ANALYSIS
- SEGMENT_IO_ANALYSIS
- SEGMENT_LENGTH_ANALYSIS
- SEGMENT_PLACEMENT_ANALYSIS
- SYNONYM_CHAIN_ANALYSIS
- UOW_DETAILED_ANALYSIS

Syntax

REPORT_DEFAULT=parameter

Parameters

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>Set the default parameter for the report keywords to YES.</td>
</tr>
<tr>
<td>NO</td>
<td>Set the default parameter for the report keywords to NO.</td>
</tr>
</tbody>
</table>

Default

REPORT_DEFAULT=YES

Alias

DEFAULT

**REPORT_HEADING**

Purpose

Use the REPORT_HEADING keyword to specify user-customized heading text for reports. This text is displayed at the right edge of each report on the fifth line from the top of the report.
Use

REPORT_HEADING is an optional keyword for the REPORT subcommand.

Syntax

REPORT_HEADING=parameter

Parameters

Specify a 1-character to 40-character string. If blanks or other special characters are embedded in the string, enclose the string in single quotation marks.

Default

REPORT_HEADING=’ ’ (The parameter field is blank; no heading is written.)

Alias

RPTH

REPORT_LINE_COUNT

Purpose

Use the REPORT_LINE_COUNT keyword to specify the number of lines on a report page.

Use

REPORT_LINE_COUNT is an optional keyword for the REPORT subcommand.

Related keyword

LINE_COUNT

Syntax

REPORT_LINE_COUNT=parameter

Parameters

Specify a value from 16 to 32767.
REPOSITORY_DSNAME

Default

The default parameter setting is the user-specified value for the LINE_COUNT keyword on the OPTIONS command.

Alias

RPTLC

REPOSITORY_DSNAME

Purpose

Use the REPOSITORY_DSNAME keyword to specify the name of the repository data set to be processed by the analysis function. This keyword can be used to activate repository processing and to override dynamic allocation of the DFSMDA repository data set.

The REPOSITORY_DSNAME keyword also can be used to specify the name of the repository catalog to be processed by the PROCESS_EPR command. This keyword can be specified on the PFPEPR00 batch utility program to request maintenance processing for the allocation rules or statistics records stored in the repository catalog. It can also be used to store, list, or delete user-specified customizations to selected product messages.

NOTE

If you do not specify the REPOSITORY_DSNAME keyword and do not supply the PFPEPR DD statement on SYSIN for PFPMAIN, dynamic allocation of the repository catalog data set name will be performed by using the DFSMDA member. However, specifying the PFPEPR statement or the REPOSITORY_DSNAME overrides dynamic allocation of the DFSMDA member.

Use

REPOSITORY_DSNAME is an optional keyword for the OPTIONS command. This keyword is ignored on the OPTIONS command when executing the PFPEPR00 utility program.

REPOSITORY_DSNAME is a required keyword for the PROCESS_EPR command.

Syntax

REPOSITORY_DSNAME='parameter'
Parameters

Specify a 1-character to 44-character data set name. You must enclose the data set name in single quotation marks.

Default

None

Alias

EPR

REPOSITORY_GROUP

Purpose

Use the REPOSITORY_GROUP keyword to specify a group identifier. The group identifier is used as a part of the record key for the repository catalog data set. It is used in the search for the dynamic allocation options for creating statistics data sets, and for adding the statistics entry to the repository catalog.

The REPOSITORY_GROUP keyword provides a means for the user to distinguish between two or more database areas having identical names. For example, to distinguish between the test and production versions of a database, set REPOSITORY_GROUP=TEST or REPOSITORY_GROUP=PROD.

Use

REPOSITORY_GROUP is an optional keyword for the OPTIONS command.

Syntax

REPOSITORY_GROUP=parameter

Parameters

Specify a 1-character to 4-character group name.

Default

REPOSITORY_GROUP=xxxx (if offline)
REPOSITORY_GROUP=yyyy (if online, where yyyy is the IMS subsystem ID)
REPOSITORY_OVERWRITE

**Purpose**

Use the REPOSITORY_OVERWRITE keyword to specify whether an existing data set will be overwritten by a new repository statistics file. This keyword has no effect if the repository statistics data set does not already exist.

**Use**

REPOSITORY_OVERWRITE is an optional keyword for the OPTIONS command:

**Syntax**

REPOSITORY_OVERWRITE=parameter

**Parameters**

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not overwrite an existing statistics data set.</td>
</tr>
<tr>
<td>YES</td>
<td>Overwrite an existing statistics data set.</td>
</tr>
</tbody>
</table>

**Default**

REPOSITORY_OVERWRITE=NO

**Alias**

None
REPOSITORY_RETENTION_COUNT

Purpose

Use the REPOSITORY_RETENTION_COUNT keyword to specify the maximum number of statistics catalog entries that will remain recorded within the repository catalog. This keyword functions in a similar manner to a GDG by deleting the oldest statistics catalog entries from the repository when the number of catalog entries exceeds the numeric value that is specified.

**NOTE**

The REPOSITORY_RETENTION_COUNT keyword can be specified in conjunction with the REPOSITORY_RETENTION_PERIOD keyword. Fast Path/EP will remove a statistics catalog entry from the repository when criteria specified on either of these keywords is met.

Use

REPOSITORY_RETENTION_COUNT is an optional keyword for the OPTIONS command.

Related keyword

REPOSITORY_RETENTION_PERIOD

Syntax

REPOSITORY_RETENTION_COUNT=parameter

Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Catalog entries are retained permanently (no limit is applied to the number of statistics catalog entries that can be stored).</td>
</tr>
<tr>
<td>1-32767</td>
<td>The oldest catalog entries are removed when the number of entries exceeds this value.</td>
</tr>
</tbody>
</table>

Default

REPOSITORY_RETENTION_COUNT=0
REPOSITORY_RETENTION_PERIOD

Purpose

Use the REPOSITORY_RETENTION_PERIOD keyword to specify the number of days that a statistics catalog entry will remain recorded within the repository catalog.

**NOTE**
The REPOSITORY_RETENTION_PERIOD keyword can be specified in conjunction with the REPOSITORY_RETENTION_COUNT keyword. Fast Path/EP will remove a statistics data set from the repository when criteria specified on either of these keywords is met.

Use

REPOSITORY_RETENTION_PERIOD is an optional keyword for the following command and subcommand:

- DELETE
- OPTIONS

Related keyword

REPOSITORY_RETENTION_COUNT

Syntax

REPOSITORY_RETENTION_PERIOD=parameter

Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The catalog entry is retained permanently.</td>
</tr>
<tr>
<td>1-32767</td>
<td>The catalog entry is removed after the specified number of days.</td>
</tr>
</tbody>
</table>
REQUIRE_AREA

Purpose

For the Prepare function, use the optional REQUIRE_AREA keyword to specify one or more areas to be selected for input (unload) and output (reload) into the restructure process, even if prepare processing did not select them. You can specify one or more areas, a range of areas, or a combination of areas and area ranges. When the REQUIRE_AREA keyword is specified, these areas are added to the list of areas that the prepare process has selected to be included in the restructure process.

The EMPTY keyword specifies whether to select empty areas for the restructure process, even if prepare processing did not select them.

NOTE

BMC does not recommend using the REQUIRE_AREA keyword. Using REQUIRE_AREA nullifies the intelligence of the prepare process to automatically select the areas that need to be processed by the Restructure function; the REQUIRE_AREA keyword should normally be omitted.

Use

REQUIRE_AREA is an optional keyword for the PREPARE command.

Syntax

REQUIRE_AREA=ALL or REQUIRE_AREA=*  
REQUIRE_AREA=parameter  
REQUIRE_AREA='parameter'  
REQUIRE_AREA=(parameter1, parameter2, ..., parametern)  
REQUIRE_AREA=(RANGE=(parameter1, parametern))  
REQUIRE_AREA=(RANGE=(parameter1, parameter2).  
  RANGE=(parameter3, parameter4)...  
REQUIRE_AREA=(RANGE=(*,parametern))  
REQUIRE_AREA=(RANGE=(parameter1,*))  
REQUIRE_AREA=(parameter,...[RANGE=(parameter1,parametern),]  
  [EMPTY=YES|NO,...]),]
REQUIRE_AREA

REQUIRE_AREA=(EMPTY=YES,...)
REQUIRE_AREA=(parameter,...EMPTY=YES,...)

**NOTE**

IAREA= parameter includes wildcard masks. For example, the syntax for areas DPAS7A0, DPAS7B0, and DPAS7C0 can be specified as follows:

- REQUIRE_AREA='DP????C0'
- REQUIRE_AREA='DPAS7*'
- REQUIRE_AREA='DPA7??'
- REQUIRE_AREA='DPAS7??'

### Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>areaname</td>
<td>Specify one or more areas by using the 1-character to 8-character area name for each area specified.</td>
</tr>
<tr>
<td>areaname mask</td>
<td>You can also specify one or more areas by using the 1-character to 8-character area name mask for each area specified. Wildcard masks using the asterisk (*) and question mark (?) can be included. When using wildcard masks, the value of the keyword must be enclosed in quotation marks. The asterisk wildcard matches all characters to the right of the asterisk. The question mark wildcard matches the characters in the string in the place the question mark holds.</td>
</tr>
<tr>
<td>areanumber</td>
<td>Specify one or more areas by using the 1-character to 5-character area number for each area specified.</td>
</tr>
<tr>
<td>RANGE</td>
<td>Use this syntax to specify a consecutive range of areas using either areaname or areanumber parameters. The area number associated with startarea must be less than (and not equal to) the area number associated with endarea. When the asterisk is used with the RANGE keyword, it can be used to specify the beginning or ending range for specific areas of the DEDB.</td>
</tr>
</tbody>
</table>

#### Default

REQUIRE_AREA=ALL

**Note:** Wildcard masks cannot be used with the RANGE keyword.
Alias

RAREA

RESET

Purpose

Use the RESET subcommand to cancel any or all customizations that were specified for the suppression or suffix level changes to selected messages with the OVERRIDE subcommand. The RESET subcommand is executed under the PFPEPR00 utility to cancel all or selected changes to messages that were stored in the repository by a prior execution of the OVERRIDE subcommand.

The RESET subcommand can also be specified on the PFPOPTS DD statement to reset messages that were customized using the PFPEPR00 utility to product defaults for the processes specified on the PFPSYSIN DD statement.

Use

RESET is an optional subcommand for the following commands:

- OPTIONS
- PROCESS_EPR

Related keyword

MESSAGE_NUMBER (required)

Syntax

PROCESS_EPR [keyword=parameter[...]]
RESET [keyword=parameter[...]]

Parameters

None

Default

None
**RESTART**

**Purpose**

Use the RESTART command to restart post-processing tasks if a failure occurs during restructure post-processing activities. Use the PLAN_FILECTL subcommand with the RESTART command to specify the name of the Restructure Plan data set, which is used for input into the restart process.

**Use**

RESTART is an optional command for the PFPSYSIN DD statement when you execute the PFCMAIN program. You must have a license for Fast Path Online Restructure/EP to use this command.

**Available keywords**

- ACCESS (required)
- DETAIL
- DBD (required)
- EARLY_TERMINATION

**Syntax**

```plaintext
RESTART [keyword=parameter[,...]]
   [subcommand[keyword=parameter[,...]]]
```

**Parameters**

None

**Default**

None

**Alias**

None
RESTRUCTURE

Purpose

Use the RESTRUCTURE command to restructure a DEDB while it is online to IMS. Use the PLAN_FILECTL subcommand with the RESTRUCTURE command to specify the name of the Restructure Plan data set, which is used for input to the restructure process.

**NOTE**

You cannot specify multiple RESTRUCTURE commands in the same command stream; you can specify only one.

Also, you cannot use a RESTRUCTURE command in the same command stream with a PREPARE, SHADOW_INIT, or RESTART command.

Use

RESTRUCTURE is an optional command for the PFPSYSIN DD statement when you execute the PFCMAIN program. You must have a license for Fast Path Online Restructure/EP to use this command.

Available keywords

- ACCESS (required)
- DBD (required)
- DETAIL
- EARLY_TERMINATION
- MESSAGE_SUPPRESSION
- OUTAGE_WINDOW

Syntax

RESTRUCTURE [keyword=parameter[,...]]
[subcommand[keyword=parameter[,...]]]

Parameters

None

Default

None
RESYNC

Purpose

Use the RESYNC command to resynchronize a secondary index database with its associated primary DEDB in online (BMP) mode only. You can use the RESYNC command with PFX indexes if you provide the appropriate password. (For more information about the authentication process, see the Fast Path Indexer/EP User Guide.)

Use

RESYNC is an optional command for the PFPSYSIN DD statement. You must have a license for Fast Path Indexer/EP to use this command.

Available keywords

- CHECKPOINT
- DBD (required)
- IAREA
- INDEX_THREADS
- INPUT_DSN_MASK (ignored in BMP mode)
- MESSAGE_SUPPRESSION
- SORT
- SORT_OPTION

Syntax

RESYNC [keyword=parameter[,...]]
subcommand [keyword=parameter[,...]]

Parameters

None

Default

None
Alias

None

**RETAINED_SUFFIX**

**Purpose**

Use the RETAINED_SUFFIX keyword to rename the original area data set name after the restructure process is complete.

**Use**

RETAINED_SUFFIX is an optional keyword for the PREPARE command.

**Related keyword**

SHADOW_SUFFIX

**Syntax**

\[
\text{RETAINED\_SUFFIX} = \text{parameter}
\]

**Parameters**

Specify a 1-character to 8-character suffix to be added to the end of the original area data set name after the DEDB online restructure process is complete.

---

**NOTE**

Because of VSAM restrictions, the name of the original (old) area data set must not exceed 44 characters.

**Default**

RETAINED_SUFFIX=O

**Alias**

ORIGINAL_SUFFIX
RETPD

Purpose

Use the RETPD keyword to specify the retention period of an output data set created by dynamic allocation.

**NOTE**

Do not specify both the RETPD keyword and the EXPDT keyword; these keywords are mutually exclusive.

The RETPD keyword can also be used with the PFPEPR00 batch utility program to specify or modify a retention period for an allocation rule stored in the repository catalog associated with the Fast Path Analyzer/EP repository facility.

Use

RETPD is an optional keyword for the following subcommands:

- ACTIVITY_FILECTL
- ADD
- DISCARD_FILECTL
- IC
- MODIFY
- OFILECTL
- PLAN_FILECTL

Related keywords

- DSNAME
- EXPDT

Syntax

RETPD=parameter

Parameters

Specify the number of days to retain the data set. Valid values are from 0 to 32767.

Default

None
Alias

None

RETRIEVE

Purpose

RETRIEVE can be used as either a command or a subcommand to request reports from accumulated statistics in the Fast Path Analyzer/EP statistics repository facility. RETRIEVE functions in the same manner, whether it is used as a command or a subcommand.

Use

RETRIEVE is an optional command that can be specified on the PFPSYSIN DD statement when executing the PFPMAIN offline utility program. RETRIEVE is also an optional subcommand that can be specified on the PFPSYSIN DD statement under the PROCESS_EPR command when executing the PFPEPR00 repository facility program.

You must have a license for Fast Path Online Analyzer/EP or Fast Path Analyzer/EP to use this command.

Available keywords

- HISTORY_DDNAME
- SELECT_AREA
- SELECT_DBDB
- SELECT_DATE
- SELECT_GROUP
- SELECT_LIMIT

Syntax

Use the following syntax when using RETRIEVE as a command:

RETRIEVE [keyword=parameter[...]]
   [subcommand[keyword=parameter[...]]]

Use the following syntax when using RETRIEVE as a subcommand:

PROCESS_EPR REPOSITORY_DSNNAME=keyword
   RETRIEVE[keyword=parameter[...]]
   [subcommand[keyword=parameter[...]]]
REUSE

Parameters

None

Default

None

Alias

None

REUSE

Purpose

Use the REUSE keyword to specify whether the supported image copy utilities should reuse image copy data sets that were previously used.

Use

REUSE is an optional keyword for the REGISTER subcommand.

Related keywords

None

Syntax

REUSE=parameter

Parameters

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not reuse the image copy data sets that were previously used.</td>
</tr>
<tr>
<td>YES</td>
<td>Reuse the image copy data sets that were previously used.</td>
</tr>
</tbody>
</table>

Default

The default value is NO.
ROOT_IO_AVERAGE

Purpose

Use the ROOT_IO_AVERAGE keyword to specify a threshold setting when analyzing or monitoring DEDB activity. The average number of I/Os necessary to find a root segment by key is computed and compared with a user-specified setting. If the computed average I/O is greater than the user-specified setting, a warning message is issued.

NOTE

If you do not specify the ROOT_IO_AVERAGE keyword, this threshold test is not performed.

Use

ROOT_IO_AVERAGE is an optional keyword for the THRESHOLD subcommand.

Related keywords

- POINTER_VALIDATION
- RAP_VALIDATION

NOTE

This threshold test is performed only if you also specify either POINTER_VALIDATION=FULL or RAP_VALIDATION=XREF.

Syntax

ROOT_IO_AVERAGE=parameter

Parameters

Specify a value from 0 to 32767.

Default

None
ROOT_IO_MAXIMUM

Purpose

Use the ROOT_IO_MAXIMUM keyword to specify a threshold setting when analyzing or monitoring DEDB activity. The maximum number of I/Os necessary to find a root segment by key is computed and compared with a user-specified setting. If the computed maximum I/O is greater than the user-specified setting, a warning message is issued.

**NOTE**

If you do not specify the ROOT_IO_MAXIMUM keyword, this threshold test is not performed.

Use

ROOT_IO_MAXIMUM is an optional keyword for the THRESHOLD subcommand.

Related keywords

- POINTER_VALIDATION
- RAP_VALIDATION

**NOTE**

This threshold test is performed only if you also specify either POINTER_VALIDATION=FULL or RAP_VALIDATION=XREF.

Syntax

```
ROOT_IO_MAXIMUM=parameter
```

Parameters

Specify a value from 0 to 32767.

Default

None


**ROUTCDE**

**Purpose**

Use the ROUTCDE keyword to specify the WTO message routing code(s) to be used for messages sent to the system operator.

**Use**

ROUTCDE is an optional keyword for the OPTIONS command.

**Related keyword**

DESC

**Syntax**

ROUTCDE=\*parameter
ROUTCDE=(\*parameter1, \*parameter2, \ldots, \*parametern)

**Parameters**

Specify one or more numeric values from 1 to 16.

**Default**

ROUTCDE=2

**NOTE**

Although this keyword is similar to the ROUTCDE keyword on the WTO macro, values 17 to 28 are not supported. The ROUTCDE keyword has the same format and follows the same coding practices and rules as the ROUTCDE keyword on the WTO macro.

**Alias**

None
SAMPLE_INTERVAL

Purpose

Use the SAMPLE_INTERVAL keyword to set the frequency of the segments selected for processing according to the associated subcommand. For example, if you specify SAMPLE_INTERVAL=5, every fifth segment will be selected. For a root segment, every fifth occurrence in the database and all of its dependents will be selected. For a dependent segment, every fifth occurrence under its parent will be selected.

Use

SAMPLE_INTERVAL is an optional keyword for the following subcommands:

- EXCLUDE
- INCLUDE

Related keyword

SAMPLE_LIMIT

Syntax

SAMPLE_INTERVAL=parameter

Parameters

Specify a value from 0 to 2147483647.

NOTE

Specify 0 to select all segments.

Default

SAMPLE_INTERVAL=0

Alias

INTERVAL
**SAMPLE_LIMIT**

**Purpose**

Use the SAMPLE_LIMIT keyword to set the maximum number of segments selected for processing by the associated subcommand. For example, if you specify "SAMPLE_LIMIT=500", the first 500 segments will be selected. For a root segment, the first 500 occurrences and all its dependents will be selected. For a dependent segment, the first 500 occurrences within its parent will be selected.

**Use**

SAMPLE_LIMIT is an optional keyword for the following subcommands:

- EXCLUDE
- INCLUDE

**Related keyword**

SAMPLE_INTERVAL

**Syntax**

SAMPLE_LIMIT=parameter

**Parameters**

Specify a value from 0 to 2147483647.

**NOTE**

Specify 0 to specify no limit on the number of segments selected.

**Default**

SAMPLE_LIMIT=0

**Alias**

LIMIT
SCAN

Purpose

Use the SCAN keyword to verify the syntax of Fast Path/EP commands, keywords, and parameters in command sets without executing Fast Path/EP products. Messages concerning the command set are routed to the data set referenced by the PFPRINT DD statement.

Use

SCAN is an optional keyword for the GLOBAL command:

Syntax

SCAN=parameter

Parameters

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not verify the syntax and parameters of the command set.</td>
</tr>
<tr>
<td>YES</td>
<td>Verify the syntax and parameters of the command set. When specified with the RESTRUCTURE command, the Restructure function reads the Restructure Plan data set, merges the content of the data set, and terminates before executing the actual restructure process.</td>
</tr>
</tbody>
</table>

Default

SCAN=NO

Alias

None
SCRIPT

Purpose
Use the SCRIPT keyword to specify a procedure to be performed.

Use
SCRIPT is a required keyword for the PERFORM subcommand.

Syntax
SCRIPT={ script ( ); }

Parameters
For details, see Chapter 3, “Expression syntax.”

Default
SCRIPT={ DMAC_PRINT( ); }

Alias
None

SDEP_PROCESS

Purpose
Use the SDEP_PROCESS keyword to specify how SDEP segment data is to be processed.
SDEP_PROCESS

Use

If the DEDB being processed has an SDEP segment defined, SDEP_PROCESS is a required keyword for the following commands:

- CHANGE
- UNLOAD

If the DEDB is being reloaded and has an SDEP segment defined, SDEP_PROCESS is an optional keyword for the RELOAD command.

Related keyword

ERROR_THRESHOLD

Syntax

SDEP_PROCESS=parameter

Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGICAL</td>
<td>This value is valid only on the CHANGE and UNLOAD commands. All valid SDEP segments are timestamped and written in logical order to the new area or unload file. The SDEP segments in each area of the new database area are physically reordered into the new area/RAP sequence of their parent root segment, but retain their logical order (entry sequence) within each database record. <strong>Warning</strong>: If your application relies on the marker segment concept, <em>do not use SDEP_PROCESS=LOGICAL.</em></td>
</tr>
</tbody>
</table>
### SDEP_PROCESS

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| PHYSICAL | This value is valid only on the CHANGE and UNLOAD commands. Processing in physical order results in all segments between the SDEP logical beginning and logical end being extracted in physical sequence and inserted into the SDEP part of the new area. The SDEP segments remain in the same physical order (this mode fully supports the marker segment concept). The following restrictions apply when SDEP_PROCESS=PHYSICAL is selected:  
- You cannot reduce the CI size of the area. You can, however, increase the CI size.  
- You cannot modify the compression parameters for the SDEP segment.  
- You cannot specify the OUTPUT subcommand for the SDEP segment.  
- You cannot modify the randomization parameters for the database such that root segments will be randomized to a different area.  
- You cannot use the FABEUR6 and FABEUR7 program extensions to process the unload file. This restriction applies to the UNLOAD command only.  
**Warning:** Because SDEPs are not processed on a pointer-by-pointer basis when you specify SDEP_PROCESS=PHYSICAL, the ERROR_THRESHOLD keyword will not detect pointer errors in the SDEP portion of the database. If you want the ERROR_THRESHOLD keyword to detect and bypass any pointer errors detected in the SDEP portion of the database, then you must specify SDEP_PROCESS=LOGICAL. |
| NONE     | This value is valid on the CHANGE, UNLOAD, and RELOAD commands. No SDEP processing occurs. SDEP segments are not written to the new area or unload file. |
| V5COMP   | This value is valid only on the CHANGE and RELOAD commands. All valid SDEP segments are written to the new area or unload file *without a timestamp*. The SDEP segments in each area of the new database area are physically reordered into the new area/RAP sequence of their parent root segment, but retain their logical order (entry sequence) within each database record.  
**Warning:** When the V5COMP value is specified on the RELOAD command, the unload file must have been created with SDEP_PROCESS=LOGICAL. |

**Default**

None

** Alias**

SDEP
SDEP_VALIDATION

Purpose

Use the SDEP_VALIDATION keyword to select the method used by the analysis function to perform logical validation of pointers to sequential dependent (SDEP) segments.

Use

SDEP_VALIDATION is an optional keyword for the following commands:

- ANALYZE
- CHANGE
- EXTEND
- GLOBAL
- IMAGECOPY
- PREPARE
- RELOAD
- REORGANIZE

Related keywords

- EXTEND_IOVF_#UOWS
- ORPHANED_SDEP_MSG
- POINTER_VALIDATION
- RAP_VALIDATION

Syntax

SDEP_VALIDATION=parameter

Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FULL</td>
<td>This value performs a complete cross-reference validation of every pointer and segment occurrence. All pointers are checked to ensure that a segment occurrence appears at the referenced location. The complete chain of sequential dependents associated with the root segment is identified. FULL validation also ensures that every segment occurrence is referenced by only one pointer.</td>
</tr>
</tbody>
</table>
For the EXTEND and REORGANIZE commands, the default value is NONE unless the EXTEND_IOVF_#UOWS keyword specifies a value greater than zero.

For all other commands, the default value is NONE if the DEDB being processed does not have an SDEP segment defined. If an SDEP segment is defined for the DEDB being processed, the default value is set automatically to match the POINTER_VALIDATION keyword value.

**Alias**

None

## SEGMENT

### Purpose

Use the SEGMENT keyword to select segments of the specified name for processing by its associated subcommand.

### Use

SEGMENT is a required keyword for the following subcommands:

- EXCLUDE
- INCLUDE
- LOADCTL
- OUTPUT
Syntax

```
SEGMENT=name
(Valid on the LOADCTL subcommand only) SEGMENT=(name, apply)
```

Parameters

Specify the 1-character to 8-character name of the segment for the `name` parameter.

If using the SEGMENT keyword with the LOADCTL subcommand, specify one of the following values for the `apply` parameter.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ONLY (default)</td>
<td>The LOADCTL action applies to the named segment only.</td>
</tr>
<tr>
<td>DEPENDENTS</td>
<td>The LOADCTL action applies to the dependents of the named segment only.</td>
</tr>
<tr>
<td>BOTH</td>
<td>The LOADCTL action applies to the named segment and its dependents.</td>
</tr>
</tbody>
</table>

Default

None

Alias

None

**SEGMENT_IO_ANALYSIS**

Purpose

Use the SEGMENT_IO_ANALYSIS keyword to request the Segment I/O Analysis Report. This report shows statistics on the placement and physical I/O required to access the dependent segments for a typical database record.

**NOTE**

This report is produced only if you also specify POINTER_VALIDATION=FULL.

Use

SEGMENT_IO_ANALYSIS is an optional keyword for the REPORT subcommand.
Related keywords

- POINTER_VALIDATION
- REPORT_DEFAULT

Syntax

SEGMENTLENGTH_ANALYSIS=parameter

Parameters

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not produce the report.</td>
</tr>
<tr>
<td>YES</td>
<td>Produce the report.</td>
</tr>
</tbody>
</table>

Default

The default parameter is set by the REPORT_DEFAULT keyword.

Alias

SIA

**SEGMENT_LENGTH_ANALYSIS**

Purpose

Use the SEGMENT_LENGTH_ANALYSIS keyword to request the Segment Length Analysis Report. This report shows statistics about the length of segment occurrences within the area.

Use

SEGMENT_LENGTH_ANALYSIS is an optional keyword for the REPORT subcommand.

Related keyword

REPORT_DEFAULT
Syntax

```
SEGMENT_LENGTH_ANALYSIS=parameter
SEGMENT_LENGTH_ANALYSIS=(parameter1, parameter2, ..., parameterN)
```

Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>Produce a report for all segment types defined in the area.</td>
</tr>
<tr>
<td>YES</td>
<td>Produce a report for all segment types defined in the area.</td>
</tr>
<tr>
<td>NONE</td>
<td>Do not produce a report.</td>
</tr>
<tr>
<td>NO</td>
<td>Do not produce a report.</td>
</tr>
<tr>
<td>1-character to 8-character segment name</td>
<td>Produce a report for the specified segment. This value can be used more than once to specify specific segments using the (parameter1, parameter2, ..., parameterN) syntax shown above.</td>
</tr>
</tbody>
</table>

Default

The default parameter is set by the REPORT_DEFAULT keyword.

Alias

SLA

**SEGMENT_PLACEMENT_ANALYSIS**

**Purpose**

Use the SEGMENT_PLACEMENT_ANALYSIS keyword to request the Segment Placement Analysis Report. This report shows segment placement in each component part of an area.

**Use**

SEGMENT_PLACEMENT_ANALYSIS is an optional keyword for the REPORT subcommand.
Related keyword

REPORT_DEFAULT

Syntax

SEGMENT_PLACEMENT_ANALYSIS=parameter

Parameters

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not produce the report.</td>
</tr>
<tr>
<td>YES</td>
<td>Produce the report.</td>
</tr>
</tbody>
</table>

Default

The default parameter is set by the REPORT_DEFAULT keyword.

Alias

SPLA

SEGMENT_RECORD_PREFIX

Purpose

Use the SEGMENT_RECORD_PREFIX keyword to specify the contents of fields within the prefix portion of the segment records written to an extract file.

**NOTE**

You must also specify EXTRACT_FORMAT=USER when the SEGMENT_RECORD_PREFIX keyword is specified.

Use

SEGMENT_RECORD_PREFIX is an optional keyword for the EXTRACT command.
Related keywords

- EXTRACT_FORMAT
- SEGMENT_RECORD_SUFFIX

Syntax

```
SEGMENT_RECORD_PREFIX=(expression [, ...][, VL=YES/NO])
```

Parameters

The parameters consist of a list of one or more expressions, each which identifies a value to be placed in the prefix portion of segment records. See Chapter 3, "Expression syntax" in this book for details.

The value or values resulting from each expression in the parameter list are placed in the output area in the order specified. Unless otherwise directed by use of an explicit conversion operator, the format, size and precision of the value(s) is derived from the operands and operators used in the expression. See Chapter 3, "Expression syntax" in this book for details.

The entire list of values resulting from the expressions can be optionally prefixed with a 2-byte length indicator. Specify VL=YES as a subparameter on the SEGMENT_RECORD_PREFIX keyword to request a length prefix; specify VL=NO to suppress the length prefix.

Default

The default value for the VL parameter is NO: VL=NO.

Alias

PREFIX

**SEGMENT_RECORD_SUFFIX**

Purpose

Use the SEGMENT_RECORD_SUFFIX keyword to specify the contents of fields within the suffix portion of the segment records written to an extract file.

**NOTE**

You must also specify EXTRACT_FORMAT=USER when the SEGMENT_RECORD_SUFFIX keyword is specified.
Use

SEGMENT_RECORD_SUFFIX is an optional keyword for the EXTRACT command.

Related keywords

- EXTRACT_FORMAT
- SEGMENT_RECORD_PREFIX

Syntax

SEGMENT_RECORD_SUFFIX=(expression [,...][,VL=YES/NO])

Parameters

The parameters consist of a list of one or more expressions, each which identifies a value to be placed in the suffix portion of segment records. For details, see Chapter 3, “Expression syntax.”

The value or values resulting from each expression in the parameter list are placed in the output area in the order specified. Unless otherwise directed by use of an explicit conversion operator, the format, size and precision of the value(s) is derived from the operands and operators used in the expression. For details, see Chapter 3, “Expression syntax.”

The entire list of values resulting from the expressions can be optionally prefixed with a 2-byte length indicator. Specify VL=YES as a subparameter on the SEGMENT_RECORD_SUFFIX keyword to request a length prefix; specify VL=NO to suppress the length prefix.

Default

The default value for the VL parameter is NO: VL=NO.

Alias

SUFFIX

SELECT_AREA

Purpose

Use the SELECT_AREA keyword to specify the DEDB area(s) to be retrieved from the repository for reporting.
Use

SELECT_AREA is an optional keyword for the following command and subcommands:

- DELETE
- LIST
- MODIFY
- RETRIEVE (as a command or subcommand)

Related keywords

- SELECT_DATE
- SELECT_DBD
- SELECT_GROUP
- SELECT_LIMIT
- HISTORY_DDNAME

Syntax

SELECT_AREA=parameter
SELECT_AREA=(parameter1, parameter2, ..., parametern)

Parameters

Specify one or more 1-character to 8-character area name masks. Enclose the mask within either single or double quotation marks if it contains any non-alphanumeric characters.

Wildcard masks using characters * and ? can be included. The asterisk wildcard specifies all parameters for all possible characters to the right of the asterisk. The question mark specifies all parameters for the character in the string in the place the question mark holds.

Default

SELECT_AREA=’*’

Alias

AREA
SELECT_DATE

Purpose

Use the SELECT_DATE keyword to specify the date and time range of statistics in the repository to be included in the reports.

Use

SELECT_DATE is an optional keyword for the following command and subcommands:

- DELETE
- LIST
- RETRIEVE (as a command or subcommand)

Related keywords

- SELECT_AREA
- SELECT_DBD
- SELECT_GROUP
- SELECT_LIMIT

Syntax

SELECT_DATE=parameter1
SELECT_DATE=(parameter1, parameter2)

Parameters

Each parameter is a date/time literal, or a character string that can be converted to a date/time. For details about coding date/time literals, see “Literal” on page 361. If periods (.) are used as the separator character, the date/time literal does not need to be enclosed within quotation marks.

Default

SELECT_DATE=(0, 9999.999.99.99.99.99)

Alias

DATE
**SELECT_DBDB**

**Purpose**

Use the SELECT_DBDB keyword to specify the name of the database(s) to be retrieved from the repository for reporting.

**Use**

SELECT_DBDB is an optional keyword for the following command and subcommands:

- DELETE
- LIST
- MODIFY
- RETRIEVE (as a command or subcommand)

**Related keywords**

- SELECT_AREA
- SELECT_DATE
- SELECT_GROUP
- SELECT_LIMIT

**Syntax**

```
SELECT_DBDB=parameter
```

**Parameters**

Specify a 1-character to 8-character database name. Enclose the mask within either single or double quotation marks if it contains any non-alphanumeric characters.

Wildcard masks using characters * and ? can be included. The asterisk wildcard specifies all parameters for all possible characters to the right of the asterisk. The question mark specifies all parameters for the character in the string in the place the question mark holds.

**Default**

```
SELECT_DBDB=*
```

**Alias**

DBD
SELECT_GROUP

Purpose

Use the SELECT_GROUP keyword to specify the repository group from which to retrieve data for reporting.

Use

SELECT_GROUP is an optional keyword for the following command and subcommands:

- DELETE
- LIST
- MODIFY
- RETRIEVE (as a command or subcommand)

Related keywords

- SELECT_AREA
- SELECT_DATE
- SELECT_DBD
- SELECT_LIMIT

Syntax

SELECT_GROUP=parameter

Parameters

Specify a 1-character to 4-character group name. Enclose the mask within either single or double quotation marks if it contains any non-alphanumeric characters.

Wildcard masks using characters * and ? can be included. The asterisk wildcard specifies all parameters for all possible characters to the right of the asterisk. The question mark specifies all parameters for the character in the string in the place the question mark holds.

Default

SELECT_GROUP=“”

Alias

GROUP
SELECT_LIMIT

Purpose

Use the SELECT_LIMIT keyword to limit the number of most recent repository entries selected per area.

Use

SELECT_LIMIT is an optional keyword for RETRIEVE, which can be used as a command or subcommand depending on the utility program being executed.

Related keywords

- SELECT_AREA
- SELECT_DATE
- SELECT_DBD
- SELECT_GROUP

Syntax

SELECT_LIMIT=number

Parameters

Specify a number between 0 and 256. The default value of 0 will not limit number of entries selected (all entries will be selected).

Default

SELECT_LIMIT=0

Alias

LIMIT

SELECT_UOW

Purpose

Use the SELECT_UOW keyword to specify the technique to be used by the reorganization function in the selection of UOWs to be reorganized. This keyword is also used to allocate and limit the maximum number of buffers used if a maximum value is specified.
If you specify SELECT_UOW=IOVF, only UOWs that extend into IOVF are considered candidates for reorganization. If a UOW does not extend into the IOVF, the UOW is skipped, and any parameter that is specified on the FRAGMENTATION_PERCENT keyword has no effect.

If you specify SELECT_UOW=ALL, all UOWs will be considered as candidate for reorganization.

The product will automatically use SELECT_UOW=ALL under either of the following conditions:

- if COMPRESS=YES or COMPRESS=segment name(s) is specified
- if the area is being extended by the EXTEND_IOVF_#UOWS keyword and an SDEP segment is defined for the database

**Use**

SELECT_UOW is an optional keyword for the REORGANIZE command.

**Related keywords**

- COMPRESS
- EXTEND_IOVF_#UOWS
- FRAGMENTATION_PERCENT

**Syntax**

```plaintext
SELECT_UOW= parameter
```

**Parameters**

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOVF</td>
<td>Consider for reorganization only UOWs that extend into IOVF.</td>
</tr>
<tr>
<td>IOVF,n1</td>
<td>Consider the UOW for reorganization only if it extends into IOVF and uses at least the minimum number of IOVF CIs specified by n1.</td>
</tr>
<tr>
<td>IOVF,n1,n2</td>
<td>Consider the UOW for reorganization only if it extends into IOVF and uses at least the minimum (n1), and no more than the maximum (n2) number of IOVF CIs. Specifying a minimum number (n1) and a maximum number (n2) is optional. If only n2 is specified, use the following syntax: (IOVF,n2).</td>
</tr>
</tbody>
</table>

**Note:** The number of buffers allocated will be limited to the value of the maximum number (n2) specified.

| ALL        | Consider all UOWs for reorganization.                                      |
SET

Default

SELECT_UOW=IOVF

Alias

None

SET

Purpose

Use the SET command to specify an outage window time by using the OUTAGE_WINDOW keyword.

Use

SET is an optional command for the Fast Path/EP operator interface.

Related keyword

OUTAGE_WINDOW

Syntax

SET [keyword=parameter[,...]]

Parameters

None

Default

None

Alias

None
**SHADOW_INIT**

**Purpose**

Use the SHADOW_INIT command to initialize the primary shadow area data set before running the restructure process. Use the PLAN_FILECTL subcommand with the SHADOW_INIT command to specify the name of the Restructure Plan data set, which is used for input into the shadow initialization process.

You can also use the SHADOW_INIT command to dynamically allocate and initialize a secondary shadow area data set before running the restructure process. The secondary shadow data set is a copy of the primary shadow data set, which can be used for backup and recovery purposes.

**Use**

SHADOW_INIT is an optional command for the PFPSYSIN DD statement when you execute the PFCMAIN program. You must have a license for Fast Path Online Restructure/EP to use this command.

**Available keywords**

- ACCESS (required)
- DBD (required)

**Syntax**

```
SHADOW_INIT [keyword=parameter[,...]]
  [subcommand[keyword=parameter[,...]]]
```

**Parameters**

None

**Default**

None

**Alias**

None
SHADOW_SUFFIX

Purpose

Use the SHADOW_SUFFIX keyword to create a name for the primary shadow area data set, which is used by the restructure process.

Use

SHADOW_SUFFIX is an optional keyword for the PREPARE command.

Related keyword

RETAINED_SUFFIX

Syntax

SHADOW_SUFFIX=parameter

Parameters

To create the name of the primary shadow area data set, specify a 1-character to 8-character suffix to be appended to the online area data set name.

NOTE

Because of VSAM restrictions, the name of the shadow area data sets must not exceed 44 characters.

Default

SHADOW_SUFFIX=Z

Alias

None
SHADOW2_DSNAME

Purpose

Use the SHADOW2_DSNAME keyword to specify a mask for a secondary shadow data set name to use for dynamic allocation of the data set to be processed by the command. The secondary shadow data set name produced by the mask (after
variable substitutions are performed) must conform to standard data set naming rules. It cannot contain a reference to a partitioned data set (PDS) member name. It can contain a reference to a generation data set group (GDG), with or without a relative generation number such as “(+1)”.

If DISP=OLD or DISP=SHR is specified, the data set name must refer to an existing catalogued data set. If DISP=NEW is specified, the data set name must not refer to an existing catalogued data set.

If the data set name matches an existing GDG base name, the product will automatically append the appropriate generation information to the base name equivalent to relative generation “(+1)”, if not specified explicitly.

DISP=(NEW,CATLG) is required when a new generation data set is being created.

Use

SHADOW2_DSNAME is an optional keyword for the PREPARE command.

NOTE
Do not specify both the SHADOW2_DSNAME and SHADOW2_SUFFIX keywords; these keywords are mutually exclusive.

Related keywords

- DATACLAS2
- MGMTCLAS2
- STORCLAS2
- VOLSER2

Syntax

SHADOW2_DSNAME='parameter'

Parameters

Specify a 1-character to 64-character data set name mask. Enclose the mask in single or double quotation marks. Use the following variables, as necessary, to create the mask:
The following table provides examples of the resulting data set name that is dynamically allocated based on the specified area variable, and the area name or number of digits in the area number:

<table>
<thead>
<tr>
<th>Area name/number</th>
<th>Specified area name variable</th>
<th>Dynamically allocated data set name</th>
</tr>
</thead>
<tbody>
<tr>
<td>A123</td>
<td>DSN=’PFPTEST.DB.&amp;AREA’</td>
<td>PFP.TEST.DB.A123</td>
</tr>
<tr>
<td>26</td>
<td>DSN=’PFPTEST.DB.PF&amp;AREA#’</td>
<td>PFP.TEST.DB.PF026</td>
</tr>
<tr>
<td>26</td>
<td>DSN=’PFPTEST.DB.PF&amp;AREA4#’</td>
<td>PFP.TEST.DB.PF0026</td>
</tr>
<tr>
<td>26</td>
<td>DSN=’PFPTEST.DB.PF&amp;AREA5#’</td>
<td>PFP.TEST.DB.PF00026</td>
</tr>
<tr>
<td>1024</td>
<td>DSN=’PFPTEST.DB.PF&amp;AREA#’</td>
<td>PFP.TEST.DB.PF1024</td>
</tr>
<tr>
<td>1024</td>
<td>DSN=’PFPTEST.DB.PF&amp;AREA4#’</td>
<td>PFP.TEST.DB.PF1024</td>
</tr>
<tr>
<td>1024</td>
<td>DSN=’PFPTEST.DB.PF&amp;AREA5#’</td>
<td>PFP.TEST.DB.PF1024</td>
</tr>
</tbody>
</table>

Using the &AREA#4 or &AREA5# variable instead of the &AREA# variable lets you standardize the length of dynamically allocated data set names when using 1000 or more areas, while still supporting area numbers 1 through 999.

**Default**

None
SHADOW2_SUFFIX

 Alias

 None

 SHADOW2_SUFFIX

 Purpose

 Use the SHADOW2_SUFFIX keyword to create a name for the optional secondary shadow area data set, which is used by the restructure process.

 Use

 SHADOW2_SUFFIX is an optional keyword for the PREPARE command:

 NOTE

 Do not specify both the SHADOW2_SUFFIX and SHADOW2_DSNAME keywords; these keywords are mutually exclusive.

 Related keyword

 SHADOW2_DSNAME

 Syntax

 \[ \text{SHADOW2_SUFFIX} = \text{parameter} \]

 Parameters

 To create the name of the secondary shadow area data set, specify a 1-character to 8-character suffix to be appended to the original area data set name.

 NOTE

 Because of VSAM restrictions, the name of the secondary shadow area data sets must not exceed 44 characters.

 Default

 None
SHUTDOWN

Purpose

Use the SHUTDOWN command to initiate product shutdown. The shutdown can occur immediately (by using the FORCE keyword) or after an end-of-task.

**NOTE**

BMC Software recommends that you always issue the DISPLAY command before issuing the SHUTDOWN command.

Use

SHUTDOWN is an optional command for the Fast Path/EP operator interface.

Available keywords

- FORCE
- ID (required)
- PRODUCT (required)

Syntax

```
SHUTDOWN [keyword=parameter[, ...]]
```

Parameters

None

Default

None

Alias

None
**SORT**

**Purpose**

When used with the EXTRACT command, the SORT keyword is used to invoke your installation’s Sort Utility and sort the sequential file using the symbolic key of the root segment.

When used with the RELOAD command, the SORT keyword is used to load sequence based on RAP (root anchor point).

When used with the BUILD, RESYNC, VERIFY, and XSCAN commands, the SORT keyword establishes the default value to be used for the SORT keyword on the IX subcommand.

When used with the IX subcommand, the SORT keyword is used to sort the XSCAN data set that is created by the XSCAN utility.

**Use**

SORT is an optional keyword for the following commands:

- BUILD
- EXTRACT
- RELOAD
- RESYNC
- VERIFY
- XSCAN

SORT is an optional keyword for the IX subcommand.

**Related keywords**

SORT_OPTION

**Syntax**

SORT=parameter

**Parameters**

Specify one of the following parameter values:
SORT_NAME

Purpose

Use the SORT_NAME keyword to specify the name of the sort package to use when Fast Path/EP performs an internal sort.

Use

SORT_NAME is an optional keyword for the OPTIONS command.

Syntax

SORT_NAME=parameter

Parameters

Specify a 1-character to 8-character name for the sort package you want to use. Valid values include:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not sort the sequential file.</td>
</tr>
<tr>
<td>YES</td>
<td>Enable the EXTRACT command or the IX subcommand to sort the sequential file.</td>
</tr>
<tr>
<td>AUTO</td>
<td>Enable the RELOAD command to sort the sequential file only if it determines that sorting will improve performance.</td>
</tr>
</tbody>
</table>

**NOTE**

YES functions only with the EXTRACT command and the IX subcommand (not with RELOAD). AUTO functions only with the RELOAD command (not with EXTRACT).

Default

When used with the EXTRACT or RELOAD command, the default is SORT=NO. When used with the IX subcommand, the default is SORT=YES.

Alias

None
Chapter 2 Command language 313

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFSORT</td>
<td>specifies the DFSORT package</td>
</tr>
<tr>
<td>LGBSORT</td>
<td>specifies the LGBSORT package if it is available</td>
</tr>
<tr>
<td>SORT</td>
<td>specifies the sort package to which you have assigned “SORT” as an alias name</td>
</tr>
<tr>
<td>SYNSORT</td>
<td>specifies the SYNSORT package</td>
</tr>
</tbody>
</table>

**Default**

SORT_NAME=LGBSORT

**NOTE**

The product function searches for the default LGBSORT module in the STEPLIB data set concatenation. LGBSORT is located in the hlq.XXLIB (merged installation) or hlq.LGBLIB (non-merged installation) data set. If the additional library containing LGBSORT is not included in the STEPLIB data set concatenation, the product switches and executes SORT_NAME=SORT.

**Alias**

None

**SORT_OPTION**

**Purpose**

Use the SORT_OPTION keyword in conjunction with the SORT keyword to pass optional sort tuning parameters to your installation’s Sort utility. SORT_OPTION should be used only when you want to override dynamic allocation of this information, which will be performed by the product if SORT_OPTION is not specified.

The SORT_OPTION keyword can also be used with the BUILD or VERIFY commands to pass optional sort tuning parameters to the Sort Utility during the creation or verification of an index database.

When used with the XSCAN command, the SORT_OPTION keyword establishes the default value to be used for the SORT_OPTION keyword on the IX subcommand.

In addition, SORT_OPTION can be used with the IX subcommand to pass optional sort tuning parameters to the Sort utility when an optional index information data set (created by the XSCAN utility) is used for input into the build, verify, and resynchronize functions.
Use

SORT OPTION is an optional keyword for the following commands:

- BUILD
- EXTRACT
- PFPSORT
- RELOAD
- VERIFY
- XSCAN

SORT OPTION is an optional keyword for the IX subcommand.

**NOTE**

The SORT OPTION keyword applies to processing in offline or online mode only. It is ignored during BMP processing.

Related keyword

SORT

Syntax

SORT OPTION=parameter
SORT OPTION=parameter1, parameter2, ..., parameterN

Parameters

Numerous parameters are available for use with this keyword to enable you to tailor and control the performance of your installation’s Sort utility. For more information, see the reference manual for the sort product used at your installation.

**NOTE**

If a parameter contains blanks or other special characters, it must be enclosed in single or double quotation marks.

Default

SORT OPTION=DYNALLOC

Alias

None
SORT_SEQUENCE

Purpose

Use the SORT_SEQUENCE keyword to specify the sort sequence to be used to sort an output unload file. Available parameters can be specified on the SORT_SEQUENCE keyword to customize the sorting sequence of root anchor points (RAPs), root key values, and logical SDEPs defined in the unload file.

Use

SORT_SEQUENCE is a required keyword for the PFPSORT command.

Syntax

SORT_SEQUENCE=(RAPparameter, KEYparameter, SDEPSEQ=parameter)
SORT_SEQUENCE=RELATIVE_RAP

Parameters

You must specify at least one of the following four values for either the RAPparameter or the KEYparameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEY_ASCEND</td>
<td>Sort the file in ascending sequence by root key.</td>
</tr>
<tr>
<td>KEY_DESCEND</td>
<td>Sort the file in descending sequence by root key.</td>
</tr>
<tr>
<td>RAP_ASCEND</td>
<td>Sort the file in ascending sequence by RAP.</td>
</tr>
<tr>
<td>RAP_DESCEND</td>
<td>Sort the file in descending sequence by RAP.</td>
</tr>
</tbody>
</table>

If you specify a RAPparameter value and a KEYparameter value, the file will be sorted first by RAP and then by key within the RAP.

The optional SDEPSEQ parameter is valid only when logical SDEPs are defined in the unload file. Otherwise, it is ignored. Specify one of the following values for the SDEPSEQ-parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIFO (default)</td>
<td>Specify that the first SDEP segment inserted by the unload utility will be the first SDEP segment retrieved by IMS using GN processing</td>
</tr>
<tr>
<td>LIFO</td>
<td>Specify that the first SDEP segment inserted by the unload utility will be the last SDEP segment retrieved by IMS using GN processing (the SDEP segments are in reverse order)</td>
</tr>
</tbody>
</table>
RELATIVE_RAP sorts the file by using only the relative RAP. The sort sequence is the same as that provided by the RELOAD command. Use RELATIVE_RAP to process files that are in a file format that is compatible with the IBM HD Reorganization Unload Utility (HDUNLOAD). If you are processing files in a different format, RELATIVE_RAP sorts the file in ascending sequence by RAP.

**Default**

None

**Alias**

SORTSEQ

**SPACE**

**Purpose**

Use the SPACE keyword to specify the space requirements of an output data set created by dynamic allocation.

**Use**

SPACE is an optional keyword for the following subcommands:

- ACTIVITY_FILECTL
- ALLOCATE
- DISCARD_FILECTL
- IC
- OFILECTL
- PLAN_FILECTL

**Related keywords**

- AVGREC
- DSNNAME

**Syntax**

```
SPACE=(units,primary,[secondary],[RLSE],[CONTIG/MXIG/ALX],
[ROUND])
```

In the syntax shown above, you must include all commas that precede any optional parameters specified.
Parameters

Specify one of the following values for the units parameter:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CYL</td>
<td>Allocation unit is in cylinders.</td>
</tr>
<tr>
<td>TRK</td>
<td>Allocation unit is in tracks.</td>
</tr>
<tr>
<td>numeric value from 512 to 32767</td>
<td>Allocation unit is either blocks or records, depending on whether the AVGREC keyword is also specified:</td>
</tr>
<tr>
<td></td>
<td>■ If the allocation unit is in blocks, the specified number indicates the average block size. Do not specify the AVGREC keyword if the unit type is in blocks.</td>
</tr>
<tr>
<td></td>
<td>■ If the allocation unit is in records the number specifies the average record size. Specify the AVGREC keyword to indicate that the allocation unit is in records.</td>
</tr>
</tbody>
</table>

For the primary parameter, specify a numeric value from 1 to 16777215 for primary space for the allocated data set. The value represents the number of allocation units of to be allocated.

For the secondary parameter, specify a numeric value from 0 to 16777215 if additional space is required for the allocated data set. The value represents the number of allocation units of to be allocated.

The RLSE positional parameter is optional. Specify RLSE to request release of unused space.

The CONTIG, MXIG, or ALX positional parameter is optional and refers to data set placement on DASD:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTIG</td>
<td>Space allocated for the data set must be contiguous on the volume. This value applies only to primary space allocation.</td>
</tr>
<tr>
<td>MXIG</td>
<td>Space allocated for the data set is the largest area of available contiguous space on the volume that is equal to or greater than the primary quantity.</td>
</tr>
<tr>
<td>ALX</td>
<td>Space allocated for the data set includes up to five separate contiguous areas on the volume and each area must be equal to or greater than the primary quantity.</td>
</tr>
</tbody>
</table>

The ROUND positional parameter is optional. Specify ROUND to request rounding of the space to complete the number of allocated units.
Default

The default value for the secondary parameter is 0: SPACE=(,,0). Defaults do not apply to any other parameters.

Alias

None

STACK_NAME

Purpose

Use the STACK_NAME keyword to specify a name for a group of output data sets. The data sets in a group are written serially, as multiple files on one or more tape volumes.

When the STACK_NAME keyword is specified, the following rules apply:

- The DDNAME keyword is not allowed.
- The DSNAME keyword is required.
- The UNIT keyword is required, and must refer to a tape device.

When a new (unique) value for the STACK_NAME keyword is encountered, the product constructs a data set stack registry. The values of the following keywords from the same subcommand are saved as part of the stack registry, and apply to the allocation of the stack group:

- EXPTD
- RETPD
- UNIT
- VOLCNT
- VOLSER

The values of these keywords specified on any subsequent usage of the same stack name have no effect. All other keywords on the subcommand refer to the individual output data set.

When the first output data set written to the stack group is opened, the product will dynamically allocate the data set using the registered stack parameters. This data set is written as file sequence number 1. Subsequent output data sets use the same allocation, and are written as file sequence number 2, 3, etc.
Use

STACK_NAME is an optional keyword for the IC subcommand.

Related keywords
- DDNAME
- DISP
- DSNAME
- EXPT
- RETPD
- UNIT
- VOLUM
- VOLSER

Syntax

STACK_NAME=parameter

Parameters

Specify a 1-character to 8-character name.

Default

None

Alias

None

STARTUOW

Purpose

Use the STARTUOW keyword to specify a number that defines the UOW where reporting is to begin.

Use

STARTUOW is an optional keyword for the REPORT subcommand.
STATUS

Related keywords

- DETAIL_LEVEL (optional)
- STOPUOW

**NOTE**
The STARTUOW and STOPUOW keywords cannot be used to generate reports for individual UOWs when statistics are retrieved from the repository by using the RETRIEVE command and area details (DETAIL_LEVEL=AREA) are written to the repository statistics data set.

Syntax

```
STARTUOW=parameter
```

Parameters

Specify a value from 0 to 32766 that is less than or equal to the number of UOWs that are defined in the DBD definitions for the area.

Default

```
STARTUOW=0
```

Alias

None

**STATUS**

**Purpose**

Use the STATUS command to generate the Restructure Activity Report on demand. Use the PLAN_FILECTL subcommand with the STATUS command to specify the name of the Restructure Plan data set (created on the Prepare function), which is used for input into the prepare, shadow initialization process, restructure, and backout processes.

In addition, you use the Restructure Activity data set (created on the Prepare function) as input into the Status function to provide information for the Restructure Activity Report. You can control the allocation of the Restructure Activity data set by using the optional ACTIVITY_FILECTL subcommand with the PREPARE command or allowing the Prepare function to automatically generate it. For more information, see the *Fast Path Online Restructure/EP User Guide*. 
Use

STATUS is an optional command for the PFPSYSIN DD statement when you execute the PFCMAIN program. You must have a license for Fast Path Online Restructure/EP to use this command.

Available keywords

- ACCESS (required)
- DBD (required)
- DETAIL

Syntax

STATUS \[\text{keyword=parameter[,...]}\]\n\[\text{subcommand[keyword=parameter[,...]]}\]

Parameters

None

Default

None

Alias

None

STOPUOW

Purpose

Use the STOPUOW keyword to specify a number that defines the UOW where reporting is to end.

Use

STOPUOW is an optional keyword for the REPORT subcommand.
Related keyword

- DETAIL_LEVEL (optional)
- STARTUOW

**NOTE**
The STOPUOW and STARTUOW keywords cannot be used to generate reports for individual UOWs when statistics are retrieved from the repository by using the RETRIEVE command and area details (DETAIL_LEVEL=AREA) are written to the repository statistics data set.

Syntax

STOPUOW=parameter

Parameters

Specify a value from 0 to 32766 or a value equal to the last UOW defined for the area. The value must be greater than or equal to the value specified by the STARTUOW keyword. If the value specified for STOPUOW is greater than the number of UOWs that are defined in the DBD definition of the area, the STOPUOW value is ignored and reporting continues to the last UOW defined for the area.

Default

The default parameter setting is the number equal to the last UOW in the area.

Alias

None

**STORCLAS**

**Purpose**

Use the STORCLAS keyword to specify the SMS storage class of an output data set created by dynamic allocation.
Use

STORCLAS is an optional keyword for the following subcommands:

- ACTIVITY_FILECTL
- ADD
- ALLOCATE
- DISCARD_FILECTL
- IC
- MODIFY
- OFILECTL
- PLAN_FILECTL

Related keyword

DSNAME

Syntax

STORCLAS=parameter

Parameters

Specify a 1-character to 8-character storage class name.

Default

None

Alias

None

STORCLAS2

Purpose

Use the STORCLAS2 keyword to specify the SMS storage class of a secondary shadow output data set created by dynamic allocation.

Use

STORCLAS2 is an optional keyword for the ALLOCATE command.
Related keyword

- SHADOW2_DSNNAME
- SHADOW2_SUFIX

Syntax

STORCLAS2=parameter

Parameters

Specify a 1-character to 8-character storage class name.

Default

None

Alias

None

SUBSET_POINTERS

Purpose

Use the SUBSET_POINTERS keyword to indicate whether subset pointers should be retained or cleared.

Use

If subset pointers are defined in the DEDB to be processed, then SUBSET_POINTERS is a required keyword for the following commands:

- CHANGE
- RELOAD
- UNLOAD

Syntax

SUBSET_POINTERS=parameter

Parameters

Specify one of the following parameters:
SYNONYM_CHAIN_ANALYSIS

### Purpose

Use the SYNONYM_CHAIN_ANALYSIS keyword to request the Synonym Chain Analysis report. This report shows statistics on the frequency and length of synonym chains and the physical I/O required to access root segments in each of 41 user-defined reporting intervals.

### Use

SYNONYM_CHAIN_ANALYSIS is an optional keyword for the REPORT subcommand.

### Related keywords

- POINTER_VALIDATION
- REPORT_DEFAULT
- SYNONYM_CHAIN_INCREMENT
- RAP_VALIDATION

**NOTE**

This report is produced only if you also specify either POINTER_VALIDATION=FULL or RAP_VALIDATION=XREF.

### Syntax

```plaintext
SYNONYM_CHAIN_ANALYSIS=parameter
```
Parameters

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not produce the report.</td>
</tr>
<tr>
<td>YES</td>
<td>Produce the report.</td>
</tr>
</tbody>
</table>

Default

The default parameter is set by the REPORT_DEFAULT keyword.

Alias

SCA

SYNONYM_CHAIN_INCREMENT

Purpose

Use the SYNONYM_CHAIN_INCREMENT keyword to specify the reporting interval for the Synonym Chain Analysis report.

Use

SYNONYM_CHAIN_INCREMENT is an optional keyword for the REPORT subcommand.

Related keyword

SYNONYM_CHAIN_ANALYSIS

Syntax

SYNONYM_CHAIN_INCREMENT=parameter

Parameters

Specify a value from 1 to 100.

Default

SYNONYM_CHAIN_INCREMENT=1
SYNONYM_CHAIN_LENGTH

Purpose

Use the SYNONYM_CHAIN_LENGTH keyword to specify a threshold setting when analyzing or monitoring DEDB activity. The average number of segments randomized to each RAP block is computed and compared with a user-specified setting. If the average is greater than the user-specified setting, a warning message is issued.

NOTE

If you do not specify the SYNONYM_CHAIN_LENGTH keyword, this threshold test is not performed.

Use

SYNONYM_CHAIN_LENGTH is an optional keyword for the THRESHOLD subcommand.

Related keywords

- POINTER_VALIDATION
- RAP_VALIDATION

NOTE

This threshold test is performed only if you also specify either POINTER_VALIDATION=FULL or RAP_VALIDATION=XREF.

Syntax

SYNONYM_CHAIN_LENGTH=parameter

Parameters

Specify a value from 0 to 32767.

Default

None
SYNONYM_CHAIN_MAXIMUM

Alias

SCL

SYNONYM_CHAIN_MAXIMUM

Purpose

Use the SYNONYM_CHAIN_MAXIMUM keyword to specify a threshold setting analyzing or monitoring DEDB activity. The maximum number of segments randomized to each RAP block is computed and compared with a user-specified setting. If the maximum is greater than the user-specified setting, a warning message is issued.

NOTE

If you do not specify the SYNONYM_CHAIN_MAXIMUM keyword, this threshold test is not performed.

Use

SYNONYM_CHAIN_MAXIMUM is an optional keyword for the THRESHOLD subcommand.

Related keywords

- POINTER_VALIDATION
- RAP_VALIDATION

NOTE

This threshold test is performed only if you also specify either POINTER_VALIDATION=FULL or RAP_VALIDATION=XREF.

Syntax

SYNONYM_CHAIN_MAXIMUM=parameter

Parameters

Specify a value from 0 to 32767.

Default

None
SYNONYM_CHAIN_PERCENT

Purpose

Use the SYNONYM_CHAIN_PERCENT keyword to specify a threshold setting when analyzing or monitoring DEDB activity. RAP blocks synonym chain lengths are compared to the user-specified allowable synonym chain length. If the percentage of those exceeding the user-specified length is greater than the user-specified threshold percentage, a warning message is issued.

NOTE

If you do not specify the SYNONYM_CHAIN_PERCENT keyword, this threshold test is not performed.

Use

SYNONYM_CHAIN_PERCENT is an optional keyword for the THRESHOLD subcommand.

Related keywords

- POINTER_VALIDATION
- RAP_VALIDATION

NOTE

This threshold test is performed only if you also specify either POINTER_VALIDATION=FULL or RAP_VALIDATION=XREF.

Syntax

SYNONYM_CHAIN_PERCENT=(parameter1, parameter2)

Parameters

Specify a value from 0 to 100 for each parameter. parameter1 is the allowable percentage of RAPs with synonym chain lengths that exceed parameter2. parameter2 is the allowable synonym chain length.
THRESHOLD

Default

None

Alias

SCP

THRESHOLD

Purpose

Use the THRESHOLD subcommand to specify threshold parameters for processing by the analysis function.

Use

THRESHOLD is an optional subcommand for the following commands:

- ANALYZE
- CHANGE
- EXTEND
- GLOBAL
- IMAGECOPY
- PREPARE
- RELOAD
- REORGANIZE
Related keywords

- DOVF_FREESPACE_PERCENT
- FREESPACE_DOVF_IOVF
- FREESPACE_RAA_DOVF
- FREESPACE_RAA_IOVF
- IOVF_USED_PERCENT
- IOVF_FREESPACE_PERCENT
- RAA_FREESPACE_PERCENT
- RAP_OVERFLOW_PERCENT
- RECORD_IO_AVERAGE
- RECORD_IO_MAXIMUM
- RECORD_IOVF_PERCENT
- ROOT_IO_AVERAGE
- ROOT_IO_MAXIMUM
- SYNONYM_CHAIN_LENGTH
- SYNONYM_CHAIN_MAXIMUM
- SYNONYM_CHAIN_PERCENT
- UOW_DOVF_PERCENT
- UOW_IOVF_AVERAGE
- UOW_IOVF_MAXIMUM
- UOW_IOVF_PERCENT

Syntax

```
command [keyword=parameter[,...]]
```

```
THRESHOLD [keyword=parameter[,...]]
```

Parameters

None

Default

None

Alias

None
**TIMESTAMP**

**Purpose**

Use the TIMESTAMP keyword to specify whether to generate a time stamp for numbered messages that are written into the PFPPRINT output file.

**Use**

TIMESTAMP is an optional keyword for the OPTIONS command.

**Syntax**

```
TIMESTAMP=parameter
```

**Parameters**

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>Generate a time stamp for each message.</td>
</tr>
<tr>
<td>NO</td>
<td>Do not generate a time stamp for each message.</td>
</tr>
</tbody>
</table>

**Default**

```
TIMESTAMP=YES
```

**Alias**

None

---

**TRACK**

**Purpose**

Use the TRACK keyword to specify the type of Remote Site Recovery (RSR) tracking (shadowing) for an area that is assigned to a global service group.

**Use**

TRACK is an optional keyword for the REGISTER subcommand.
Related keywords

If you specify the TRACK keyword, you must also specify the GSGNAME keyword.

Syntax

```
TRACK=parameter
```

Parameters

Specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB</td>
<td>Use the database-readiness type of tracking.</td>
</tr>
<tr>
<td>RCV</td>
<td>Use the recovery-readiness type of tracking.</td>
</tr>
</tbody>
</table>

Default

If you specify the GSGNAME keyword for the REGISTER subcommand, the default value is DB.

Alias

None

**TYPE_RUN**

Purpose

Use the TYPE_RUN keyword to specify the execution mode for an area extension.

Use

TYPE_RUN is an optional keyword for the EXTEND command.

Syntax

```
TYPE_RUN=parameter
```

Parameters

Specify one of the following parameters:
UNIT

Purpose

Use the UNIT keyword to specify the physical device type to be used for a data set accessed by dynamic allocation. Use the keyword also to specify the number of physical devices to be allocated.

Use

UNIT is an optional keyword for the following subcommands:

- ACTIVITY_FILECTL
- DISCARD_FILECTL
- IC
- MODIFY
- OFILECTL
- PLAN_FILECTL

UNIT is a required keyword for the ADD subcommand.

Related keywords

- DSNAME
- STACK_NAME

Syntax

UNIT=(name[,count])
Parameters

Specify values for the parameters as follows:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Specify a 1-character to 8-character alphanumeric name of the device type to which the image copy is written.</td>
</tr>
<tr>
<td>count</td>
<td>Specify the number of devices. Valid values range from 1 to 59.</td>
</tr>
</tbody>
</table>

Default

The default value for the count parameter is 1: UNIT=(,1)

Alias

None

UNLOAD

Purpose

Use the UNLOAD command to unload in offline mode one or more areas from a DEDB in a file format compatible with the IBM HD ReorganizationUnload utility. The ACB of the database being unloaded is used to describe the areas and is referenced by the IMSACB DD statement. This command can be used to produce an unload file to be processed later by the IBM HD Reorganization Reload Utility to create an HDAM database for BTS testing.

NOTE

Only one UNLOAD command can be executed per job step, and it must be the only command specified in the PFPSYSIN input.

Use

UNLOAD is an optional command for the PFPSYSIN DD statement. You must have a license for Fast Path Reorg/EP or Fast Path Online Reorg/EP to use this command.
Available keywords

- BYPASS_RECORD
- COMPRESS
- DBD (required)
- ERROR_THRESHOLD
- EXPAND
- FORMAT
- IAREA
- ICACHE
- INPUT_DSN_MASK
- INPUT_THREADS
- OAREA
- OUTPUT_DSN_MASK
- SDEP_PROCESS
- SUBSET_POINTERS

Syntax

UNLOAD [keyword=parameter[,...]]
[ subcommand [keyword=parameter [,,...]]]

Parameters

None

Default

None

Alias

None

UOW_DETAILED_ANALYSIS

Purpose

Use the UOW_DETAILED_ANALYSIS keyword to request the UOW Detailed Analysis report. This report shows UOW free space statistics.
Use

UOW_DETAILED_ANALYSIS is an optional keyword for the REPORT subcommand.

Related keyword

REPORT_DEFAULT

Syntax

UOW_DETAILED_ANALYSIS=parameter

Parameters

Specify one of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not produce the report.</td>
</tr>
<tr>
<td>YES</td>
<td>Produce the report.</td>
</tr>
</tbody>
</table>

Default

The default parameter is set by the REPORT_DEFAULT keyword.

Alias

UDA

**UOW_DOVF_PERCENT**

Purpose

Use the UOW_DOVF_PERCENT keyword to specify a threshold setting when analyzing or monitoring DEDB activity. The percentage of UOWs that have a segment in at least one DOVF block is computed and compared with a user-specified threshold. If the computed percentage is greater than the user-specified setting, a warning message is issued.

**NOTE**

If you do not specify the UOW_DOVF_PERCENT keyword, this threshold test is not performed.
UOW_IOVF_AVERAGE

Use

UOW_DOVF_PERCENT is an optional keyword for the THRESHOLD subcommand.

Syntax

UOW_DOVF_PERCENT=parameter

Parameters

Specify a value from 0 to 100.

Default

None

Alias

UDP

UOW_IOVF_AVERAGE

Purpose

Use the UOW_IOVF_AVERAGE keyword to specify a threshold setting when analyzing or monitoring DEDB activity. The average number of IOVF blocks assigned to each UOW is computed and compared with a user-specified setting. If the computed average is greater than the user-specified setting, a warning message is issued.

NOTE

If you do not specify the UOW_IOVF_AVERAGE keyword, this threshold test is not performed.

Use

UOW_IOVF_AVERAGE is an optional keyword for the THRESHOLD subcommand.

Syntax

UOW_IOVF_AVERAGE=parameter
Parameters

Specify a value from 0 to 32767.

Default

None

Alias

UIA

**UOW_IOVF_MAXIMUM**

Purpose

Use the UOW_IOVF_MAXIMUM keyword to specify a threshold setting when analyzing or monitoring DEDB activity. The maximum number of IOVF blocks assigned to each UOW is computed and compared to a user-specified setting. If the computed maximum blocks is greater than the user-specified setting, a warning message is issued.

---

**NOTE**

If you do not specify the UOW_IOVF_MAXIMUM keyword, this threshold test is not performed.

---

Use

UOW_IOVF_MAXIMUM is an optional keyword for the THRESHOLD subcommand.

Syntax

UOW_IOVF_MAXIMUM=parameter

Parameters

Specify a value from 0 to 32767.

Default

None
Alias

UIM

UOW_IOVF_PERCENT

Purpose

Use the UOW_IOVF_PERCENT keyword to specify a threshold setting when analyzing or monitoring DEDB activity. The percentage of UOWs that have one or more IOVF blocks is computed and compared to a user-specified setting. If the computed percentage is greater than the user-specified setting, a warning message is issued.

**NOTE**

If you do not specify the UOW_IOVF_PERCENT keyword, this threshold test is not performed.

Use

UOW_IOVF_PERCENT is an optional keyword for the THRESHOLD subcommand.

Syntax

UOW_IOVF_PERCENT=parameter

Parameters

Specify a value from 0 to 100.

Default

None

Alias

UIP
USER_RECORD

Purpose

Use the USER_RECORD subcommand to specify an output record to be written to an extract file.

NOTE

You must specify EXTRACT_FORMAT=USER when the USER_RECORD subcommand is specified.

Use

USER_RECORD is an optional subcommand for the EXTRACT command.

Available keywords

- BREAK (required)
- FIELDS (required)

Syntax

EXTRACT [keyword=parameter[,...]]
USER_RECORD[keyword=parameter[,...]]

Parameters

None

Default

None

Alias

None
VERIFY

Purpose

Use the VERIFY command to verify one or more index databases against an associated primary DEDB in either the offline or online (BMP) processing mode. You must have a license for the Fast Path Indexer/EP product to use this command.

You can use the VERIFY command with PFX and IBM native indexes if you provide the appropriate password. (For more information about the authentication process, see the Fast Path Indexer/EP User Guide.)

Use

VERIFY is an optional command for the PFPSYSIN DD statement. You must have a license for Fast Path Indexer/EP to use this command.

Available keywords

- CHECKPOINT
- DBD (required)
- IAREA (not valid in BMP mode)
- ICACHE (ignored in BMP mode)
- INDEX_THREADS (ignored in BMP mode)
- INPUT_DSN_MASK (ignored in BMP mode)
- INPUT_THREADS (ignored in BMP mode)
- SORT (ignored in BMC mode)
- SORT_OPTION (ignored in BMP mode)

Syntax

VERIFY [keyword=parameter[,...]]

Parameters

None

Default

None

Alias

None
VOLCNT

Purpose

Use the VOLCNT keyword to specify the maximum number of volumes that will be required by an output data set created by dynamic allocation using nonspecific volumes. This keyword is needed when the number of volumes required will exceed five, unless a list of specific volume serial numbers is supplied by using the VOLSER keyword.

**NOTE**

Do not specify both the VOLCNT and the VOLSER keywords; these keywords are mutually exclusive.

Use

VOLCNT is an optional keyword for the following subcommands:

- ACTIVITY_FILECTL
- DISCARD_FILECTL
- IC
- OFILECTL
- PLAN_FILECTL

Related keywords

- DSNNAME
- VOLSER

Syntax

VOLCNT=parameter

Parameters

Specify a value from 1 to 255.

Default

If the VOLCNT keyword is not specified, the default is the IBM z/OS® default.

Alias

None
VOLSER

Purpose

Use the VOLSER keyword to specify one or more volume-serial identifiers for the data set accessed by dynamic allocation. If the VOLSER keyword is omitted for an output data set, then non-specific (private) volume(s) are requested.

When used in conjunction with the STACK_NAME keyword, the VOLSER keyword specifies a list of tape volumes to which the output data sets are written. While any number of volume-serial identifiers can be listed, only five at a time will be used for any individual data set written to the stack group. If an individual data set requires more than five volumes, the sixth (and subsequent) volumes for the data set will use non-specific (private) tape volumes. When specified on the ALLOCATE subcommand, a list of volume serial-identifiers cannot repeat a volume.

**NOTE**

Do not specify both the VOLSER and the VOLCNT keywords; these keywords are mutually exclusive.

Use

VOLSER is an optional keyword for the following subcommands:

- ACTIVITY_FILECTL
- ADD
- ALLOCATE
- DISCARD_FILECTL
- IC
- MODIFY
- OFILECTL
- PLAN_FILECTL

Related keywords

- DSNAME
- VOLCNT

Syntax

VOLSER=volser
VOLSER=(volser[,volser...])
VOLSER=(*[,*])
Parameters

Specify 1-character to 6-character volume serial numbers. An asterisk (*) specified on the VOLSER keyword has special meaning for SMS processing.

Default

None

Alias

None

VOLSER2

Purpose

Use the VOLSER2 keyword to specify one or more volume-serial identifiers for the secondary shadow output data set accessed by dynamic allocation. If the VOLSER2 keyword is omitted for an output data set, then non-specific (private) volume(s) are requested.

When used in conjunction with the STACK_NAME keyword, the VOLSER2 keyword specifies a list of tape volumes to which the output data sets are written. While any number of volume-serial identifiers can be listed, only five at a time will be used for any individual data set written to the stack group. If an individual data set requires more than five volumes, the sixth (and subsequent) volumes for the data set will use non-specific (private) tape volumes. When specified on the ALLOCATE subcommand, a list of volume serial-identifiers cannot repeat a volume.

**NOTE**

Do not specify both the VOLSER2 and the VOLCNT keywords; these keywords are mutually exclusive.

Use

VOLSER2 is an optional keyword for the following ALLOCATE command.

Related keyword

- SHADOW2_DSNAME
- SHADOW2_SUFFIX
- VOLCNT
VSO

Syntax

VOLSER2=volser
VOLSER2=(volser[,volser...])
VOLSER2=(*,*)

Parameters

Specify 1-character to 6-character volume serial numbers. An asterisk (*) specified on
the VOLSER2 keyword has special meaning for SMS processing.

Default

None

Alias

None

VSO

Purpose

Use the VSO keyword to specify whether an area should reside in virtual storage the
next time the control region is initialized or when the next /START AREA command
is processed.

Use

VSO is an optional keyword for the REGISTER subcommand.

Related keywords

PREOPEN

Syntax

VSO=NO
VSO=(YES, [CFSTR1=parameter,] [CFSTR2=parameter,]
[LKASID=parameter,] [MAS=parameter,] [PRELOAD=parameter])
Parameters

For VSO, to indicate that the area should not reside in virtual storage, specify NO. To indicate that the area should reside in virtual storage, specify YES and a value for the optional CFSTR1, CFSTR2, LKASID, MAS, or PRELOAD parameters.

For CFSTR1, specify the name of the first coupling facility structure for the area. Ensure that you follow the naming conventions for a z/OS coupling facility structure. This parameter is valid only for VSO areas of DEDBs that are defined with SHARELVL2 or SHARELVL3. If the DEDB is SHARELVL2 or SHARELVL3, the area name is the default.

For CFSTR2, specify the name of the second coupling facility structure for the area. Ensure that you follow the naming conventions for a z/OS coupling facility structure. This parameter is valid only for VSO areas of DEDBs that are defined with SHARELVL2 or SHARELVL3.

For LKASID, specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not use local data caching for the specified area for buffer lookaside on read requests.</td>
<td></td>
</tr>
<tr>
<td>YES</td>
<td>Use local data caching for the specified area for buffer lookaside on read requests.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This value is valid only for VSO areas of DEDBs that are defined with SHARELVL2 or SHARELVL3.</td>
<td></td>
</tr>
</tbody>
</table>

For MAS, specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>The coupling facility structure should contain data only for this area.</td>
<td></td>
</tr>
<tr>
<td>YES</td>
<td>A shared VSO should reside in a multi-area coupling facility structure.</td>
<td></td>
</tr>
</tbody>
</table>

For PRELOAD, specify one of the following parameter values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>Do not load a VSO area into a data space or coupling facility structure the next time it is opened.</td>
<td></td>
</tr>
<tr>
<td>YES</td>
<td>Load a VSO area into a data space or coupling facility structure the next time it is opened.</td>
<td></td>
</tr>
</tbody>
</table>
Default

The default value is NO.

Alias

None

WARNING

Purpose

Use the WARNING keyword to specify the job step completion code produced when warning messages are issued. If the highest condition code returned by all tasks in a job indicates a warning, you can have that code translated to any other code of your choosing.

Use

WARNING is an optional keyword for the OPTIONS command.

Related keywords

None

Syntax

WARNING=parameter

Parameters

Specify a value from 0 to 255.

Default

WARNING=4

Alias

None
WHERE

Purpose
Use the WHERE keyword to specify selection criteria for selecting a segment for processing by its associated subcommand.

Use
WHERE is an optional keyword for the following subcommands:
- EXCLUDE
- INCLUDE
- LOADCTL
- OUTPUT

Related keywords
SEGMENT

Syntax
WHERE=(expression)

Parameters
The parameter consists of a Boolean expression that will evaluate to either a true or false result. For details, see Chapter 3, “Expression syntax” in this book.

Default
None

Alias
None

WORK_DATASET

Purpose
Use the WORK_DATASET keyword to specify the options to be used for dynamic allocation of a work data set.
Use

WORK_DATASET is an optional keyword for the OPTIONS command.

Syntax

WORK_DATASET=(AVGREC=parameter, DATACLAS=parameter,
DISP=parameter, EXPDT=parameter, LIKE=parameter,
MGMTCLAS=parameter, RETPD=parameter, SPACE=parameter,
STORCLAS=parameter, UNIT=parameter, VOLCNT=parameter,
VOLSER=parameter)

Parameters

Specify one or more of the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVGREC=parameter</td>
<td>Specify B, U, K, or M:</td>
</tr>
<tr>
<td></td>
<td>▪ B indicates that the allocation units specified by the units subparameter of the SPACE keyword represents an average block size.</td>
</tr>
<tr>
<td></td>
<td>▪ U indicates that the allocation units specified by the units subparameter of the SPACE keyword represents an average record size and the multiplier of the primary and secondary quantities is 1.</td>
</tr>
<tr>
<td></td>
<td>▪ K indicates that the allocation units specified by the units subparameter of the SPACE keyword represents an average record size and the multiplier of the primary and secondary quantities is 1024.</td>
</tr>
<tr>
<td></td>
<td>▪ M indicates that the allocation units specified by the units subparameter of the SPACE keyword represents an average record size, and the multiplier of the primary and secondary quantities is 1048576.</td>
</tr>
<tr>
<td>DATACLAS=parameter</td>
<td>Specify a 1-character to 8-character data class name.</td>
</tr>
</tbody>
</table>
## Command language

### DISP=parameter

Specify a status, normal, or conditional parameter.

**Status parameters are** NEW, OLD, and USE:
- **NEW** creates a new data set.
- **OLD** indicates an existing data set exclusively.
- **USE** indicates conditional allocation. If the data set does not exist, it is created (as if NEW had been specified). If the data set already exists, it is reallocated (as if OLD had been specified). The normal and conditional parameters are also changed from CATLG (if specified) to KEEP.

**Normal and conditional parameters are** DELETE, KEEP, CATLG, and UNCATLG:
- **DELETE** releases the space allocated for the data set.
- **KEEP** maintains the data set on the volume.
- **CATLG** places and entry pointing to the data set in the catalog.
- **UNCATLG** removes the catalog entry to the data set, but retains the data set.

### EXPDT=parameter

Specify a date in either of the following formats: yyddd or yyyy/ddd.

### LIKE=parameter

Specify a 1-character to 64-character data set name or data set name mask.

### MGMTCLAS=parameter

Specify a 1-character to 8-character management class name.

### RETPD=parameter

Specify the number of days to retain the data set. Valid values are from 0 to 32767.
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| SPACE=parameter | Specify a units, primary, secondary, RLSE, data set placement on DASD, or round parameter.  

Units parameters are CYL, TRK, and value from 512 to 3267:

- CYL indicates allocation unit is in cylinders.
- TRK indicates allocation unit is in tracks.
- A value from 512 to 32767 indicates that the allocation unit is either blocks or records, depending on whether the AVGREC keyword is also specified:
  - If the allocation unit is in blocks, the specified number indicates the average block size. Do not specify the AVGREC keyword if the unit type is in blocks.
  - If the allocation unit is in records the number specifies the average record size. Specify the AVGREC keyword to indicate that the allocation unit is in records.

For the primary parameter, specify a numeric value from 1 to 16777215 for primary space for the allocated data set. The value represents the number of allocation units of to be allocated.

For the secondary parameter, specify a numeric value from 0 to 16777215 if additional space is required for the allocated data set. The value represents the number of allocation units of to be allocated.

The next positional parameter is **RLSE**, which is optional. Specify **RLSE** to request release of unused space.

The next positional parameter is optional and refers to data set placement on DASD. Specify one of the following values:

- CONTIG indicates that space allocated for the data set must be contiguous on the volume. This value applies only to primary space allocation.
- MXIG indicates that space allocated for the data set is the largest area of available contiguous space on the volume that is equal to or greater than the primary quantity.
- ALX indicates that space allocated for the data set includes up to five separate contiguous areas on the volume and each area must be equal to or greater than the primary quantity.

Specify the optional **ROUND** parameter to request rounding of the space to complete the number of allocated units.

| STORCLAS=parameter | Specify a 1-character to 8-character storage class name. |
Use the XSCAN command to scan a Fast Path primary DEDB to locate index source segments and then produce an output data set containing index information. You can scan a single area or multiple areas in an existing primary DEDB in either offline mode or online (IFP) mode.

You can use the XSCAN command with PFX and IBM native indexes if you provide the appropriate password. (For more information about the authentication process, see the *Fast Path Indexer/EP User Guide.*

Use

XSCAN is an optional command for the PFPSYSIN DD statement. You must have a license for Fast Path Indexer/EP to use this command.
Available keywords

- DBD (required)
- IAREA (not valid in IFP mode)
- ICACHE
- INPUT_DSN_MASK
- SORT
- SORT_OPTION

Syntax

XSCAN [keyword=parameter[...]]
subcommand [keyword=parameter[...]]

Parameters

None

Default

None

Alias

None
Expression syntax

This chapter provides the syntax rules for coding an expression. An expression can be used within certain keywords available in certain Fast Path/EP products.

This chapter includes the following topics:

Introduction to expressions .................................................. 356
   Using expressions with online and offline data extract .................. 356
   Using expressions with DEDB CHANGE, UNLOAD, and RELOAD ......... 358
   Using expressions with DEDB online restructure .......................... 359
Operands .............................................................................. 360
   Literal .............................................................................. 361
   Field variable ...................................................................... 362
   Built-in variable .................................................................... 363
   Function .............................................................................. 365
   Descriptions of valid functions ................................................ 366
   Assignment variable .................................................................. 369
Data conversions ................................................................. 371
   Data-type ............................................................................ 371
   Size .................................................................................. 371
   Precision ............................................................................. 373
Operators .............................................................................. 373
   Precedence of operators ........................................................ 374
   Assignment operator .............................................................. 375
   Boolean operator ................................................................... 376
   Comparison operator .............................................................. 376
   Numeric operator .................................................................... 377
   Conditional operator .............................................................. 379
Diagnosing problems ............................................................ 379
Introduction to expressions

An expression is defined as a sequence of operands and operators that specify a computation to be performed or condition to be met. An expression can be as simple as a single operand, or can contain a complex sequence of operators and operands necessary to achieve a desired result. An operand can be a self-defining constant value known as a literal as well as any of a broad range of functions or variables, which are defined in detail in this chapter. Operators can be numeric, comparative, Boolean, conditional or assignment – and can be used in various combinations.

Expressions can be coded as parameters on any of four keywords available in Fast Path/EP command language:

- FIELDS
- SEGMENT_RECORD_PREFIX
- SEGMENT_RECORD_SUFFIX
- WHERE

These keywords, in combination with other command language elements, are used to specify segment selection criteria and to customize the output data produced by the product.

Using expressions with online and offline data extract

Expressions can be used as a customization tool with the online or offline data extract process provided by the Fast Path Online Analyzer/EP or Fast Path Analyzer/EP product. Expressions can be used with the WHERE keyword to select the segments to be extracted. Expressions in the FIELDS, SEGMENT_RECORD_PREFIX, and SEGMENT_RECORD_SUFFIX keywords can be used to control both the content and format of the output file.

For more information about using expressions with online data extracts, see the Fast Path Online Suite User Guide. For more information about using expressions with offline data extracts, see the Fast Path Offline Suite User Guide.
Table 14 summarizes the types of expressions that can be used for customizing segment output with online and offline data extracts.

Table 14  Available expressions for customizing segment output for data extracts

<table>
<thead>
<tr>
<th>Keyword used to state expression</th>
<th>Keyword is coded under</th>
<th>Function of expression</th>
<th>Syntax of expression statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHERE</td>
<td>INCLUDE subcommand</td>
<td>To include segment record for extract processing if conditional criteria are met</td>
<td>INCLUDE SEGMENT=segment name, WHERE=(expression)</td>
</tr>
<tr>
<td>WHERE</td>
<td>EXCLUDE subcommand</td>
<td>To exclude segment record from extract processing if conditional criteria are met</td>
<td>EXCLUDE SEGMENT=segment name, WHERE=(expression)</td>
</tr>
<tr>
<td>WHERE</td>
<td>OUTPUT subcommand</td>
<td>To control content of a segment record in extract output by including segment if conditional criteria are met</td>
<td>OUTPUT SEGMENT=segment name, FIELDS=(expression), WHERE= (expression)</td>
</tr>
<tr>
<td>FIELDS</td>
<td>OUTPUT subcommand</td>
<td>To control content of a segment record in extract output</td>
<td>OUTPUT SEGMENT=segment name, FIELDS=(expression)</td>
</tr>
</tbody>
</table>

Table 15 summarizes the types of expressions that can be used only when the USER option is selected for the EXTRACT_FORMAT keyword. These expressions are used to customize the format of the output record for online and offline data extracts.

Table 15  Available expressions for customizing output record for USER format data extracts (part 1 of 2)

<table>
<thead>
<tr>
<th>Keyword used to state expression</th>
<th>Keyword is coded under</th>
<th>Purpose of expression</th>
<th>Syntax of expression statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEGMENT_RECORD_PREFIX</td>
<td>EXTRACT command</td>
<td>To control content of segment record prefix in extract output file</td>
<td>SEGMENT_RECORD_PREFIX=(expression)</td>
</tr>
<tr>
<td>SEGMENT_RECORD SUFFIX</td>
<td>EXTRACT command</td>
<td>To control content of segment record suffix in extract output file</td>
<td>SEGMENT_RECORD SUFFIX=(expression)</td>
</tr>
<tr>
<td>FIELDS</td>
<td>USER_RECORD subcommand with BREAK keyword</td>
<td>Create and control content of a database-level header record</td>
<td>BREAK=(DATABASE,BEFORE), FIELDS=(expression)</td>
</tr>
<tr>
<td>FIELDS</td>
<td>USER_RECORD subcommand with BREAK keyword</td>
<td>Create and control content of a database-level trailer record</td>
<td>BREAK=(DATABASE,AFTER), FIELDS=(expression)</td>
</tr>
</tbody>
</table>
Expressions can be used as a customization tool with processes provided by the Fast Path Reorg/EP product. Expressions can be used with the WHERE keyword to select segments for processing by the specified primary command. For the CHANGE, UNLOAD, and RELOAD functions, expressions in the FIELDS keyword can be used to modify the content and format segments in the output database. For more information, see the Fast Path Offline Suite User Guide.

Table 15 summarizes the types of expressions that can be used under a CHANGE, UNLOAD or RELOAD command.

### Table 15: Available expressions for customizing output record for USER format data extracts (part 2 of 2)

<table>
<thead>
<tr>
<th>Keyword used to state expression</th>
<th>Keyword is coded under</th>
<th>Purpose of expression</th>
<th>Syntax of expression statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIELDS</td>
<td>USER_RECORD</td>
<td>Create and control content of an area-level header record</td>
<td>BREAK=(AREA,BEFORE), FIELDS=(expression)</td>
</tr>
<tr>
<td>FIELDS</td>
<td>USER_RECORD</td>
<td>Create and control content of an area-level trailer record</td>
<td>BREAK=(AREA,AFTER), FIELDS=(expression)</td>
</tr>
</tbody>
</table>

### Using expressions with DEDB CHANGE, UNLOAD, and RELOAD

Expressions can be used as a customization tool with processes provided by the Fast Path Reorg/EP product. Expressions can be used with the WHERE keyword to select segments for processing by the specified primary command. For the CHANGE, UNLOAD, and RELOAD functions, expressions in the FIELDS keyword can be used to modify the content and format segments in the output database. For more information, see the Fast Path Offline Suite User Guide.

Table 16 summarizes the types of expressions that can be used under a CHANGE, UNLOAD or RELOAD command.

### Table 16: Available expressions for customizing DEDB CHANGE, UNLOAD, and RELOAD processes (part 1 of 2)

<table>
<thead>
<tr>
<th>Keyword used to state expression</th>
<th>Keyword is coded under</th>
<th>Purpose of expression</th>
<th>Syntax of expression statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHERE</td>
<td>INCLUDE</td>
<td>To include segment record in an unload output file, a reloaded area, or changed area if conditional criteria are met</td>
<td>INCLUDE SEGMENT=segment name, WHERE=(expression)</td>
</tr>
<tr>
<td>WHERE</td>
<td>EXCLUDE</td>
<td>To exclude segment record from processing if conditional criteria are met</td>
<td>EXCLUDE SEGMENT=segment name, WHERE=(expression)</td>
</tr>
</tbody>
</table>
Expressions can be used as a customization tool with the DEDB online restructure processes provided by the Fast Path Online Restructure/EP product. During a DEDB online restructure, the OUTPUT subcommand and its associated keywords can be used to control output selection and to modify the layout or content of selected segment data.

The required SEGMENT keyword is used to specify the name of the segment to be processed by the OUTPUT subcommand. The following optional keywords can be used to state expressions:

- The WHERE keyword can be used to select segments for processing.
- The FIELDS keyword can be used to modify the content and format segments in the output database during Restructure function processing.

For more information, see the Fast Path Online Restructure/EP User Guide.

Table 17 summarizes the types of expressions that can be used with the PREPARE command.

**Table 16** Available expressions for customizing DEDB CHANGE, UNLOAD, and RELOAD processes (part 2 of 2)

<table>
<thead>
<tr>
<th>Keyword used to state expression</th>
<th>Keyword is coded under</th>
<th>Purpose of expression</th>
<th>Syntax of expression statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHERE</td>
<td>OUTPUT subcommand with FIELDS keyword</td>
<td>To control content of a segment record in unload output file, reloaded area, or changed area by including segment if conditional criteria are met</td>
<td>OUTPUT SEGMENT=segment name, FIELDS=(expression), WHERE=(expression)</td>
</tr>
<tr>
<td>FIELDS</td>
<td>OUTPUT subcommand</td>
<td>To control content of a segment record in unload output file, reloaded area, or changed area</td>
<td>OUTPUT SEGMENT=segment name, FIELDS=(expression)</td>
</tr>
</tbody>
</table>
Operands

You can specify an operand for an expression in one of the following ways:

- literal
- field variable
- built-in variable
- function
- assignment variable

Detailed descriptions of these operands are presented in this section.

In addition to its actual data portion, certain operands can include an optional data-type. A data-type consists of a valid data-type code, optionally preceded by a length declaration. Table 18 shows the valid data-type codes. If a length is not specified with a data-type, then the default length is used.

For literal operands, the default length column in Table 18 does not apply. If the optional length declaration is not specified, the product will use the minimum number of bytes required to store the literal value.

Table 17  Available expressions for customizing DEDB online restructure processes

<table>
<thead>
<tr>
<th>Keyword used to state expression</th>
<th>Keyword is coded under</th>
<th>Purpose of expression</th>
<th>Syntax of expression statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHERE</td>
<td>OUTPUT subcommand with FIELDS keyword</td>
<td>to control the content of a segment record in a restructured area by modifying the segment if conditional criteria are met</td>
<td>OUTPUT SEGMENT=segment name, FIELDS=(expression), WHERE=(expression)</td>
</tr>
<tr>
<td>FIELDS</td>
<td>OUTPUT subcommand</td>
<td>to control the content of a segment record in a restructured area</td>
<td>OUTPUT SEGMENT=segment name, FIELDS=(expression)</td>
</tr>
</tbody>
</table>

Table 18  Valid data-types used with operands (part 1 of 2)

<table>
<thead>
<tr>
<th>Data-type code</th>
<th>Data-type description</th>
<th>Min/max length</th>
<th>Default length</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Fixed-length character string</td>
<td>1-32768</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>Signed fixed-point binarya</td>
<td>1-8</td>
<td>4</td>
</tr>
<tr>
<td>P</td>
<td>Packed decimala</td>
<td>1-16</td>
<td>4</td>
</tr>
</tbody>
</table>
Literal

A literal is a series of one or more alphabetic or numeric characters contained within apostrophes (' ') or quotation marks (" "), as shown in the following examples:

```
'42'
'117.50'
'ABC'
“This literal is comprised of all alphabetic characters.”
```

A literal can also specify an optional data-type and an optional length declaration, which precedes the character string. A literal with a data-type specified is referred to as a typed literal. The default data-type is char(C).

Character string typed literal

For the character string data-type (C), the characters can be any printable EBCDIC character code. An example of a character string typed literal follows:

```
C'ABC'
```
### Numeric typed literal

For numeric data-types (F, P, Z), the string must consist entirely of numeric digits, but can optionally contain a leading sign and/or a decimal point. Examples of a numeric typed literal follow:

```
F'1.0'
4P"100.00"
```

### Hexadecimal typed literal

For the hexadecimal data-type (X), the characters must consist entirely of hexadecimal digits (0-9, A-F). An example of a hexadecimal typed literal follows:

```
X'7F'
```

### Field variable

A **field** variable specifies a physical location of a data item within a database segment. It is not restricted to the locations defined in the DBD by a FIELD statement. A field variable identifies the segment, the column within that segment in which the field begins, and the characteristics of the field. A field variable can be expressed in one of the following ways:

- `column : length`  
- `column : length [ data-type]`  
- `field-name`  
- `field-name [ : length data-type]`

Some examples of field variables are as follows:

```
3:1 (column and length)  
4:* (column and length)  
5:6P (column, length, and data type)  
SEGAKEY (field name and length)  
SEGAKEY:4Z (field name, length, and data-type)
```

A **column** is used to indicate the starting position of the field. A column corresponds to the relative byte within the segment, beginning with column 1. For variable length segments, column 1 refers to the two-byte segment length field. For fixed length segments, column 1 refers to the first data byte. Place a colon (:) following the column number.
A *length* can be specified as a number to indicate a field consisting of a fixed number of bytes. The length can also be specified using an asterisk (*) to indicate a field with a variable number of bytes, consisting of all bytes from the specified column to the end of the database segment. For compilation purposes, the size of this field is computed using the maximum segment length defined in the DBD; at run-time, the size of the field is computed using the actual length of each segment occurrence.

The *data-type* is used to indicate the data format for the field. Table 18 on page 360 shows the valid data-type character codes.

A *field-name* refers to the name given on a FIELD statement within the SEGM statement in the DBD. The position, length, and characteristics for the field variable are obtained from the parameters on the FIELD statement. Specifying a data-type following the *field-name* will override the length and data format obtained from the FIELD statement in the DBD. Place a colon (:) between the *field-name* and *data-type*.

**Built-in variable**

A *built-in* variable is a name that represents a value from data areas provided by the Fast Path/EP product or by IMS, such as the definition of the database, area, or a segment. Such values can be referenced by name and used in an expression as needed.

Built-in variables that define a segment can optionally be qualified by using a segment name, as in the following:

```
[segment-name.] variable
```

An example expression that includes a segment name definition, followed by a built-in variable follows:

```
OUTPUT SEGMENT=SEGB,FIELDS=(SEGA.SEGMENT_CODE,3:*)
```

If no segment qualification is specified, a default segment is implied. The default segment is identified elsewhere, normally using the SEGMENT keyword. So if SEGA was not specified, then SEGB would be used as shown in the above example.

Built-in variables that are available for use within expressions are shown in Table 20.

**NOTE**

The default for these variables is local time.
Table 20  Built-in variables used with expressions

<table>
<thead>
<tr>
<th>Built-in variable name</th>
<th>Data-type</th>
<th>Description</th>
<th>Usage availability/limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Common Variables</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AREA_NAME</td>
<td>8C</td>
<td>Area DDname</td>
<td>available for all expression types</td>
</tr>
<tr>
<td>AREA_NUMBER</td>
<td>1X</td>
<td>Area number</td>
<td>available for all expression types</td>
</tr>
<tr>
<td>DBD AREAS</td>
<td>2F</td>
<td>Number of areas defined in DBD</td>
<td>available for all expression types</td>
</tr>
<tr>
<td>DBD LEVELS</td>
<td>2F</td>
<td>Number of segment levels defined in DBD</td>
<td>available for all expression types</td>
</tr>
<tr>
<td>DBD NAME</td>
<td>8C</td>
<td>DBD member name</td>
<td>available for all expression types</td>
</tr>
<tr>
<td>DBD SEGMENTS</td>
<td>2F</td>
<td>Number of segments defined in DBD</td>
<td>available for all expression types</td>
</tr>
<tr>
<td>RUNDATEa</td>
<td>4X</td>
<td>Date that the program began execution</td>
<td>available for all expression types</td>
</tr>
<tr>
<td>RUNDATETIMEa</td>
<td>12X</td>
<td>Date and time the program began execution</td>
<td>available for all expression types</td>
</tr>
<tr>
<td>RUNTIMEa</td>
<td>4X</td>
<td>Time the program began execution</td>
<td>available for all expression types</td>
</tr>
<tr>
<td>TODAYa</td>
<td>4X</td>
<td>Alias of RUNDATE</td>
<td>available for all expression types</td>
</tr>
<tr>
<td><strong>Segment Definition Variable</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEGMENT CKEY</td>
<td>n1Cb</td>
<td>Concatenated key for current segment</td>
<td>not available for expressions that define:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>database-level header records</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>area-level header records</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>database-level trailer records</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>area-level trailer records</td>
</tr>
<tr>
<td>SEGMENT CODE</td>
<td>1X</td>
<td>Segment code</td>
<td></td>
</tr>
<tr>
<td>SEGMENT LEVEL</td>
<td>1X</td>
<td>Segment hierarchical level</td>
<td></td>
</tr>
<tr>
<td>SEGMENT NAME</td>
<td>8C</td>
<td>Segment name</td>
<td></td>
</tr>
<tr>
<td>SEGMENT LENGTH</td>
<td>2F</td>
<td>Length of segment</td>
<td></td>
</tr>
</tbody>
</table>

a  The DATE_TIME_FORMAT keyword does not apply to this variable.
b  The maximum length for a concatenated key is computed using the DBD definition. The length specified includes a 2-byte prefix containing the current length of the character string (fixed point binary). This format matches the format used by the PL/I programming language.

Table 21 shows the format and an example for certain built-in variables.
A function is a name that represents any of several built-in routines that return a value. Such values can be used in an expression as needed. Function names must be followed by a list of 0 (none) or more parameter values (each of which might be its own expression) enclosed within parentheses:

\[
\text{function-name} ([\text{parameter}[, ...]])
\]

### Table 21 Format and example of RUNDATE, TODAY, RUNTIME and RUNDATETIME built-in variables

<table>
<thead>
<tr>
<th>Built-in variable name</th>
<th>Format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUNDATE</td>
<td>‘yyyyydddS’</td>
<td>X’2001163F’</td>
</tr>
<tr>
<td>TODAY</td>
<td>‘yyyyydddS’</td>
<td>X’2001163F’</td>
</tr>
<tr>
<td>RUNTIME</td>
<td>‘hhmmsssth’</td>
<td>X’17171254’</td>
</tr>
<tr>
<td>RUNDATETIME</td>
<td>‘yyyyydddShhmmssthmijuOFF’</td>
<td>X’200163F1717125400000000’</td>
</tr>
</tbody>
</table>

### Table 22 Functions used with expressions

<table>
<thead>
<tr>
<th>Function name (parameter)</th>
<th>Data-type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS_PACKED</td>
<td>1X</td>
<td>Checks field for packed decimal value</td>
</tr>
<tr>
<td>IS_ZONED</td>
<td>1X</td>
<td>Checks field for zoned decimal value</td>
</tr>
<tr>
<td>LENGTH</td>
<td>2F</td>
<td>Current length of string</td>
</tr>
<tr>
<td>SEGMENT_CODE</td>
<td>1X</td>
<td>Segment code corresponding to segment name</td>
</tr>
<tr>
<td>SEGMENT_LEVEL</td>
<td>1X</td>
<td>Segment level for segment code</td>
</tr>
<tr>
<td>SEGMENT_NAME</td>
<td>8C</td>
<td>Segment name corresponding to segment code</td>
</tr>
<tr>
<td>SEGMENT_PARENT</td>
<td>1X</td>
<td>Segment code for parent of the segment</td>
</tr>
<tr>
<td>SYSDATE</td>
<td>4X</td>
<td>System date</td>
</tr>
<tr>
<td>SYSDATETIME</td>
<td>12X</td>
<td>System date and time</td>
</tr>
<tr>
<td>SYSTIME</td>
<td>8X</td>
<td>System time</td>
</tr>
</tbody>
</table>
Descriptions of valid functions

Valid functions which can be used with expressions are described in this section.

---

**NOTE**
The default for these functions is UTC time.

---

**IS_Packed(field variable)**

The IS_Packed function is used to test whether or not a field variable contains a valid packed decimal value before you attempt to use it. One parameter is required, which identifies the field variable to be tested. The function returns a ‘1’ if the field contains a valid packed decimal value, or returns a ‘0’ if not a packed decimal value.

---

**NOTE**
You perform this test on a character field, rather than a packed decimal field.

Because it returns a Boolean value, the IS_Packed function is useful “as is” when specified on the WHERE keyword as follows:

```plaintext
WHERE=(IS_Packed (16:4C))
```

By using the function with a conditional operator on the FIELDS keyword, you can avoid a data exception on the field and substitute a valid value as follows:

```plaintext
FIELDS=(...,IS_Packed (16:4C)
  THEN 16:4P
  ELSE 4P'0',...)
```

If the field contains a valid packed value, it is referenced directly (THEN 16:4P), but if not, a constant value is used (ELSE 4P’0’).

**IS_Zoned(field variable)**

The IS_Zoned function is used to test whether or not a field variable contains a valid zoned decimal value before you attempt to use it. One parameter is required, which identifies the field variable to be tested. The function returns a ‘1’ if the field contains a valid zoned decimal value, or returns a ‘0’ if not a zoned decimal value.
Because it returns a Boolean value, the IS_ZONED function is useful “as is” when specified on the WHERE keyword as follows:

```
WHERE=(IS_ZONED (16:4C))
```

By using the IS_ZONED function with a conditional operator on the FIELDS keyword, you can avoid a data exception on the field and substitute a valid value as follows:

```
FIELDS=(IS_ZONED (16:4C)
  THEN 16:4Z
  ELSE 4Z'0',...)
```

If the field contains a valid zoned value, it is referenced directly (THEN 16:4Z), but if not, a constant value is used (ELSE 4Z'0').

**LENGTH(field variable or literal)**

The LENGTH function returns the length in bytes of a field variable or literal. One parameter is required, which identifies the value to be examined. The maximum size of the field is returned.

If all of the parameters of a LENGTH function are literal values (or literal expressions), then the function is evaluated immediately when encountered in the command input. The result is treated as a literal value and can be used anywhere that a simple literal value can be used (on any keyword). For example, the literal function LENGTH ("ABC") is the same as 3.

**SEGMENT_CODE(segment-name)**

The SEGMENT_CODE function returns the segment code associated with a segment definition. A character string identifying the segment definition by name is the only optional parameter that can be specified. If this parameter is omitted, the segment code for the current (default) segment occurrence is returned. Examples of the SEGMENT_CODE function follow:

```
OUTPUT SEGMENT=SEGA,FIELDS=(SEGMENT_CODE(),3:*)
OUTPUT SEGMENT=SEGB,
  FIELDS=(SEGMENT_CODE(SEGA),'SEGB',3:*)
```
**SEGMENT_LEVEL(segment-code)**

The SEGMENT_LEVEL function returns the hierarchical level associated with a segment definition. One parameter can be supplied optionally: a number identifying the segment definition by its segment code. If the parameter is omitted, the hierarchical level for the current (default) segment occurrence is returned.

**SEGMENT_NAME(segment-code)**

The SEGMENT_NAME function returns the name associated with a segment definition. One parameter can be supplied optionally: a number identifying the segment definition by its segment code. If the parameter is omitted, the name of the current (default) segment occurrence is returned.

**SEGMENT_PARENT(segment-code)**

The SEGMENT_PARENT function returns the segment code associated with the hierarchical parent of a segment. One parameter can be supplied optionally: a number identifying the segment definition by its segment code. If the parameter is omitted, the segment code of the parent of the current (default) segment occurrence is returned. The parent of the root segment is defined to be segment code ‘0’.

**SYSDATE ( )**

The SYSDATE function returns a date value (4X). No parameters exist for this function; however, the ( ) must be specified.

The format and an example of the SYSDATE function are the same as those shown for the RUNDATE built-in variable shown in Table 21 on page 365.

**SYSDATETIME ( )**

The SYSDATETIME function returns a date and time value (12X). No parameters exist for this function; however, the ( ) must be specified.

The format and an example of the SYSDATETIME function are the same as those shown for the RUNDATETIME built-in variable shown in Table 21 on page 365.
**SYSTIME ( )**

The SYSTIME function returns a time of day value (8X). No parameters exist for this function; however, the ( ) must be specified. The format is ‘hhmmssthmijuOFF’. An example of the SYSTIME function follows:

X'213722835376020D'

---

**NOTE**

The DATE_TIME_FORMAT keyword does not apply to the following functions:

- SYSDATE()
- SYSDATETIME()
- SYSTIME()

---

**Assignment variable**

For the FIELDS keyword, an assignment variable can be specified in either of the following ways:

: length and <data-type>
: <string operand>

---

**NOTE**

Use of an assignment variable is not allowed within an expression specified using the WHERE keyword.

An assignment variable expressed as :length and <data-type> is useful to force the conversion of a field from one data-type, size and/or precision to another. The expression shown in the following example will cause the field beginning in column 14 (a 4-byte variable in zoned decimal format) to be converted into a 6-byte packed decimal field, with two decimal digits added to the right.

: 6.2P := 14:4Z

An assignment variable is also useful to force the result of a computation into a particular data-type, size and/or precision. For example, the characteristics of the value resulting from the expression shown in the following example will be 3.0P. (The first operand is converted to a 3-byte packed decimal value, and the second is converted to a 1-byte packed value. No decimal alignment is required. The size of the result will be the size of the larger operand).
If it is desired that the result be returned to zoned decimal format, a conversion such as the following example could be used.

\[
:3^p := 14:4Z + 1
\]

An assignment variable expressed as :<string operand> is useful for converting a date or time value into an EBCDIC character string. The string operand can contain substitution masks as shown in Table 23. Any character string that does not match a substitution mask will appear unmodified in the result. The Example column in Table 23 shows the value generated if the data/time source value is Monday, January 18, 1999 at 17:42:16.743.205 Central Standard Time.

<table>
<thead>
<tr>
<th>Mask</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>%YYYY%</td>
<td>4-digit year (e.g., 1999)</td>
<td>1999</td>
</tr>
<tr>
<td>%YY%</td>
<td>2-digit year</td>
<td>99</td>
</tr>
<tr>
<td>%MO%</td>
<td>month of year</td>
<td>01</td>
</tr>
<tr>
<td>%MONTH%</td>
<td>month of year</td>
<td>January</td>
</tr>
<tr>
<td>%MON%</td>
<td>abbreviated month</td>
<td>JAN</td>
</tr>
<tr>
<td>%DDD%</td>
<td>day of year</td>
<td>018</td>
</tr>
<tr>
<td>%DD%</td>
<td>day of month</td>
<td>18</td>
</tr>
<tr>
<td>%WEEKDAY%</td>
<td>day of the week</td>
<td>Monday</td>
</tr>
<tr>
<td>%DAY%</td>
<td>abbreviated day of the week</td>
<td>MON</td>
</tr>
<tr>
<td>%24%</td>
<td>hour (military: 00-23)</td>
<td>17</td>
</tr>
<tr>
<td>%HH%</td>
<td>hour (standard: 01-12)</td>
<td>05</td>
</tr>
<tr>
<td>%AM%</td>
<td>AM or PM</td>
<td>PM</td>
</tr>
<tr>
<td>%MI%</td>
<td>minute (00-59)</td>
<td>42</td>
</tr>
<tr>
<td>%SS%</td>
<td>second (00-59)</td>
<td>16</td>
</tr>
<tr>
<td>%T%</td>
<td>1/10 of second</td>
<td>7</td>
</tr>
<tr>
<td>%TH%</td>
<td>1/100 of second</td>
<td>74</td>
</tr>
<tr>
<td>%THM%</td>
<td>1/1000 of second (millisecond)</td>
<td>743</td>
</tr>
<tr>
<td>%I%</td>
<td>1/10 of millisecond</td>
<td>2</td>
</tr>
<tr>
<td>%II%</td>
<td>1/100 of millisecond</td>
<td>20</td>
</tr>
<tr>
<td>%IJU%</td>
<td>1/1000 of millisecond (microsecond)</td>
<td>205</td>
</tr>
<tr>
<td>%OFF%</td>
<td>time zone offset from UDT</td>
<td>–06:00</td>
</tr>
<tr>
<td>%LOCAL%</td>
<td>display as local time</td>
<td>17:42:16</td>
</tr>
<tr>
<td>%UTC%</td>
<td>display as UTC</td>
<td>23:42:16</td>
</tr>
</tbody>
</table>
Data conversions

An example of a string operand containing a mask for conversion follows:

```sql
: 'YYYY/MM/DD' := SYSDATE()
```

This conversion mask would result in an 8-byte character string containing the formatted system date 1999/01/18.

Data conversions

Understanding the data conversion that can be performed on an expression is necessary for achieving desired results in your output file. BMC recommends that you make both the operands the same data-type. When necessary, a data conversion will be performed automatically for an operand in order to modify its data-type, size and/or precision.

Data-type

Type conversion is the process of changing the general data-type of an operand from one form to another. For example, an operand in zoned decimal type (Z) will be converted into packed decimal type (P) in order to perform a numeric operation such as addition. Changing an operand from one format to another does not change its logical value.

Size

When an operand is converted from one data-type to another, it is normal for its size (length in bytes) to change. As a general rule, the size of the converted target operand will be the minimum size necessary to hold the value from original (source) operand. For example, when converting an operand from 5Z (5-byte zoned decimal) into packed decimal, the converted target value will be 3P (3-byte packed decimal).

Increasing the size of an operand without changing its data-type will not cause any data loss. For numeric operands (data-type F, P, or Z), this process appends leading zeroes as necessary; the value and significance is not affected. For fixed-length string (C) operands, this process appends trailing blanks as necessary.
Decreasing the size of an operand without changing its data type might cause a loss of data. For numeric operands, the process is performed by removing leading zeroes as necessary; an overflow exception is recognized if the value is too large for the reduced size. For character string operands (data-type ‘C’), the process is performed by removing trailing characters as necessary. If non-blank characters are removed, the value of the converted operand is not the same as the original.

Table 24 on page 372 provides detailed information about the size of an operand when converting from one type to another.

### Table 24  Size of operand after conversion

<table>
<thead>
<tr>
<th>Target data-type / Source data-type</th>
<th>C</th>
<th>F</th>
<th>P</th>
<th>X</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>—a</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>noneb</td>
</tr>
<tr>
<td>C</td>
<td>—a</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>noneb</td>
</tr>
<tr>
<td>F</td>
<td>—a</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>noneb</td>
</tr>
<tr>
<td>P</td>
<td>—a</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>noneb</td>
</tr>
<tr>
<td>X</td>
<td>—a</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>noneb</td>
</tr>
<tr>
<td>Z</td>
<td>—a</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>noneb</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source Size</th>
<th>Target Size</th>
<th>Source Size</th>
<th>Target Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3-5</td>
<td>4</td>
<td>3-5</td>
<td>4</td>
</tr>
<tr>
<td>6+</td>
<td>8</td>
<td>6+</td>
<td>8</td>
</tr>
</tbody>
</table>

- a Denotes same data-type; no conversion exposure exists.
- b Invalid conversion.
- c This conversion cannot occur unless the length of the resulting target character string is specified explicitly.
- d Length of source operand.
- e Intermediate conversion to type P is performed.

Table 25 refers to cells in the ‘P’ column of Table 24 above where the size of the converted target data is dependent of the size of the source.

### Table 25  Size of target (source type=F or X; target type=P)

<table>
<thead>
<tr>
<th>Size of source</th>
<th>Size of target</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
</tr>
</tbody>
</table>
Precision

Precision is defined as the number of significant decimal digits to the right of an assumed decimal point. In general, the precision is unaffected by changes to the data-type or size of an operand. However, the precision of an operand will be changed under certain conditions. Increasing the precision of an operand has no effect on its value; in general, the size of the operand must also be increased simultaneously. Decreasing the precision of an operand might cause a loss of precise value as fractional digits are eliminated, because the product will round the value of the operand when its precision is reduced.

For example, increasing the precision has no effect in the following example:

\[
5.4Z := 6.4.2Z
\]

because 000456.1800 is equal to 0456.18.

Decreasing the precision (rounding) could cause loss of precise value, as shown in the following example:

\[
4.2Z := 6.5.4Z
\]

where rounding .1783 to .18 might not be acceptable.

Operators

An operator is used to specify the way in which two operands are to be combined or compared. All valid operators that can be used in an expression are described in Table 26.

### Table 26 Size of target (source type=F or X; target type=P)

<table>
<thead>
<tr>
<th>Size of source</th>
<th>Size of target</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
</tr>
</tbody>
</table>
Table 26  Operators used with expressions

<table>
<thead>
<tr>
<th>Operation and operator</th>
<th>Action</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Assignment Operator</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>operand2 := operand1</td>
<td>Assignment</td>
<td>operand1</td>
</tr>
<tr>
<td><strong>Boolean Operators</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>operand1 AND operand2</td>
<td>Logical AND</td>
<td>Boolean</td>
</tr>
<tr>
<td>operand1 OR operand2</td>
<td>Logical OR</td>
<td>Boolean</td>
</tr>
<tr>
<td>NOT operand1</td>
<td>Logical negation</td>
<td>Boolean</td>
</tr>
<tr>
<td><strong>Comparison Operators</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>operand1 EQ operand2</td>
<td>Equal to</td>
<td>Boolean</td>
</tr>
<tr>
<td>operand1 NE operand2</td>
<td>Not equal to</td>
<td>Boolean</td>
</tr>
<tr>
<td>operand1 GT operand2</td>
<td>Greater than</td>
<td>Boolean</td>
</tr>
<tr>
<td>operand1 LT operand2</td>
<td>Less than</td>
<td>Boolean</td>
</tr>
<tr>
<td>operand1 GE operand2</td>
<td>Greater than or equal to</td>
<td>Boolean</td>
</tr>
<tr>
<td>operand1 LE operand2</td>
<td>Less than or equal to</td>
<td>Boolean</td>
</tr>
<tr>
<td>operand1 == operand2</td>
<td>Equal to</td>
<td>Boolean</td>
</tr>
<tr>
<td>operand1 &lt;&gt; operand2</td>
<td>Not equal to</td>
<td>Boolean</td>
</tr>
<tr>
<td>operand1 &gt; operand2</td>
<td>Greater than</td>
<td>Boolean</td>
</tr>
<tr>
<td>operand1 &lt; operand2</td>
<td>Less than</td>
<td>Boolean</td>
</tr>
<tr>
<td>operand1 &gt;= operand2</td>
<td>Greater than or equal to</td>
<td>Boolean</td>
</tr>
<tr>
<td>operand1 &lt;= operand2</td>
<td>Less than or equal to</td>
<td>Boolean</td>
</tr>
<tr>
<td><strong>Numeric Operators</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>operand1 + operand2</td>
<td>Addition</td>
<td>Sum</td>
</tr>
<tr>
<td>operand1 – operand2</td>
<td>Subtraction</td>
<td>Difference</td>
</tr>
<tr>
<td>operand1 * operand2</td>
<td>Multiplication</td>
<td>Product</td>
</tr>
<tr>
<td>operand1 / operand2</td>
<td>Division</td>
<td>Quotient</td>
</tr>
<tr>
<td><strong>Conditional Operator</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>operand1 THEN operand2</td>
<td>Conditional selection</td>
<td>operand2 or operand3</td>
</tr>
<tr>
<td>ELSE operand3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Precedence of operators**

When an expression contains more than one operator, the operations are performed based on the order of precedence:

1. * / (numeric)
2. + – (numeric)
3. comparison
4. Boolean NOT
5. Boolean AND
6. Boolean OR
7. conditional
8. assignment (lowest precedence)

A sequence of operators with equal precedence are performed in a left-to-right sequence. Since operators and operands enclosed in parentheses are always performed first, parentheses can be used to change the order of operations. For example, in the expression that follows, the field in column 18 is first multiplied by 2, and then added with the field in column 14.

\[ 14:4Z + 2 \times 18:2P \]

By adding the parentheses in the expression in the following example, the field in column 14 is added with 2, and then the sum is multiplied by the field in column 18.

\[ (14:4Z + 2) \times 18:2P \]

**Assignment operator**

An assignment operator is used to replace the value of an existing variable (operand1) with a new value (operand2). The use of an assignment operator is restricted to expressions used within the FIELDS keyword only.

**NOTE**

Use of an assignment operator is not allowed within an expression specified using the WHERE keyword.

The assignment operator (:=) is used to perform an explicit conversion from one data-type to another. The value of the source operand (operand1) is converted into the data-type specified by the target variable (operand2). The rules for data conversion are provided earlier in this chapter.
**Boolean operator**

The operands for a Boolean operator must be of a Boolean type. In other words, each operand must be of a type that can be evaluated by a Boolean operator. If either operand is not of a Boolean type, it will be converted to Boolean.

The expression shown in the following example performs a compound comparison on a data field to test a range.

```
WHERE=(10:1 >= 'A' AND 10:1 <= 'Z')
```

**Comparison operator**

The operands for a comparison operator might require data conversion before the comparison is performed. The two operands must be in a common (or compatible) format.

When comparing character string literals (data-type C), the comparison is performed from left to right. The shorter string is considered to be padded on the right with blanks.

For comparison of numeric operands (F, P and X data-types), the product will ensure that the implied decimal points of the two operands are aligned. If the operands do not have the same precision, the operand with the smaller precision will have its precision increased to match the other operand.

For example, the expression that follows compares two data fields:

```
WHERE=(42:4F > 46:2F)
```

If the result is the comparison yields a “true” result, then the WHERE statement will be processed as part of the command set.

Table 27 shows the data format that will be selected by the product when a comparison operator is used with the various possible combinations of operand types. One (or both) operands will be converted into the selected type before performing the comparison.
**Numeric operator**

The operands for a numeric operator might require data conversion before the operation is performed. With few exceptions, the two operands must be in a common format. Table 27 shows the data-type that will be selected when a numeric operator is used with the various possible combinations of operand types. One (or both) operands will be converted into the selected data-type before performing the operation.

**Table 27** Target data-type for operations with mixed operands (used with comparison operator or numeric operator)

<table>
<thead>
<tr>
<th>Operand2 / Operand1</th>
<th>C</th>
<th>F</th>
<th>P</th>
<th>X</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>C (for comparison operator)</td>
<td>none&lt;sup&gt;a&lt;/sup&gt;</td>
<td>none&lt;sup&gt;a&lt;/sup&gt;</td>
<td>none&lt;sup&gt;a&lt;/sup&gt;</td>
<td>none&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>F</td>
<td>none</td>
<td>F</td>
<td>P</td>
<td>F</td>
<td>P</td>
</tr>
<tr>
<td>P</td>
<td>none</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>X</td>
<td>none</td>
<td>F</td>
<td>P</td>
<td>X</td>
<td>P</td>
</tr>
<tr>
<td>Z</td>
<td>none</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
</tbody>
</table>

<sup>a</sup> Invalid combination of operands.

**Addition and subtraction**

The addition operator (+) is used to compute the sum of the two operands. The subtraction operator (–) is used to compute the difference between the two operands.

The product will ensure that the implied decimal points of the two operands are aligned. If both operands do not have the same precision, the operand with the smaller precision will have its precision increased to match the other operand. If operand1 is smaller in size than operand2, the size of operand1 is increased to match that of operand2.

The data-type, size and precision of the value that results from the operation will match that of operand1 (after all necessary data conversions have been performed).

For example, the expression that follows causes two data fields to be added. The second operand (type F) will be converted to packed, and the decimal is aligned with the first operand.

```
SEGMENT_RECORD_PREFIX=(SEGMENT_NAME,16:4.1P + 20:2F)
```


**Multiplication**

The multiplication operator (*) is used to compute the product of the two operands.

For multiplication of fixed-point operands, there is a limit to the size of the operands. Binary operands cannot exceed 4 bytes; packed operands cannot exceed 8 bytes. If an operand exceeds the limit, the product will reduce its size; it will attempt to do so by first reducing its precision, if possible. The sum of precisions of the two operands cannot exceed 15. If it does, the product will reduce the precision of the operand with the larger precision.

The data format of the value that results from the operation matches that selected for the operands. Its size is the sum of the sizes of the two operands, but not larger than the maximum size for the data format. Its precision is the sum of the precisions of the two operands.

For example, the expression that follows multiplies a data field by a constant. The constant is encoded internally as 2P. The result of the operation is 6.2P.

```
WHERE=(SEGKEY EQ (18:4.2P * 100))
```

**Division**

The division operator (/) is used to compute the quotient of the two operands. The quotient is not rounded; any remainder from the division is discarded.

For division of fixed-point operands, there is a limit to the size of the divisor. Binary operands cannot exceed 4 bytes; packed operands cannot exceed 8 bytes. If the divisor exceeds the limit, the product will reduce its size; it will attempt to do so by first reducing its precision, if possible. The precision of operand1 cannot be less than the precision of operand2. If so, the product will increase the precision of operand1 to match that of operand2.

After all other required data conversions have been completed, the size of operand1 will be increased to match the size of operand1 plus the size of operand2, but not larger than the maximum size for the data format. This intermediate size is also used to determine the size of the result, as explained in the following paragraph.

The data format of the value of the result from the operation matches that selected for the operands. Its size is the size of the intermediate for operand1 (as described in the preceding paragraph) minus the size of operand2. Its precision is the precision of operand1 minus the precision of operand2.

For example, the expression that follows divides a data field by a constant. The constant is encoded internally as 2P, yielding an unrounded 6P result with no fractional digits.
To retain two fractional digits, add 0.00 to the dividend before dividing, as shown in the following example:

\[
\frac{22.6P}{100}
\]

(22.6P + 0.00) / 100

**Conditional operator**

The first operand (operand1) for a conditional operator must be of a Boolean type. In other words, the operand1 must be of a type that can be evaluated by a Boolean operator. If operand1 is not of a Boolean type, it will be converted to Boolean.

If the value of operand1 is “true”, the result of the conditional operation is the value of operand2. Operand3 is not evaluated.

If the value of operand1 is “false”, the result of the conditional operation is the value of operand3. Operand2 is not evaluated.

For example, the expression that follows performs a conditional test and selects either a data field or a constant depending on the test.

```
FIELDS=(3:13, IS_PACKED(16:4C) THEN 16:4P ELSE 4P’0’, 19:*)
```

**Diagnosing problems**

See Appendix A of the *Fast Path Offline Suite User Guide* or the *Fast Path Online Suite User Guide* for detailed information on diagnosing problems with expressions.
**DEDB reports**

The Fast Path/EP products produce detailed reports on all aspects of database space and condition. This chapter provides a general overview of the DEDB analysis, unload/reload, and extend reports, along with descriptions of the fields contained within each report.

This chapter discusses the following topics:

- DEDB analysis reports
  - Area Summary Report
  - Free Space Analysis Report
  - IOVF Space Analysis Report
  - UOW Detailed Analysis Report
  - Pointer Analysis Report
  - Segment I/O Analysis Report
  - Segment Length Analysis Report
  - Segment Placement Analysis Report
  - Record Length Analysis Report
  - Record Placement Analysis Report
  - Record Profile Analysis Report
  - Synonym Chain Analysis Report

- DEDB Unload Reports
  - Unload Input Area Summary Report
  - Unload Output Area Summary Report
  - Unload Database Summary Report

- DEDB Reload Reports
  - Reload Input Area Summary Report
  - Reload Output Area Summary Report
  - Reload Database Summary Report

- DEDB Extend Report
  - Extend Area Summary Report
DEDB analysis reports

Fast Path Analyzer/EP produces a total of 11 reports. The Area Summary Report is the most general of the reports, and is generated automatically any time the Fast Path Analyzer/EP ANALYZE command is executed. The other 10 reports can be categorized in three general groups. These reports contain detailed data about database performance, space usage, segment characteristics, and record characteristics.

As shown in Table 28, a specific keyword is available to request each report individually when executing the Fast Path Analyzer/EP REPORT subcommand. Table 28 also shows that certain reports require specific keyword values for generation.

For additional examples of the reports, refer to PFPANLYZ in the REPORTS data set.

Table 28  Fast Path Analyzer/EP reports and generation keywords

<table>
<thead>
<tr>
<th>Category</th>
<th>Report title</th>
<th>Keyword that generates report</th>
<th>Prerequisite</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space Utilization</td>
<td>Free Space Analysis</td>
<td>FREESPACE_ANALYSIS</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>IOVF Space Analysis</td>
<td>IOVF_SPACE_ANALYSIS</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>UOW Detailed Analysis</td>
<td>UOW_DETAILED_ANALYSIS</td>
<td>none</td>
</tr>
<tr>
<td>Segment</td>
<td>Pointer Analysis</td>
<td>POINTER_ANALYSIS</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>Segment I/O Analysis</td>
<td>SEGMENT_IO_ANALYSIS</td>
<td>POINTER_VALIDATION=FULL must be specified</td>
</tr>
<tr>
<td></td>
<td>Segment Length Analysis</td>
<td>SEGMENT_LENGTH_ANALYSIS</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>Segment Placement Analysis</td>
<td>SEGMENT_PLACEMENT_ANALYSIS</td>
<td>none</td>
</tr>
<tr>
<td>Record</td>
<td>Record Length Analysis</td>
<td>RECORD_LENGTH_ANALYSIS</td>
<td>POINTER_VALIDATION=FULL must be specified</td>
</tr>
<tr>
<td></td>
<td>Record Placement Analysis</td>
<td>RECORD-placement_ANALYSIS</td>
<td>POINTER_VALIDATION=FULL must be specified</td>
</tr>
<tr>
<td></td>
<td>Record Profile Analysis</td>
<td>RECORD_PROFILE_ANALYSIS</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>Synonym Chain Analysis</td>
<td>SYNONYM_CHAIN_ANALYSIS</td>
<td>POINTER_VALIDATION=FULL or RAP_VALIDATION=X REF must be specified</td>
</tr>
</tbody>
</table>
Area Summary Report

The Area Summary Report (Figure 14) provides basic information about each area that has been analyzed. It is generated automatically with each report set and precedes all other reports that are presented. Any threshold exception conditions are shown on the report. For an additional example of the Area Summary Report, see the PFPANLYZ member in the REPORTS data set.

Table 29 lists the fields that are displayed in this report.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD Name</td>
<td>The database description (DBD) name of the database.</td>
</tr>
<tr>
<td>Area ddname</td>
<td>The ddname of the area.</td>
</tr>
<tr>
<td>Area dsname</td>
<td>The dsname of the area.</td>
</tr>
<tr>
<td>Area Number</td>
<td>The area number.</td>
</tr>
<tr>
<td>Randomizing Module Name</td>
<td>The value that is defined in the DBD.</td>
</tr>
<tr>
<td>UOW</td>
<td>The first value is the number of blocks in each unit of work (UOW); the second value is the number of dependent overflow (DOVF) blocks in each UOW, as defined in the DBD.</td>
</tr>
</tbody>
</table>
Area Summary Report

Table 29  Area Summary Report fields (part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROOT</td>
<td>The first value is the number of UOWs in the area; the second value is the number of overflow UOWs in the area, as defined in the DBD.</td>
</tr>
<tr>
<td>CISIZE</td>
<td>The Control Interval Size (CISIZE) value that is defined in the DBD.</td>
</tr>
</tbody>
</table>

Root Addressable Area Portion

This part of the report shows data concerning the root addressable (RAA) portion of the area.

Table 30  RRA fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOWs in RAA</td>
<td>The number of UOWs in the root addressable part of the area.</td>
</tr>
<tr>
<td>Total CIs per UOW</td>
<td>The number of CIs for each UOW.</td>
</tr>
<tr>
<td>RAP CIs per UOW</td>
<td>The number of root anchor point (RAP) CIs for each UOW.</td>
</tr>
<tr>
<td>DOVF CIs per UOW</td>
<td>The number of DOVF CIs in each UOW.</td>
</tr>
<tr>
<td>Total Root Anchor Points</td>
<td>The number of root addressable blocks in the area.</td>
</tr>
<tr>
<td>Total Dependent Overflow CIs</td>
<td>The number of DOVF CIs in the area.</td>
</tr>
</tbody>
</table>

Independent Overflow Portion

This part of the report shows data concerning the independent overflow (IOVF) portion of the area.

Table 31  IOVF fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Independent Overflow CIs</td>
<td>The number of independent overflow (IOVF) CIs in the area.</td>
</tr>
<tr>
<td>Space Map CIs</td>
<td>The number of IOVF CIs that are used to map free space.</td>
</tr>
<tr>
<td>IOVF Data CIs</td>
<td>The number of IOVF CIs that are usable for data storage.</td>
</tr>
</tbody>
</table>
Sequential Dependent

This part of the report shows data concerning the sequential dependent (SDEP) portion of the area.

**Table 32  SDEP portion fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Sequential Dependent CIs</td>
<td>The number of CIs that are available for sequential dependents.</td>
</tr>
<tr>
<td>SDEP Logical Begin</td>
<td>The location of the first sequential dependent (SDEP) segment occurrence in the area. The first value is the cycle count, and the second value is the relative byte address (RBA).</td>
</tr>
<tr>
<td>SDEP Logical End</td>
<td>The location of the last SDEP segment occurrence in the area. The first value is the cycle count, and the second value is the RBA.</td>
</tr>
</tbody>
</table>

Significant RBA Values

This part of the report shows data concerning significant relative byte address (RBA) values.

**Table 33  RBA values portion fields**

<table>
<thead>
<tr>
<th>Fields</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Root Anchor Point</td>
<td>The RBA of the first root anchor point in the area.</td>
</tr>
<tr>
<td>First IOVF CI</td>
<td>The RBA of the first IOVF CI in the area.</td>
</tr>
<tr>
<td>REORG UOW</td>
<td>The RBA of the REORG UOW of the area.</td>
</tr>
<tr>
<td>First SDEP CI</td>
<td>The RBA of the first SDEP CI in the area.</td>
</tr>
<tr>
<td>End of AREA</td>
<td>The RBA of the end of the area.</td>
</tr>
</tbody>
</table>

Performance Factors

This part of the report shows the level of fragmentation for the area.

Fragmentation—The fragmentation factor represents the level of fragmentation or disorganization in the area based on free space elements and out-of-block pointers.

To calculate this value, each block of data in the area is evaluated to determine conditions that differ from those found in a database with perfect organization. This value is based on the following assumptions:

- Only one free space element should be available in a block. Excessive numbers (more than one) of free space elements within a block indicate disorganization.
Segments within a block should be connected sequentially. The RBA pointer within these segments should chain between segments within the same block. Only one segment within a block should contain a pointer to another block. Excessive numbers (more than one) of pointers that point outside of the block indicate disorganization.

Free Space Analysis Report

The Free Space Analysis Report (Figure 15 on page 387) 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 additional 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 (the 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 that is defined in the area.
- Scraps (free space less than 4 bytes in length) are included in free space calculations, but they are not included in FSE calculations.
Table 34 lists the fields that are displayed in this report.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD Name</td>
<td>The DBD name of the database.</td>
</tr>
<tr>
<td>Area ddname</td>
<td>The ddname of the area.</td>
</tr>
</tbody>
</table>
Area Overview

This part of the report provides an overview of the free space available in various components of the area (RAP, DOVF, IOVF, and SDEP). The values are for the entire area, regardless of any UOW range selected.

Table 34  Free Space Analysis Report fields (part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area Number</td>
<td>The area number.</td>
</tr>
<tr>
<td>UOW Range</td>
<td>The range of UOWs that are included in the report. The value All indicates the entire area is being reported; a range value (start to stop) indicates the range of UOWs being reported.</td>
</tr>
</tbody>
</table>

UOW Range Overview

This part of the report is produced only when a range of UOWs has been selected for reporting. It provides an overview of the free space available in RAP, DOVF, and IOVF control intervals within the selected UOWs.
Analysis of RAP Blocks

This part of the report provides statistics about free space that is available with RAP blocks.

### Table 36  UOW range fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total % FS</td>
<td>The amount of unused space in each component (RAP, DOVF, and IOVF) of the selected UOW range, expressed as a percentage of total available space. This value includes all FSEs, scraps and unused blocks.</td>
</tr>
<tr>
<td>Usable % FS</td>
<td>The amount of unused space in each component (RAP, DOVF, and IOVF) of the selected UOW range, minus scraps and FSEs shorter than the length of the shortest segment (maximum defined length plus prefix length). This is the amount of space that IMS can use to insert new segments.</td>
</tr>
<tr>
<td>Unused</td>
<td>The free space that is attributable to unused blocks. Unused RAP blocks can be used by IMS to insert new database records only; they cannot be used for new segment occurrences of existing database records.</td>
</tr>
<tr>
<td>FS for Root</td>
<td>The percentage of RAP blocks with free space available for a new root segment.</td>
</tr>
</tbody>
</table>

### Analysis of RAP Blocks

This part of the report provides statistics about free space that is available with RAP blocks.

### Table 37  RAP blocks fields (part 1 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FS %</td>
<td>A free space percentage range that is used for classifying blocks for reporting purposes.</td>
</tr>
<tr>
<td>CIs</td>
<td>The number and percentage of RAP blocks that contain free space in the bounds of a reporting range:</td>
</tr>
<tr>
<td></td>
<td>■ No.—The total number of RAP blocks.</td>
</tr>
<tr>
<td></td>
<td>■ %—The percentage of RAP blocks.</td>
</tr>
<tr>
<td>No. of FSEs</td>
<td>Statistics about the number of FSEs in RAP blocks that contain free space in the bounds of a reporting range. Scraps are not included in the calculations.</td>
</tr>
<tr>
<td></td>
<td>■ Avg—The average number of FSEs in a block.</td>
</tr>
<tr>
<td></td>
<td>■ SDev—The standard deviation for the average.</td>
</tr>
<tr>
<td></td>
<td>■ Max—The maximum number of FSEs in a block.</td>
</tr>
<tr>
<td></td>
<td>■ Min—The minimum number of FSEs in a block.</td>
</tr>
</tbody>
</table>
Overflow Usage Summary

This part of the report summarizes information about overflow usage.

Cl's Using Overflow—The number and percentage of RAP blocks containing a pointer to a segment in a DOVF or IOVF block. Unused RAP blocks are excluded from the calculation.

- **No.**—The total number of used RAP blocks that contain a pointer to a segment in a DOVF or IOVF block.
- **%**—The percentage of used RAP blocks that contain a pointer to a segment in a DOVF or IOVF block.

Analysis of Overflow Blocks

This part of the report provides statistics about free space that is available in DOVF blocks.

Table 37 RAP blocks fields (part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of FSEs</td>
<td>Statistics about the size of FSEs in RAP blocks that contain free space in the bounds of a reporting range. Scraps are not included in the calculations.</td>
</tr>
<tr>
<td></td>
<td>- Avg—The average size of the FSEs.</td>
</tr>
<tr>
<td></td>
<td>- SDev—The standard deviation for the average.</td>
</tr>
<tr>
<td></td>
<td>- Max—The length of the longest FSE.</td>
</tr>
<tr>
<td></td>
<td>- Min—The length of the shortest FSE.</td>
</tr>
<tr>
<td>Using Ovflow</td>
<td>The number of RAP blocks in a free space reporting range that contain a pointer to a segment in a DOVF or IOVF block.</td>
</tr>
</tbody>
</table>

Table 38 DOVF blocks fields (part 1 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%DOVF Part FS</td>
<td>A free space percentage range that is used for classifying UOWs for reporting purposes. DOVF part free space for a UOW includes all free space in used and empty DOVF blocks in the UOW.</td>
</tr>
<tr>
<td>UOWs</td>
<td>The UOWs in the area where DOVF blocks contain free space in the bounds of the reporting range:</td>
</tr>
<tr>
<td></td>
<td>- No.—The total number of UOWs.</td>
</tr>
<tr>
<td></td>
<td>- %—The percentage of UOWs.</td>
</tr>
</tbody>
</table>
Table 38  DOVF blocks fields (part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOVF CIs Used</td>
<td>Statistics on the number of DOVF blocks containing at least one segment for each UOW in a free space reporting range:</td>
</tr>
<tr>
<td></td>
<td>▪ Avg—The average number of DOVF blocks that are used by a UOW.</td>
</tr>
<tr>
<td></td>
<td>▪ SDev—The standard deviation for the average.</td>
</tr>
<tr>
<td></td>
<td>▪ Max—The maximum number of DOVF blocks that are used by a UOW.</td>
</tr>
<tr>
<td></td>
<td>▪ Min—The minimum number of DOVF blocks that are used by a UOW.</td>
</tr>
<tr>
<td>No. of FSEs</td>
<td>Statistics on the number of FSEs in each DOVF block in a free space reporting range:</td>
</tr>
<tr>
<td></td>
<td>▪ Avg—The average number of FSEs in a DOVF block.</td>
</tr>
<tr>
<td></td>
<td>▪ SDev—The standard deviation for the average.</td>
</tr>
<tr>
<td></td>
<td>▪ Max—The maximum number of FSEs in a DOVF block.</td>
</tr>
<tr>
<td></td>
<td>▪ Min—The minimum number of FSEs in a DOVF block.</td>
</tr>
<tr>
<td>Size of FSEs</td>
<td>Statistics on the size of FSEs in each DOVF block in a free space reporting range:</td>
</tr>
<tr>
<td></td>
<td>▪ Avg—The average size of FSEs in a DOVF block.</td>
</tr>
<tr>
<td></td>
<td>▪ SDev—The standard deviation for the average.</td>
</tr>
<tr>
<td></td>
<td>▪ Max—The maximum size of FSEs in a DOVF block.</td>
</tr>
<tr>
<td></td>
<td>▪ Min—The minimum size of FSEs in a DOVF block.</td>
</tr>
<tr>
<td>In IOVF</td>
<td>The number of UOWs in a free space reporting range that also use one or more IOVF blocks.</td>
</tr>
<tr>
<td>IOVF CIs Used</td>
<td>Statistics on the number of IOVF blocks that are used by each UOW in a free space reporting range:</td>
</tr>
<tr>
<td></td>
<td>▪ Avg—The average number of IOVF blocks that are used by a UOW.</td>
</tr>
<tr>
<td></td>
<td>▪ SDev—The standard deviation for the average.</td>
</tr>
<tr>
<td></td>
<td>▪ Max—The maximum number of IOVF blocks that are used by a UOW.</td>
</tr>
<tr>
<td></td>
<td>▪ Min—The minimum number of IOVF blocks that are used by a UOW.</td>
</tr>
<tr>
<td>FSEs/IOVF CI</td>
<td>The average number of FSEs in each IOVF block that are used by each UOW in a free space reporting range.</td>
</tr>
</tbody>
</table>
DOVF Usage Summary

This part of the report provides the number and percentage of UOWs that use DOVF. It also provides statistics on the number of DOVF blocks that are used for UOWs that use DOVF.

Table 39  DOVF fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOWs Using DOVF</td>
<td>The number and percentage of UOWs that use DOVF:</td>
</tr>
<tr>
<td></td>
<td>■ No.—The number of UOWs that use DOVF.</td>
</tr>
<tr>
<td></td>
<td>■ % UOWs—The percentage of UOWs that use DOVF.</td>
</tr>
<tr>
<td>NO. DOVF CIs Used</td>
<td>Statistics on the number of DOVF blocks that are used for UOWs that use DOVF:</td>
</tr>
<tr>
<td></td>
<td>■ Avg—The average number of DOVF blocks that are used by a UOW.</td>
</tr>
<tr>
<td></td>
<td>■ SDev—The standard deviation for the average.</td>
</tr>
<tr>
<td></td>
<td>■ Max—The maximum number of DOVF blocks that are used by a UOW.</td>
</tr>
<tr>
<td></td>
<td>■ Min—The minimum number of DOVF blocks that are used by a UOW.</td>
</tr>
</tbody>
</table>

IOVF Usage Summary

This part of the report provides the number and percentage of UOWs that use IOVF. It also provides statistics on the number of IOVF blocks used for UOWs that use IOVF.
Table 40  IOVF fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOWs Using IOVF</td>
<td>The number and percentage of UOWs that use IOVF:</td>
</tr>
<tr>
<td></td>
<td>■ No.—The number of UOWs that use IOVF.</td>
</tr>
<tr>
<td></td>
<td>■ % UOWs—The percentage of UOWs that use IOVF.</td>
</tr>
<tr>
<td>No. IOVF CIs Used</td>
<td>Statistics on the number of IOVF blocks that are used for UOWs that use IOVF:</td>
</tr>
<tr>
<td></td>
<td>■ Avg—The average number of IOVF blocks that are used by a UOW.</td>
</tr>
<tr>
<td></td>
<td>■ SDev—The standard deviation for the average.</td>
</tr>
<tr>
<td></td>
<td>■ Max—The maximum number of IOVF blocks that are used by a UOW.</td>
</tr>
<tr>
<td></td>
<td>■ Min—The minimum number of IOVF blocks that are used by a UOW.</td>
</tr>
<tr>
<td>Available IOVF CIs</td>
<td>The number and percentage of unallocated (totally empty) IOVF blocks:</td>
</tr>
<tr>
<td></td>
<td>■ No.—The number of unallocated IOVF blocks.</td>
</tr>
<tr>
<td></td>
<td>■ %CIs—The percentage of unallocated IOVF blocks.</td>
</tr>
</tbody>
</table>

IOVF Space Analysis Report

The IOVF Space Analysis Report (Figure 16) provides IOVF block usage information obtained from each IOVF space map block to show how IOVF is being allocated. For an additional example of the report, see the PFPANLYZ member in the REPORTS data set.

The IOVF Space Analysis Report can be used to determine an IOVF usage pattern. The range scale indicates the amount of full CIs in the IOVF section.

Figure 16  IOVF Space Analysis Report
Table 41 lists the fields that are displayed in this report.

### Table 41  IOVF Space Analysis Report fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD Name</td>
<td>The DBD name of the database.</td>
</tr>
<tr>
<td>Area ddname</td>
<td>The ddname of the area.</td>
</tr>
<tr>
<td>Area Number</td>
<td>The area number.</td>
</tr>
<tr>
<td>Map RBA</td>
<td>The RBA of the IOVF space map block.</td>
</tr>
<tr>
<td>% IOVF CIs Used</td>
<td>A graph of the percentage of total entries that are controlled by the IOVF space map blocks that are allocated to a UOW.</td>
</tr>
<tr>
<td>No. Used</td>
<td>The number of entries in the IOVF space map block that are allocated to a UOW.</td>
</tr>
<tr>
<td>Note</td>
<td>A warning message indicating that a space map block is full (100% of the entries are allocated to a UOW).</td>
</tr>
</tbody>
</table>

### UOW Detailed Analysis Report

The UOW Detailed Analysis Report (Figure 17) provides detailed information about each UOW. You can use the report to better perform the following analysis tasks:

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

For an additional example of the report, see the PFPANLYZ member in the REPORTS data set.
Table 42 lists the fields that are displayed in this report.

**Table 42  UOW Detailed Analysis Report fields (part 1 of 2)**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD Name</td>
<td>The DBD name of the database.</td>
</tr>
<tr>
<td>Area ddname</td>
<td>The ddname of the area.</td>
</tr>
<tr>
<td>Area Number</td>
<td>The area number.</td>
</tr>
<tr>
<td>UOW Range</td>
<td>The range of UOWs that are included in the report. The value <em>All</em> indicates that the entire area is being reported; a range value (start to stop) indicates the range of UOWs being reported.</td>
</tr>
<tr>
<td>UOW No.</td>
<td>The UOW number.</td>
</tr>
<tr>
<td>Number CIs Used</td>
<td>Number of control intervals used per UOW</td>
</tr>
<tr>
<td></td>
<td>■ RAP—The number of RAP blocks in the UOW that contain data.</td>
</tr>
<tr>
<td></td>
<td>■ DOVF—The number of DOVF blocks in the UOW that contain data.</td>
</tr>
<tr>
<td></td>
<td>■ IOVF—The number of IOVF blocks that are allocated to the UOW.</td>
</tr>
<tr>
<td>No. RAPs Use Ovfl</td>
<td>The number of RAP blocks containing a pointer that references a DOVF block or an IOVF block.</td>
</tr>
<tr>
<td>% RAP FS</td>
<td>Statistics about the percentage of free space (FS) in used RAP blocks. Empty RAP blocks are excluded from the calculations.</td>
</tr>
<tr>
<td></td>
<td>■ Avg—The average percentage of free space in used RAP blocks.</td>
</tr>
<tr>
<td></td>
<td>■ Max—The maximum percentage of free space in used RAP blocks.</td>
</tr>
<tr>
<td></td>
<td>■ Min—The minimum percentage of free space in used RAP blocks.</td>
</tr>
<tr>
<td>% DOVF Part FS</td>
<td>The percentage of free space in the DOVF blocks in the UOW. DOVF free space for a UOW includes free space in used and unused DOVF blocks.</td>
</tr>
<tr>
<td>No. DB Records</td>
<td>The number of database records in the UOW.</td>
</tr>
</tbody>
</table>
For each segment type within a database, the Pointer Analysis Report (Figure 18) 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

The Pointer Analysis Report provides valuable information for performance tuning of the area. When pointers point into a different CI, additional I/O is required to follow the pointer.

**NOTE**

The Pointer Analysis Report can be generated when you specify either POINTER_VALIDATION=FULL or POINTER_VALIDATION=QUICK on the command set.

For an additional example of the report, see the PFPANLYZ member in the REPORTS data set.
Table 43 lists the fields that are displayed in this report.

Table 43  Pointer Analysis Report fields (part 1 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD Name</td>
<td>The DBD name of the database.</td>
</tr>
<tr>
<td>Area DDname</td>
<td>The ddname of the area.</td>
</tr>
<tr>
<td>Area DSname</td>
<td>The dsname of the area.</td>
</tr>
<tr>
<td>Area Number</td>
<td>The area number.</td>
</tr>
<tr>
<td>UOW Range</td>
<td>The range of UOWs included in the report. The value All indicates the entire area is being reported; a range value (start to stop) indicates the range of UOWs being reported.</td>
</tr>
<tr>
<td>Source Segment</td>
<td>The name of the segment containing the pointer.</td>
</tr>
<tr>
<td>Seg CD</td>
<td>The segment code for the segment as defined in the DBD.</td>
</tr>
<tr>
<td>Pointer Type</td>
<td>The type of pointer. For each segment type, this column will contain one of the following values:</td>
</tr>
<tr>
<td></td>
<td>- PTF (physical twin forward)</td>
</tr>
<tr>
<td></td>
<td>- PCF (physical child first)</td>
</tr>
<tr>
<td></td>
<td>- PCL (physical child last)</td>
</tr>
<tr>
<td></td>
<td>- SSPTR (subset pointer)</td>
</tr>
<tr>
<td></td>
<td>- SDEP-first (pointer from root to SDEP)</td>
</tr>
<tr>
<td></td>
<td>- SDEP-next (pointer from SDEP to next SDEP)</td>
</tr>
<tr>
<td>Target Segment</td>
<td>The name of the segment that is pointed to by the pointer.</td>
</tr>
<tr>
<td>Null Pointer Occurrences</td>
<td>The number of null pointers contained within the segment.</td>
</tr>
<tr>
<td>Null Pointer %</td>
<td>The percentage of total pointers within the segment that are null.</td>
</tr>
<tr>
<td>Point to Same CI Occurrences</td>
<td>The number of pointers contained within the segment that point to a location within the same control interval (CI).</td>
</tr>
<tr>
<td>Point to Same CI %</td>
<td>The percentage of total pointers within the segment that point to a location within the same control interval (CI).</td>
</tr>
</tbody>
</table>
The Segment I/O Analysis Report (Figure 19) provides the following statistics for a typical database record:

- direct dependent segments in each part of an area (SDEP segments are not included)
- physical I/O operations that are required to retrieve dependent segments
- physical I/O that is required to retrieve an average database record
- physical I/O that is required to retrieve an average root segment

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 can also be used to determine the optimal reorganization point and to validate the effects of UOW and ROOT value parameter changes. For an additional example of the report, see the PFPANLYZ member in the REPORTS data set.

**NOTE**
The Segment I/O Analysis Report can be generated only when POINTER_VALIDATION=FULL is selected on the primary command.

### Table 43 Pointer Analysis Report fields (part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point to Different CI</td>
<td>The number of pointers contained within the segment that point to a location within a different control interval (CI).</td>
</tr>
<tr>
<td>Occurrences</td>
<td></td>
</tr>
<tr>
<td>Point to Different CI %</td>
<td>The percentage of total pointers within the segment that point to a location within a different control interval (CI).</td>
</tr>
</tbody>
</table>

### Segment I/O Analysis Report

The Segment I/O Analysis Report (Figure 19) provides the following statistics for a typical database record:

- direct dependent segments in each part of an area (SDEP segments are not included)
- physical I/O operations that are required to retrieve dependent segments
- physical I/O that is required to retrieve an average database record
- physical I/O that is required to retrieve an average root segment

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 can also be used to determine the optimal reorganization point and to validate the effects of UOW and ROOT value parameter changes. For an additional example of the report, see the PFPANLYZ member in the REPORTS data set.

**NOTE**
The Segment I/O Analysis Report can be generated only when POINTER_VALIDATION=FULL is selected on the primary command.
** Record I/O **   Avg:   1.62      SDev:   0.87      Max:     6      Min:     1
*** Root I/O ***   Avg:   1.39      SDev:   0.70      Max:     5      Min:     1

Table 44 lists the fields that are displayed in this report.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD Name</td>
<td>The DBD name of the database.</td>
</tr>
<tr>
<td>Area ddname</td>
<td>The ddname of the area.</td>
</tr>
<tr>
<td>Area Number</td>
<td>The area number.</td>
</tr>
<tr>
<td>UOW Range</td>
<td>The range of UOWs included in the report. The value <em>All</em> indicates the</td>
</tr>
<tr>
<td></td>
<td>entire area is being reported; a range value <em>(start to stop)</em> indicates</td>
</tr>
<tr>
<td></td>
<td>the range of UOWs being reported.</td>
</tr>
<tr>
<td>SegName</td>
<td>The name of the segment as defined in the DBD.</td>
</tr>
<tr>
<td>Seg CD</td>
<td>The segment code of the segment.</td>
</tr>
<tr>
<td>Seg Lvl</td>
<td>The hierarchical level of the segment.</td>
</tr>
<tr>
<td>Avg Freq</td>
<td>The average occurrences of the segment in a database record.</td>
</tr>
<tr>
<td>Segment Placement</td>
<td>Statistics on the occurrences of a segment in a database record.</td>
</tr>
<tr>
<td>Statistics</td>
<td>In RAA Base</td>
</tr>
<tr>
<td></td>
<td>Information about segment occurrences in a database record that is stored</td>
</tr>
<tr>
<td></td>
<td>in RAP blocks:</td>
</tr>
<tr>
<td></td>
<td>▪ Avg—The average number of segments in RAP blocks.</td>
</tr>
<tr>
<td></td>
<td>▪ SDev—The standard deviation for the average.</td>
</tr>
<tr>
<td></td>
<td>▪ Max—The maximum number of segments in RAP blocks.</td>
</tr>
<tr>
<td></td>
<td>▪ Min—The minimum number of segments in RAP blocks.</td>
</tr>
<tr>
<td>In DOVF</td>
<td>Information about segment occurrences in a database record that is stored</td>
</tr>
<tr>
<td></td>
<td>in DOVF blocks:</td>
</tr>
<tr>
<td></td>
<td>▪ Avg—The average number of segments in DOVF blocks.</td>
</tr>
<tr>
<td></td>
<td>▪ SDev—The standard deviation for the average.</td>
</tr>
<tr>
<td></td>
<td>▪ Max—The maximum number of segments in DOVF blocks.</td>
</tr>
<tr>
<td></td>
<td>▪ Min—The minimum number of segments in DOVF blocks.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>In IOVF</strong></td>
<td>Information about segment occurrences in a database record that is stored in IOVF blocks:</td>
</tr>
<tr>
<td></td>
<td>▪ Avg—The average number of segments in IOVF blocks.</td>
</tr>
<tr>
<td></td>
<td>▪ SDev—The standard deviation for the average.</td>
</tr>
<tr>
<td></td>
<td>▪ Max—The maximum number of segments in IOVF blocks.</td>
</tr>
<tr>
<td></td>
<td>▪ Min—The minimum number of segments in IOVF blocks.</td>
</tr>
<tr>
<td><strong>Physical I/O Statistics</strong></td>
<td>Statistics about the number of physical I/O operations that are required to retrieve the dependent segments in a hierarchical path:</td>
</tr>
<tr>
<td></td>
<td>▪ Avg—The average number of physical I/O operations that are required to retrieve the dependent segments in a hierarchical path.</td>
</tr>
<tr>
<td></td>
<td>▪ SDev—The standard deviation for the average.</td>
</tr>
<tr>
<td></td>
<td>▪ Max—The maximum number of physical I/O operations that are required to retrieve the dependent segments in a hierarchical path.</td>
</tr>
<tr>
<td></td>
<td>▪ Min—The minimum number of physical I/O operations that are required to retrieve the dependent segments in a hierarchical path.</td>
</tr>
<tr>
<td></td>
<td>The values reported for a segment indicate the I/O required to retrieve all occurrences of that segment type in a database record. The values show the physical I/O operations that are required for that segment type only, assuming that all parent segments in the hierarchical path had been retrieved.</td>
</tr>
</tbody>
</table>
The Segment Length Analysis Report (Figure 20) 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 allows you to see the level of compression for compressed segments.

For an additional example of the report, see the PFPANLYZ member in the REPORTS data set.

Table 44  Segment I/O Analysis Report fields (part 3 of 3)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record I/O</td>
<td>Statistics about the number of physical I/O operations that are required to retrieve an entire database record:</td>
</tr>
<tr>
<td></td>
<td>■ Avg—The average number of physical I/O operations that are required to retrieve an entire database record.</td>
</tr>
<tr>
<td></td>
<td>■ SDev—The standard deviation for the average.</td>
</tr>
<tr>
<td></td>
<td>■ Max—The maximum number of physical I/O operations that are required to retrieve an entire database record.</td>
</tr>
<tr>
<td></td>
<td>■ Min—The minimum number of physical I/O operations that are required to retrieve an entire database record.</td>
</tr>
<tr>
<td>Root I/O</td>
<td>Statistics about the number of physical I/O operations that are required to retrieve a root segment:</td>
</tr>
<tr>
<td></td>
<td>■ Avg—The average number of physical I/O operations that are required to retrieve a root segment.</td>
</tr>
<tr>
<td></td>
<td>■ SDev—The standard deviation for the average.</td>
</tr>
<tr>
<td></td>
<td>■ Max—The maximum number of physical I/O operations that are required to retrieve a root segment.</td>
</tr>
<tr>
<td></td>
<td>■ Min—The minimum number of physical I/O operations that are required to retrieve a root segment.</td>
</tr>
</tbody>
</table>

Segment Length Analysis Report

The Segment Length Analysis Report (Figure 20) 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 allows you to see the level of compression for compressed segments.

For an additional example of the report, see the PFPANLYZ member in the REPORTS data set.

Figure 20  Segment Length Analysis Report (part 1 of 2)
Figure 20  Segment Length Analysis Report (part 2 of 2)

<table>
<thead>
<tr>
<th>Segment Length</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100-140</td>
<td></td>
<td></td>
</tr>
<tr>
<td>141-181</td>
<td></td>
<td></td>
</tr>
<tr>
<td>182-222</td>
<td></td>
<td></td>
</tr>
<tr>
<td>223-263</td>
<td></td>
<td></td>
</tr>
<tr>
<td>264-304</td>
<td></td>
<td></td>
</tr>
<tr>
<td>305-345</td>
<td></td>
<td></td>
</tr>
<tr>
<td>346-386</td>
<td></td>
<td></td>
</tr>
<tr>
<td>387-427</td>
<td></td>
<td></td>
</tr>
<tr>
<td>428-468</td>
<td></td>
<td></td>
</tr>
<tr>
<td>469-509</td>
<td></td>
<td></td>
</tr>
<tr>
<td>510-550</td>
<td></td>
<td></td>
</tr>
<tr>
<td>551-591</td>
<td></td>
<td></td>
</tr>
<tr>
<td>592-632</td>
<td></td>
<td></td>
</tr>
<tr>
<td>633-673</td>
<td></td>
<td></td>
</tr>
<tr>
<td>674-714</td>
<td></td>
<td></td>
</tr>
<tr>
<td>715-755</td>
<td></td>
<td></td>
</tr>
<tr>
<td>756-796</td>
<td></td>
<td></td>
</tr>
<tr>
<td>797-837</td>
<td></td>
<td></td>
</tr>
<tr>
<td>838-878</td>
<td></td>
<td></td>
</tr>
<tr>
<td>879-900</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 45 lists the fields that are displayed in this report.

Table 45  Segment Length Analysis Report fields (part 1 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD Name</td>
<td>The DBD name of the database.</td>
</tr>
<tr>
<td>Area ddname</td>
<td>The ddname of the area.</td>
</tr>
<tr>
<td>Area Number</td>
<td>The area number.</td>
</tr>
<tr>
<td>Segment Name</td>
<td>The name of the segment that is being reported.</td>
</tr>
<tr>
<td>Bytes</td>
<td>The maximum and minimum segments lengths that are defined in the DMB for this segment type. For fixed length segments, the bytes value contains a single value, which reflects the defined length of the maximum segment length from the Record Profile Analysis Report.</td>
</tr>
</tbody>
</table>
| Segment Length| The range of segment lengths that are included on this report line. The reporting intervals for non-compressed segments use the following calculation: $(\text{max} - \text{min}) / 20$
where $\text{max}$ is the maximum defined length for the segment type and $\text{min}$ is the minimum defined length for the segment type.

The reporting intervals for compressed segments use the following calculation:
$max / 20$

where $\text{max}$ is the maximum defined length for the segment type.

| % Segments | A graph of the percentage of all segment occurrences with a length in the indicated range. |
The Segment Placement Analysis Report (Figure 21) provides an overview of where segments reside in the database. For an additional example of the report, see the PFPANLYZ member in the REPORTS data set.

**NOTE**

SDEP segments are not included in this report.

### Table 45  Segment Length Analysis Report fields (part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segments</td>
<td>The number and percentage of segment occurrences with a length in the indicated range:</td>
</tr>
<tr>
<td></td>
<td>■ No.—The number of segment occurrences with a length in the indicated range.</td>
</tr>
<tr>
<td></td>
<td>■ %—The percentage of all segment occurrences with a length in the indicated range.</td>
</tr>
<tr>
<td>Length Statistics</td>
<td>Statistics on the length of the segment occurrences with a length in the indicated range:</td>
</tr>
<tr>
<td></td>
<td>■ Avg—The average length of the segment occurrences.</td>
</tr>
<tr>
<td></td>
<td>■ SDev—The standard deviation of the average.</td>
</tr>
</tbody>
</table>

### Figure 21  Segment Placement Analysis Report

The Segment Placement Analysis Report (Figure 21) provides an overview of where segments reside in the database. For an additional example of the report, see the PFPANLYZ member in the REPORTS data set.

**NOTE**

SDEP segments are not included in this report.
Table 46 lists the fields that are displayed in this report.

Table 46  Segment Placement Analysis Report fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD Name</td>
<td>The DBD name of the database.</td>
</tr>
<tr>
<td>Area ddname</td>
<td>The ddname of the area.</td>
</tr>
<tr>
<td>Area Number</td>
<td>The area number.</td>
</tr>
<tr>
<td>UOW Range</td>
<td>The range of UOWs included in the report. The value <em>All</em> indicates the entire area is being reported; a range value (<em>start to stop</em>) indicates the range of UOWs being reported.</td>
</tr>
<tr>
<td>SegName</td>
<td>The name of the segment as defined in the DBD.</td>
</tr>
<tr>
<td>Seg CD</td>
<td>The segment code of the segment.</td>
</tr>
<tr>
<td>Seg Lvl</td>
<td>The hierarchical level of the segment.</td>
</tr>
<tr>
<td>Total Segs</td>
<td>The total number of occurrences of the segment in the area.</td>
</tr>
<tr>
<td>In RAA Base</td>
<td>The number and percentage of segment occurrences that are stored in RAP blocks:</td>
</tr>
<tr>
<td></td>
<td>▪ No.—The number of occurrences of the segment that are stored in RAP blocks.</td>
</tr>
<tr>
<td></td>
<td>▪ %—The percentage of the total segment occurrences that are stored in RAP blocks.</td>
</tr>
<tr>
<td>In DOVF</td>
<td>The number and percentage of segment occurrences that are stored in DOVF blocks:</td>
</tr>
<tr>
<td></td>
<td>▪ No.—The number of occurrences of the segment that are stored in DOVF blocks.</td>
</tr>
<tr>
<td></td>
<td>▪ %—The percentage of the total segment occurrences that are stored in DOVF blocks.</td>
</tr>
<tr>
<td>In IOVF</td>
<td>The number and percentage of segment occurrences that are stored in IOVF blocks:</td>
</tr>
<tr>
<td></td>
<td>▪ No.—The number of occurrences of the segment that are stored in IOVF blocks.</td>
</tr>
<tr>
<td></td>
<td>▪ %—The percentage of the total segment occurrences that are stored in IOVF blocks.</td>
</tr>
</tbody>
</table>

Record Length Analysis Report

The Record Length Analysis Report (Figure 22) 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.
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. For an additional example of the report, see the PFPANLYZ member in the REPORTS data set.

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.

**NOTE**

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

---

**Figure 22  Record Length Analysis Report (part 1 of 2)**

<table>
<thead>
<tr>
<th>Record Length</th>
<th>% Records</th>
<th>--Records--</th>
<th>% of Total</th>
<th>--Record Length Statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 2 3 4 5 6 7 8 9 0</td>
<td>No.</td>
<td>%</td>
<td>Avg</td>
</tr>
<tr>
<td>1-250</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>251-500</td>
<td>5 1.4 1.4</td>
<td>409</td>
<td>448</td>
<td>336</td>
</tr>
<tr>
<td>501-750</td>
<td>1 0.3 1.7</td>
<td>734</td>
<td>734</td>
<td>734</td>
</tr>
<tr>
<td>751-1,000</td>
<td>13 3.7 5.4</td>
<td>869</td>
<td>993</td>
<td>784</td>
</tr>
<tr>
<td>1,001-1,250</td>
<td>20 5.7 11.2</td>
<td>1,125</td>
<td>1,238</td>
<td>1,010</td>
</tr>
<tr>
<td>1,251-1,500</td>
<td>32 9.2 20.3</td>
<td>1,392</td>
<td>1,497</td>
<td>1,263</td>
</tr>
<tr>
<td>1,501-1,750</td>
<td>27 7.7 28.1</td>
<td>1,601</td>
<td>1,728</td>
<td>1,508</td>
</tr>
<tr>
<td>1,751-2,000</td>
<td>42 12.0 40.1</td>
<td>1,895</td>
<td>1,997</td>
<td>1,772</td>
</tr>
<tr>
<td>2,001-2,250</td>
<td>9 2.6 42.7</td>
<td>2,176</td>
<td>2,247</td>
<td>2,063</td>
</tr>
<tr>
<td>2,251-2,500</td>
<td>31 8.9 51.6</td>
<td>2,389</td>
<td>2,500</td>
<td>2,269</td>
</tr>
<tr>
<td>2,501-2,750</td>
<td>20 5.7 57.3</td>
<td>2,662</td>
<td>2,739</td>
<td>2,554</td>
</tr>
<tr>
<td>2,751-3,000</td>
<td>19 5.4 62.8</td>
<td>2,854</td>
<td>2,985</td>
<td>2,757</td>
</tr>
<tr>
<td>3,001-3,250</td>
<td>17 4.9 67.6</td>
<td>3,137</td>
<td>3,242</td>
<td>3,021</td>
</tr>
<tr>
<td>3,251-3,500</td>
<td>13 3.7 71.3</td>
<td>3,388</td>
<td>3,464</td>
<td>3,306</td>
</tr>
<tr>
<td>3,501-3,750</td>
<td>10 2.9 74.2</td>
<td>3,667</td>
<td>3,741</td>
<td>3,538</td>
</tr>
<tr>
<td>3,751-4,000</td>
<td>14 4.0 78.2</td>
<td>3,875</td>
<td>3,979</td>
<td>3,764</td>
</tr>
</tbody>
</table>
Table 47 lists the fields that are displayed in this report.

**Table 47  Record Length Analysis Report fields**

<table>
<thead>
<tr>
<th>Fields</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD Name</td>
<td>The DBD name of the database.</td>
</tr>
<tr>
<td>Area ddname</td>
<td>The ddname of the area.</td>
</tr>
<tr>
<td>Area Number</td>
<td>The area number.</td>
</tr>
<tr>
<td>UOW Range</td>
<td>The range of UOWs included in the report. The value All indicates the entire area is being reported; a range value (start to stop) indicates the range of UOWs being reported.</td>
</tr>
</tbody>
</table>

**Record Length Analysis**

This part of the report shows the distribution of database records by length.

**Table 48  Record length analysis fields (part 1 of 2)**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record Length</td>
<td>The range of record lengths that are included on the report line. The increment for successive lines is specified by the RECORD_LENGTH_INCREMENT keyword.</td>
</tr>
<tr>
<td>% Records</td>
<td>A graph of the percentage of all database records with a length in the indicated range:</td>
</tr>
</tbody>
</table>
Table 48  Record length analysis fields (part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Records</td>
<td>The number and percentage of database records in the indicated range.</td>
</tr>
<tr>
<td>No.</td>
<td>The number of database records with a length in the indicated range.</td>
</tr>
<tr>
<td>%</td>
<td>The percentage of all database records with a length in the indicated range.</td>
</tr>
<tr>
<td>Record Length Statistics</td>
<td>Statistics about the length of the records in the indicated range:</td>
</tr>
<tr>
<td>Avg</td>
<td>The average length of the database records with a length in the indicated range.</td>
</tr>
<tr>
<td>Max</td>
<td>The maximum length of the database records with a length in the indicated range.</td>
</tr>
<tr>
<td>Min</td>
<td>The minimum length of the database records with a length in the indicated range.</td>
</tr>
</tbody>
</table>

Largest Database Records

This part of the report contains information about the 10 largest database records in the area, in descending order. You can change this number by specifying the LARGEST_DATABASE_RECORDS keyword under the primary command that creates the information.

Table 49  Largest database records fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record Size</td>
<td>The size of the database record in bytes.</td>
</tr>
<tr>
<td>Segments In Record</td>
<td>The number of segment occurrences in the database record.</td>
</tr>
<tr>
<td>Root RBA</td>
<td>The RBA of the root segment.</td>
</tr>
<tr>
<td>Root Segment Sequence Field</td>
<td>The value of the root segment sequence field (key) is displayed in hexadecimal and character formats. The first 16 bytes of the key are displayed.</td>
</tr>
</tbody>
</table>
The Record Placement Analysis Report (Figure 23) shows the number of database records in each of seven placement categories and the statistics about the number of DOVF and IOVF blocks that are 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 provides information to help you determine:

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

For an additional example of the report, see the PFPANLYZ member in the REPORTS data set.

**NOTE**

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD Name</td>
<td>The DBD name of the database.</td>
</tr>
<tr>
<td>Area ddname</td>
<td>The ddname of the area.</td>
</tr>
<tr>
<td>Area Number</td>
<td>The area number.</td>
</tr>
</tbody>
</table>
Table 50  Record Placement Analysis Report fields (part 2 of 3)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOW Range</td>
<td>The range of UOWs that are included in the report. The value <em>All</em> indicates the entire area is being reported; a range value <em>(start to stop)</em> indicates the range of UOWs that are being reported.</td>
</tr>
<tr>
<td>Records In</td>
<td>The number of database records is provided for the following categories:</td>
</tr>
<tr>
<td></td>
<td>- Base Only—The number of database records in which all segments are contained in the base RAP block.</td>
</tr>
<tr>
<td></td>
<td>- Base + DOVF—The number of database records in which at least one segment is contained in the base RAP block, at least one segment is contained in a DOVF block, and no segments are stored in an IOVF block.</td>
</tr>
<tr>
<td></td>
<td>- Base + DOVF + IOVF—The number of database records in which at least one segment is contained in the base RAP block, at least one segment is contained in a DOVF block, and at least one segment is stored in an IOVF block.</td>
</tr>
<tr>
<td></td>
<td>- Base + IOVF—The number of database records in which at least one segment is contained in the base RAP block, at least one segment is contained in a IOVF block, and no segments are stored in a DOVF block.</td>
</tr>
<tr>
<td></td>
<td>- DOVF Only—The number of database records in which all segments are stored in DOVF blocks.</td>
</tr>
<tr>
<td></td>
<td>- DOVF + IOVF—The number of database records in which at least one segment is contained in a DOVF block, at least one segment is contained in an IOVF block, and no segments are stored in a RAP block.</td>
</tr>
<tr>
<td></td>
<td>- IOVF Only—The number of database records in which all segments are contained in IOVF blocks.</td>
</tr>
<tr>
<td>Records</td>
<td>The number and percentage of database records in the indicated categories:</td>
</tr>
<tr>
<td></td>
<td>- No.—The number of database records in which segments are contained in the indicated category.</td>
</tr>
<tr>
<td></td>
<td>- %—The percentage of all database records in which segments are contained in the indicated category.</td>
</tr>
</tbody>
</table>
The Record Profile Analysis Report (Figure 24) shows statistics about the number and length of the database records and 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 can be used to determine the following:

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

For an additional example of the report, see the PFPANLYZ member in the REPORTS data set.
Figure 24  Record Profile Analysis Report

Table 51 lists the fields that are displayed in this report.

Table 51  Record Profile Analysis Report fields (part 1 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD Name</td>
<td>The DBD name of the database.</td>
</tr>
<tr>
<td>Area ddname</td>
<td>The ddname of the area.</td>
</tr>
<tr>
<td>Area Number</td>
<td>The area number.</td>
</tr>
<tr>
<td>UOW Range</td>
<td>The range of UOWs included in the report. The value All indicates the entire area is being reported; a range value (start to stop) indicates the range of UOWs being reported.</td>
</tr>
<tr>
<td>No. DB Records</td>
<td>The number of database records in an area.</td>
</tr>
<tr>
<td>Rec Length Avg</td>
<td>The average length of a database record, including the lengths of the data portions and the segment prefixes.</td>
</tr>
<tr>
<td>Rec Length SDev</td>
<td>The standard deviation for the average length. This value is reported only when POINTER_VALIDATION=FULL has been specified.</td>
</tr>
<tr>
<td>Rec Length Max</td>
<td>The length of the longest database record in an area. This value is reported only when POINTER_VALIDATION=FULL has been specified.</td>
</tr>
<tr>
<td>Rec Length Min</td>
<td>The length of the shortest database record in an area. This value is reported only when POINTER_VALIDATION=FULL has been specified.</td>
</tr>
<tr>
<td>SegName</td>
<td>The name of the segment as defined in the DBD.</td>
</tr>
<tr>
<td>Seg CD</td>
<td>The segment code of the segment.</td>
</tr>
<tr>
<td>Seg Lvl</td>
<td>The hierarchical level of the segment.</td>
</tr>
<tr>
<td>Total Segs</td>
<td>The total number of occurrences of the segment in an area.</td>
</tr>
</tbody>
</table>
### Table 51  Record Profile Analysis Report fields (part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment Frequency</td>
<td>Statistics on the segment frequency in a database record:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Avg</strong>—The average number of occurrences of a segment in a database record.</td>
</tr>
<tr>
<td></td>
<td>- <strong>SDev</strong>—The standard deviation for the average. This value is reported only when <code>POINTER_VALIDATION=FULL</code> has been specified.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Max</strong>—The maximum number of occurrences of a segment for a database record. This value is reported only when <code>POINTER_VALIDATION=FULL</code> has been specified.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Min</strong>—The minimum number of occurrences of a segment for a database record. This value is reported only when <code>POINTER_VALIDATION=FULL</code> has been specified.</td>
</tr>
<tr>
<td>Defined Length</td>
<td>The segment lengths as defined in the DBD (excluding the segment prefix):</td>
</tr>
<tr>
<td></td>
<td>- <strong>Max</strong>—The maximum segment length as defined in the DBD.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Min</strong>—The minimum segment length as defined in the DBD. For fixed length segments, this value is blank.</td>
</tr>
<tr>
<td>Actual Length</td>
<td>Statistics about the actual segment lengths in the database area (excluding the segment prefix). For compressed segments, these lengths represent the number of bytes that are used after compression.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Avg</strong>—The average segment length in the database area.</td>
</tr>
<tr>
<td></td>
<td>- <strong>SDev</strong>—The standard deviation for the average.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Max</strong>—The maximum segment length in the database area.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Min</strong>—The minimum segment length in the database area.</td>
</tr>
</tbody>
</table>
Synonym Chain Analysis Report

The Synonym Chain Analysis Report (Figure 25) gives detailed information about the frequency and length of synonym chains. The report provides statistics about root segment placement and physical I/O requirements. It provides valuable information to determine the following:

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

**NOTE**

The Synonym Chain Analysis Report can be generated only when POINTER_VALIDATION=FULL or RAP_VALIDATION=XREF is specified.

For an additional example of the report, see the PFPANLYZ member in the REPORTS data set.

**Figure 25  Synonym Chain Analysis Report (part 1 of 2)**

<table>
<thead>
<tr>
<th>No. Roots Per RAP</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>% RAPs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAPs</td>
<td>1</td>
<td>17</td>
<td>22</td>
<td>17</td>
<td>21</td>
<td>13</td>
<td>7</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Synonym Chain Statistics:</td>
<td>Average</td>
<td>St.Dev.</td>
<td>Maximum</td>
<td>RBA (Max)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Length of Synonym Chain</td>
<td>3.06</td>
<td>2.02</td>
<td>10</td>
<td>0007E000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I/O to read Synonym Chain</td>
<td>1.44</td>
<td>0.93</td>
<td>5</td>
<td>0007E000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Synonym Chain Analysis**

<table>
<thead>
<tr>
<th>No. Roots/ RAP</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>---RAPS---</td>
<td>11</td>
<td>17</td>
</tr>
<tr>
<td>No. %</td>
<td>9.6</td>
<td>14.9</td>
</tr>
<tr>
<td>Avg/ SDev/ Min</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Max/ Avg/ SDev</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Min/ Max/ Max</td>
<td>0.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

[Table continues...]

**NOTE**

The Synonym Chain Analysis Report can be generated only when POINTER_VALIDATION=FULL or RAP_VALIDATION=XREF is specified.
Table 52 lists the fields that are displayed in this report.

**Table 52** Synonym Chain Analysis Report fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD Name</td>
<td>The DBD name of the database.</td>
</tr>
<tr>
<td>Area ddname</td>
<td>The ddname of the area.</td>
</tr>
<tr>
<td>Area Number</td>
<td>The area number.</td>
</tr>
<tr>
<td>UOW Range</td>
<td>The range of UOWs included in the report. The value <em>All</em> indicates the entire area is being reported; a range value (start to stop) indicates the range of UOWs being reported.</td>
</tr>
</tbody>
</table>

**Roots per RAP Distribution**

This part of the report shows the distribution of the synonym chain length (roots per RAP).

**Table 53** Roots per RAP Distribution fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. Roots Per RAP</td>
<td>The number of root segments chained from a RAP (the length of the synonym chain). The <em>y</em> axis increases in length (up to 40 lines) as required to include all RAPs. The increment for successive lines is specified by the SYNONYM_CHAIN_INCREMENT keyword.</td>
</tr>
<tr>
<td>% RAPs</td>
<td>A graph of the percentage of RAP blocks that have the indicated number of root segments in the synonym chain.</td>
</tr>
<tr>
<td>No. RAPs</td>
<td>The number (count) of RAP blocks that have the indicated number of root segments in the synonym chain.</td>
</tr>
</tbody>
</table>
**Synonym Chain Statistics**

This part of the report displays summary information about the synonym chains and the physical I/O required to retrieve them.

**Table 54  Synonym Chain Statistics fields**

<table>
<thead>
<tr>
<th>Fields</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length of Synonym Chain</td>
<td>This report line displays the average number of root segments chained from the RAPs and the standard deviation. It also displays the maximum number of root segments that are chained from any RAP, and the RBA of that RAP. If multiple RAPs have the same chain length, only the first RAP (lowest RBA) is reported.</td>
</tr>
<tr>
<td>I/O to Read Synonym Chain</td>
<td>This report line displays the average number of physical I/O operations that are required to retrieve all of the root segments chained from any RAP, and the standard deviation. If also displays the maximum number of physical I/O operations required to retrieve all of the root segments chained to any RAP, and the RBA of that RAP. If multiple RAP chains require the same number of I/O operations, only the first RAP (lowest RBA) is reported.</td>
</tr>
</tbody>
</table>

**Synonym Chain Analysis**

This part of the report displays information about the root segments randomized to a RAP and the physical I/O required to retrieve them.

**Table 55  Synonym Chain Analysis fields (part 1 of 3)**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. Roots/RAP</td>
<td>The number of root segments that are chained from a RAP (the length of the synonym chain). This table increases in length (up to 40 lines) to include all RAPs. The increment for successive lines is specified by the SYNONYM_CHAIN_INCREMENT keyword.</td>
</tr>
<tr>
<td>RAPs</td>
<td>The number and percentage of RAP blocks that have the indicated number of root segments in the synonym chain: No.—The number of RAP blocks. %—The percentage of RAP blocks.</td>
</tr>
<tr>
<td>Root Placement Statistics</td>
<td>Statistics about the root segments for RAP blocks containing the indicated synonym chain length.</td>
</tr>
</tbody>
</table>
### Synonym Chain Analysis fields (part 2 of 3)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In RAA Base</strong></td>
<td>Root segments in the synonym chain that are placed in a RAP block:</td>
</tr>
<tr>
<td>Avg</td>
<td>The average number of root segments in a RAP block.</td>
</tr>
<tr>
<td>SDev</td>
<td>The standard deviation for the average.</td>
</tr>
<tr>
<td>Max</td>
<td>The maximum number of root segments in a RAP block.</td>
</tr>
<tr>
<td>Min</td>
<td>The minimum number of root segments in a RAP block.</td>
</tr>
<tr>
<td><strong>In DOVF</strong></td>
<td>Root segments in the synonym chain that are placed in a DOVF block:</td>
</tr>
<tr>
<td>Avg</td>
<td>The average number of root segments in a DOVF block.</td>
</tr>
<tr>
<td>SDev</td>
<td>The standard deviation for the average.</td>
</tr>
<tr>
<td>Max</td>
<td>The maximum number of root segments in a DOVF block.</td>
</tr>
<tr>
<td>Min</td>
<td>The minimum number of root segments in a DOVF block.</td>
</tr>
<tr>
<td><strong>In IOVF</strong></td>
<td>Root segments in the synonym chain that are placed in an IOVF block:</td>
</tr>
<tr>
<td>Avg</td>
<td>The average number of root segments in an IOVF block.</td>
</tr>
<tr>
<td>SDev</td>
<td>The standard deviation for the average.</td>
</tr>
<tr>
<td>Max</td>
<td>The maximum number of root segments in an IOVF block.</td>
</tr>
<tr>
<td>Min</td>
<td>The minimum number of root segments in an IOVF block.</td>
</tr>
</tbody>
</table>
DEDB Unload Reports

Fast Path Reorg/EP produces the following unload reports:

- Unload Input Area Summary
- Unload Output Area Summary
- Unload Database Summary

These reports are generated automatically any time the Fast Path Reorg/EP UNLOAD command is executed. The reports contain detailed information about database performance, space usage, segment characteristics, and record characteristics related to the unload process.

For additional examples of the reports, refer to PFPUNLD in the REPORTS data set.

For information about reports that are generated during the reload process, see “DEDB Reload Reports” on page 426.

Unload Input Area Summary Report

The Unload Input Area Summary Report (Figure 26) provides basic information about each input area that has been unloaded. For an additional example of the report, see the PFPUNLD member in the REPORTS data set.
Figure 26  Unload Input Area Summary Report

Table 56 lists the fields that are displayed in this report.

Table 56  Unload Input Area Summary Report fields (part 1 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>The database name.</td>
</tr>
<tr>
<td>Area</td>
<td>The area name.</td>
</tr>
<tr>
<td>ddname</td>
<td>The ddname for the area.</td>
</tr>
<tr>
<td>dsname</td>
<td>The dsname of the area data set.</td>
</tr>
<tr>
<td>Area Number</td>
<td>The area number.</td>
</tr>
<tr>
<td>Randomizing Module Name</td>
<td>The value that is defined in the</td>
</tr>
<tr>
<td></td>
<td>DBD.</td>
</tr>
</tbody>
</table>
Table 56  Unload Input Area Summary Report fields (part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOW</td>
<td>The first value is the number of blocks in each unit of work (UOW); the second value is the number of dependent overflow (DOVF) blocks in each UOW, as defined in the DBD.</td>
</tr>
<tr>
<td>ROOT</td>
<td>The first value is the number of UOWs in the area; the second value is the number of overflow UOWs in the area, as defined in the DBD.</td>
</tr>
<tr>
<td>CISIZE</td>
<td>The Control Interval Size (CISIZE) value that is defined in the DBD.</td>
</tr>
</tbody>
</table>

**Root Addressable Area Portion**

This part of the report shows data concerning the root addressable (RAA) portion of the area.

Table 57  Root Addressable Area Portion fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOWs in RAA</td>
<td>The number of UOWs in the root addressable part of the area.</td>
</tr>
<tr>
<td>Total CIs per UOW</td>
<td>The number of CIs for each UOW.</td>
</tr>
<tr>
<td>RAP CIs per UOW</td>
<td>The number of root anchor point (RAP) CIs for each UOW.</td>
</tr>
<tr>
<td>DOVF CIs per UOW</td>
<td>The number of DOVF CIs in each UOW.</td>
</tr>
<tr>
<td>Total Root Anchor Points</td>
<td>The number of root addressable blocks in the area.</td>
</tr>
<tr>
<td>Total Dependent Overflow CIs</td>
<td>The number of DOVF CIs in the area.</td>
</tr>
</tbody>
</table>

**Independent Overflow Portion**

This part of the report shows data concerning the independent overflow (IOVF) portion of the area.

Table 58  Independent Overflow Portion fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Independent Overflow CIs</td>
<td>The number of independent overflow (IOVF) CIs in the area.</td>
</tr>
<tr>
<td>IOVF Space Map CIs</td>
<td>The number of IOVF CIs that are used to map free space.</td>
</tr>
<tr>
<td>IOVF Data CIs</td>
<td>The number of IOVF CIs that are usable for data storage.</td>
</tr>
</tbody>
</table>
Sequential Dependent Portion

This part of the report shows data concerning the sequential dependent (SDEP) portion of the area.

Table 59  Sequential Dependent Portion fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Sequential Dependent CIs</td>
<td>The number of CIs that are available for sequential dependents.</td>
</tr>
<tr>
<td>SDEP Logical Begin</td>
<td>The location of the first sequential dependent (SDEP) segment occurrence in the area. The first value is the cycle count, and the second value is the relative byte address (RBA).</td>
</tr>
<tr>
<td>SDEP Logical End</td>
<td>The location of the last SDEP segment occurrence in the area. The first value is the cycle count, and the second value is the RBA.</td>
</tr>
<tr>
<td>SDEP Logical Begin Time Stamp</td>
<td>The time of the first valid sequential dependent (SDEP) segment occurrence in the area.</td>
</tr>
</tbody>
</table>

Significant RBA Values

This part of the report shows data concerning significant relative byte address (RBA) values.

Table 60  Significant RBA Values fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Root Anchor Point</td>
<td>The RBA of the first root anchor point in the area.</td>
</tr>
<tr>
<td>First IOVF CI</td>
<td>The RBA of the first IOVF CI in the area.</td>
</tr>
<tr>
<td>REORG UOW</td>
<td>The RBA of the REORG UOW of the area.</td>
</tr>
<tr>
<td>First SDEP CI</td>
<td>The RBA of the first SDEP CI in the area.</td>
</tr>
<tr>
<td>End of AREA</td>
<td>The RBA of the end of the area.</td>
</tr>
</tbody>
</table>

Segment Input Summary

This part of the report provides a summary of the total number of segments read and selected from the area data set.

Table 61  Segment Input Summary fields (part 1 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment Name</td>
<td>The name of the segment as defined in the DBD.</td>
</tr>
<tr>
<td>Segment Code</td>
<td>The segment code of the segment.</td>
</tr>
</tbody>
</table>
Table 61 Segment Input Summary fields (part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment Type</td>
<td>The type of segment.</td>
</tr>
<tr>
<td>Compress Routine</td>
<td>The name of the compression routine defined for the segment in the DBD.</td>
</tr>
<tr>
<td>Bytes</td>
<td>The maximum and minimum segment length (in bytes) that are defined in the DMB for this segment type.</td>
</tr>
<tr>
<td></td>
<td>- Max—The maximum segment length that is defined in the DMB for this segment type.</td>
</tr>
<tr>
<td></td>
<td>- Min—The minimum segment length that is defined in the DMB for this segment type.</td>
</tr>
<tr>
<td></td>
<td>- Len—The length format defined for this segment type. The segment type can be fixed length (FL) or variable length (VL).</td>
</tr>
<tr>
<td>Segment Counts</td>
<td>The total number of segments read and selected for the input area.</td>
</tr>
<tr>
<td></td>
<td>- Read—The total number of segments read for the input area.</td>
</tr>
<tr>
<td></td>
<td>- Selected—The total number of segments that met the specified selection criteria for the input area.</td>
</tr>
</tbody>
</table>

Unload Output Area Summary Report

The Unload Output Area Summary Report (Figure 27) provides basic information about each output file that has been written. For an additional example of the report, see the PFPUNLD member in the REPORTS data set.

Figure 27 Unload Output Area Summary Report (part 1 of 2)
Table 62 lists the fields that are displayed in this report.

### Table 62  Unload Output Area Summary Report fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>The database name.</td>
</tr>
<tr>
<td>Area</td>
<td>The name of the area.</td>
</tr>
<tr>
<td>ddname</td>
<td>The ddname for the area.</td>
</tr>
<tr>
<td>dsname</td>
<td>The dsname of the area data set.</td>
</tr>
<tr>
<td>Area Number</td>
<td>The area number.</td>
</tr>
<tr>
<td>Randomizing Module Name</td>
<td>The value that is defined in the DBD.</td>
</tr>
<tr>
<td>UOW</td>
<td>The first value is the number of blocks in each unit of work (UOW); the second value is the number of dependent overflow (DOVF) blocks in each UOW, as defined in the DBD.</td>
</tr>
<tr>
<td>ROOT</td>
<td>The first value is the number of UOWs in the area; the second value is the number of overflow UOWs in the area, as defined in the DBD.</td>
</tr>
<tr>
<td>CISIZE</td>
<td>The Control Interval Size (CISIZE) value that is defined in the DBD</td>
</tr>
</tbody>
</table>
Root Addressable Area Portion

This part of the report shows data concerning the root addressable (RAA) portion of the area.

Table 63  Root Addressable Area Portion fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOWs in RAA</td>
<td>The number of UOWs in the root addressable part of the area.</td>
</tr>
<tr>
<td>Total CIs per UOW</td>
<td>The number of CIs for each UOW.</td>
</tr>
<tr>
<td>RAP CIs per UOW</td>
<td>The number of root anchor point (RAP) CIs for each UOW.</td>
</tr>
<tr>
<td>DOVF CIs per UOW</td>
<td>The number of DOVF CIs in each UOW.</td>
</tr>
<tr>
<td>Total Root Anchor Points</td>
<td>The number of root addressable blocks in the area.</td>
</tr>
<tr>
<td>Total Dependent Overflow CIs</td>
<td>The number of DOVF CIs in the area.</td>
</tr>
</tbody>
</table>

Independent Overflow Portion

This part of the report shows data concerning the independent overflow (IOVF) portion of the area.

Table 64  Independent Overflow Portion fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Independent Overflow CIs</td>
<td>The number of independent overflow (IOVF) CIs in the area.</td>
</tr>
<tr>
<td>IOVF Space Map CIs</td>
<td>The number of IOVF CIs that are used to map free space.</td>
</tr>
<tr>
<td>IOVF Data CIs</td>
<td>The number of IOVF CIs that are usable for data storage.</td>
</tr>
</tbody>
</table>

Sequential Dependent Portion

This part of the report shows data concerning the sequential dependent (SDEP) portion of the area.

Table 65  Sequential Dependent Portion fields (part 1 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Sequential Dependent CIs</td>
<td>The number of CIs that are available for sequential dependents.</td>
</tr>
<tr>
<td>SDEP Logical Begin</td>
<td>The location of the first sequential dependent (SDEP) segment occurrence in the area. The first value is the cycle count, and the second value is the relative byte address (RBA).</td>
</tr>
</tbody>
</table>
### Significant RBA Values

This part of the report shows data concerning significant relative byte address (RBA) values.

**Table 65  Sequential Dependent Portion fields (part 2 of 2)**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDEP Logical End</td>
<td>The location of the last SDEP segment occurrence in the area. The first value is the cycle count, and the second value is the RBA.</td>
</tr>
</tbody>
</table>

### Segment Output Summary

This part of the report provides a summary of the total number of segments selected and written to the output file.

**Table 66  Significant RBA Values fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Root Anchor Point</td>
<td>The RBA of the first root anchor point in the area.</td>
</tr>
<tr>
<td>First IOVF CI</td>
<td>The RBA of the first IOVF CI in the area.</td>
</tr>
<tr>
<td>REORG UOW</td>
<td>The RBA of the REORG UOW of the area.</td>
</tr>
<tr>
<td>First SDEP CI</td>
<td>The RBA of the first SDEP CI in the area.</td>
</tr>
<tr>
<td>End of AREA</td>
<td>The RBA of the end of the area.</td>
</tr>
</tbody>
</table>

**Table 67  Segment Output Summary fields (part 1 of 2)**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment Name</td>
<td>The name of the segment as defined in the DBD.</td>
</tr>
<tr>
<td>Segment Code</td>
<td>The segment code of the segment.</td>
</tr>
<tr>
<td>Segment Type</td>
<td>The type of segment.</td>
</tr>
<tr>
<td>Compress Routine</td>
<td>The name of the compression routine defined for the segment in the DBD.</td>
</tr>
</tbody>
</table>
The Unload Database Summary Report (Figure 28) provides basic information about the totals for each segment that has been unloaded for all areas processed in the job. For an additional example of the report, see the PFPUNLD member in the REPORTS data set.

### Unload Database Summary Report

The Unload Database Summary Report provides basic information about the totals for each segment that has been unloaded for all areas processed in the job. For an additional example of the report, see the PFPUNLD member in the REPORTS data set.

### Table 67 Segment Output Summary fields (part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bytes</strong></td>
<td>The maximum and minimum segment length (in bytes) that are defined in the DMB for this segment type:</td>
</tr>
<tr>
<td></td>
<td>- Max—The maximum segment length that is defined in the DMB for this segment type.</td>
</tr>
<tr>
<td></td>
<td>- Min—The minimum segment length that is defined in the DMB for this segment type.</td>
</tr>
<tr>
<td></td>
<td>- Len—The length format defined for this segment type. The segment type can be fixed length (FL) or variable length (VL).</td>
</tr>
<tr>
<td><strong>Segment Counts</strong></td>
<td>The total number of segments selected and written for the output file:</td>
</tr>
<tr>
<td></td>
<td>- Selected—The total number of segments selected for the output file.</td>
</tr>
<tr>
<td></td>
<td>- Written—The total number of segments written for the output file.</td>
</tr>
</tbody>
</table>

### Figure 28 Unload Database Summary Report

**Table:** Unload Database Output Summary fields (part 2 of 2)

<table>
<thead>
<tr>
<th>Segment</th>
<th>Segment</th>
<th>Type</th>
<th>Compress</th>
<th>Bytes</th>
<th>Read</th>
<th>Selected</th>
<th>Written</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEGA</td>
<td>1</td>
<td>ROOT</td>
<td></td>
<td>500</td>
<td>126</td>
<td>126</td>
<td>126</td>
</tr>
<tr>
<td>SDEP</td>
<td>2</td>
<td>SDEP</td>
<td></td>
<td>130</td>
<td>239</td>
<td>239</td>
<td>239</td>
</tr>
<tr>
<td>SEGB</td>
<td>3</td>
<td>DDEP</td>
<td></td>
<td>200</td>
<td>389</td>
<td>389</td>
<td>389</td>
</tr>
<tr>
<td>SEGC</td>
<td>4</td>
<td>DDEP</td>
<td></td>
<td>150</td>
<td>569</td>
<td>569</td>
<td>569</td>
</tr>
<tr>
<td>SEGD</td>
<td>5</td>
<td>DDEP</td>
<td></td>
<td>130</td>
<td>541</td>
<td>541</td>
<td>541</td>
</tr>
<tr>
<td>SEGE</td>
<td>6</td>
<td>DDEP</td>
<td></td>
<td>145</td>
<td>325</td>
<td>325</td>
<td>325</td>
</tr>
<tr>
<td>Totals</td>
<td></td>
<td></td>
<td></td>
<td>2,189</td>
<td>2,189</td>
<td>2,189</td>
<td>2,189</td>
</tr>
</tbody>
</table>
Table 68 lists the fields that are displayed in this report.

### Table 68  Unload Database Summary Report fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>The database name.</td>
</tr>
<tr>
<td>Area</td>
<td>This field denotes all areas processed.</td>
</tr>
<tr>
<td>ddname</td>
<td>This field denotes all ddnames processed for the area.</td>
</tr>
<tr>
<td>dsname</td>
<td>The dsname is not applicable for this report.</td>
</tr>
<tr>
<td>Segment Name</td>
<td>The name of the segment as defined in the DBD.</td>
</tr>
<tr>
<td>Segment Code</td>
<td>The segment code of the segment.</td>
</tr>
<tr>
<td>Segment Type</td>
<td>The type of segment.</td>
</tr>
<tr>
<td>Compress Routine</td>
<td>The name of the compression routine defined for the segment in the DBD.</td>
</tr>
<tr>
<td>Bytes</td>
<td>The maximum and minimum segment length (in bytes) that are defined in the DMB for this segment type:</td>
</tr>
<tr>
<td></td>
<td>- Max—The maximum segment length that is defined in the DMB for this segment type.</td>
</tr>
<tr>
<td></td>
<td>- Min—The minimum segment length that is defined in the DMB for this segment type.</td>
</tr>
<tr>
<td></td>
<td>- Len—The length format defined for this segment type. The segment types can be fixed length (FL) or variable length (VL).</td>
</tr>
<tr>
<td>Segment Counts</td>
<td>The total number of segments read, selected, and written for all areas processed during the unload process:</td>
</tr>
<tr>
<td></td>
<td>- Read—The total number of segments read for all areas processed during the unload process.</td>
</tr>
<tr>
<td></td>
<td>- Selected—The total number of segments selected for all areas processed during the unload process.</td>
</tr>
<tr>
<td></td>
<td>- Written—The total number of segments written for all areas processed during the unload process.</td>
</tr>
</tbody>
</table>

**DEDB Reload Reports**

Fast Path Reorg/EP produces the following reload reports:

- Reload Input Area Summary
- Reload Output Area Summary
- Reload Database Summary
These reports are generated automatically any time the Fast Path Reorg/EP RELOAD command is executed. The reports contain detailed information about database performance, space usage, segment characteristics, and record characteristics related to the reload process.

For additional examples of the reports, refer to PFPRELD in the REPORTS data set.

For information about reports that are generated during the unload process, see “DEDB Unload Reports” on page 417.

Reload Input Area Summary Report

The Reload Input Area Summary Report (Figure 29) provides basic information about each input file that has been reloaded. For an additional example of the report, see the PFPRELD member in the REPORTS data set.

Figure 29  Reload Input Area Summary Report (part 1 of 2)
Table 69 lists the fields that are displayed in this report.

### Table 69  Reload Input Area Summary Report fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>The database name.</td>
</tr>
<tr>
<td>Area</td>
<td>The area name.</td>
</tr>
<tr>
<td>ddname</td>
<td>The ddname for the input file.</td>
</tr>
<tr>
<td>dsname</td>
<td>The dsname of the input file.</td>
</tr>
<tr>
<td>Area Number</td>
<td>The area number.</td>
</tr>
<tr>
<td>Randomizing Module Name</td>
<td>The value that is defined in the DBD.</td>
</tr>
<tr>
<td>UOW</td>
<td>The first value is the number of blocks in each unit of work (UOW); the second value is the number of dependent overflow (DOVF) blocks in each UOW, as defined in the DBD.</td>
</tr>
<tr>
<td>ROOT</td>
<td>The first value is the number of UOWs in the area; the second value is the number of overflow UOWs in the area, as defined in the DBD.</td>
</tr>
<tr>
<td>CISIZE</td>
<td>The Control Interval Size (CISIZE) value that is defined in the DBD.</td>
</tr>
</tbody>
</table>

### Root Addressable Area Portion

This part of the report shows data concerning the root addressable (RAA) portion of the area.

### Table 70  Root Addressable Area Portion fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOWs in RAA</td>
<td>The number of UOWs in the root addressable part of the area.</td>
</tr>
<tr>
<td>Total CIs per UOW</td>
<td>The number of CIs for each UOW.</td>
</tr>
<tr>
<td>RAP CIs per UOW</td>
<td>The number of root anchor point (RAP) CIs for each UOW.</td>
</tr>
<tr>
<td>DOVF CIs per UOW</td>
<td>The number of DOVF CIs in each UOW.</td>
</tr>
<tr>
<td>Total Root Anchor Points</td>
<td>The number of root addressable blocks in the area.</td>
</tr>
<tr>
<td>Total Dependent Overflow CIs</td>
<td>The number of DOVF CIs in the area.</td>
</tr>
</tbody>
</table>
Independent Overflow Portion

This part of the report shows data concerning the independent overflow (IOVF) portion of the area.

Table 71  Independent Overflow Portion fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Independent Overflow CIs</td>
<td>The number of independent overflow (IOVF) CIs in the area.</td>
</tr>
<tr>
<td>IOVF Space Map CIs</td>
<td>The number of IOVF CIs that are used to map free space.</td>
</tr>
<tr>
<td>IOVF Data CIs</td>
<td>The number of IOVF CIs that are used usable for data storage.</td>
</tr>
</tbody>
</table>

Sequential Dependent Portion

This part of the report shows data concerning the sequential dependent (SDEP) portion of the area.

Table 72  Sequential Dependent Portion fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Sequential Dependent CIs</td>
<td>The number of CIs that are available for sequential dependents.</td>
</tr>
<tr>
<td>SDEP Logical Begin</td>
<td>The location of the first sequential dependent (SDEP) segment occurrence in the area. The first value is the cycle count, and the second value is the relative byte address (RBA).</td>
</tr>
<tr>
<td>SDEP Logical End</td>
<td>The location of the last SDEP segment occurrence in the area. The first value is the cycle count, and the second value is the RBA.</td>
</tr>
<tr>
<td>SDEP Logical Begin Time Stamp</td>
<td>The time of the first valid sequential dependent (SDEP) segment occurrence in the area.</td>
</tr>
</tbody>
</table>

Significant RBA Values

This part of the report shows data concerning significant relative byte address (RBA) values.

Table 73  Significant RBA Values fields (part 1 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Root Anchor Point</td>
<td>The RBA of the first root anchor point in the area.</td>
</tr>
<tr>
<td>First IOVF CI</td>
<td>The RBA of the first IOVF CI in the area.</td>
</tr>
<tr>
<td>REORG UOW</td>
<td>The RBA of the REORG UOW of the area.</td>
</tr>
</tbody>
</table>
Segment Input Summary

This part of the report provides a summary of the total number of segments read and selected from the area data set.

Table 74  Segment Input Summary fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment Name</td>
<td>The name of the segment as defined in the DBD.</td>
</tr>
<tr>
<td>Segment Code</td>
<td>The segment code of the segment.</td>
</tr>
<tr>
<td>Segment Type</td>
<td>The type of segment.</td>
</tr>
<tr>
<td>Compress Routine</td>
<td>The name of the compression routine defined for the segment in the DBD.</td>
</tr>
<tr>
<td>Bytes</td>
<td>The maximum and minimum segment length (in bytes) that are defined in the DMB for this segment type:</td>
</tr>
<tr>
<td></td>
<td>- Max—The maximum segment length that is defined in the DMB for this segment type.</td>
</tr>
<tr>
<td></td>
<td>- Min—The minimum segment length that is defined in the DMB for this segment type.</td>
</tr>
<tr>
<td></td>
<td>- Len—The length format defined for this segment type. The segment type can be fixed length (FL) or variable length (VL).</td>
</tr>
<tr>
<td>Segment Counts</td>
<td>The total number of segments read and selected from the input file:</td>
</tr>
<tr>
<td></td>
<td>- Read—The total number of segments read from the input file.</td>
</tr>
<tr>
<td></td>
<td>- Selected—The total number of segments that met any specified selection criteria from the input file.</td>
</tr>
</tbody>
</table>

Reload Output Area Summary Report

The Reload Output Area Summary Report (Figure 30) provides basic information about each output area that has been reloaded. For an additional example of the report, see the PFPRELD member in the REPORTS data set.
Figure 30  Reload Output Area Summary Report

Table 75 Reload Output Area Summary Report (part 1 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>The database name.</td>
</tr>
<tr>
<td>Area</td>
<td>The area name.</td>
</tr>
<tr>
<td>ddname</td>
<td>The ddname for the area.</td>
</tr>
<tr>
<td>dsname</td>
<td>The dsname of the area data set.</td>
</tr>
<tr>
<td>Area Number</td>
<td>The area number.</td>
</tr>
<tr>
<td>Randomizing Module Name</td>
<td>The value that is defined in the DBD.</td>
</tr>
</tbody>
</table>
### Root Addressable Area Portion

This part of the report shows data concerning the root addressable (RAA) portion of the area.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOW</td>
<td>The first value is the number of blocks in each unit of work (UOW); the second value is the number of dependent overflow (DOVF) blocks in each UOW, as defined in the DBD.</td>
</tr>
<tr>
<td>ROOT</td>
<td>The first value is the number of UOWs in the area; the second value is the number of overflow UOWs in the area, as defined in the DBD.</td>
</tr>
<tr>
<td>CISIZE</td>
<td>The Control Interval Size (CISIZE) value that is defined in the DBD.</td>
</tr>
</tbody>
</table>

### Independent Overflow Portion

This part of the report shows data concerning the independent overflow (IOVF) portion of the area.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Independent Overflow CIs</td>
<td>The number of independent overflow (IOVF) CIs in the area.</td>
</tr>
<tr>
<td>IOVF Space Map CIs</td>
<td>The number of IOVF CIs that are used to map free space.</td>
</tr>
<tr>
<td>IOVF Data CIs</td>
<td>The number of IOVF CIs that are used usable for data storage.</td>
</tr>
</tbody>
</table>
**Sequential Dependent Portion**

This part of the report shows data concerning the sequential dependent (SDEP) portion of the area.

**Table 78  Sequential Dependent Portion fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Sequential Dependent CIs</td>
<td>The number of CIs that are available for sequential dependents.</td>
</tr>
<tr>
<td>SDEP Logical Begin</td>
<td>The location of the first sequential dependent (SDEP) segment occurrence in the area. The first value is the cycle count, and the second value is the relative byte address (RBA).</td>
</tr>
<tr>
<td>SDEP Logical End</td>
<td>The location of the last SDEP segment occurrence in the area. The first value is the cycle count, and the second value is the RBA.</td>
</tr>
<tr>
<td>SDEP Logical Begin Time Stamp</td>
<td>The time of the first valid sequential dependent (SDEP) segment occurrence in the area.</td>
</tr>
</tbody>
</table>

**Significant RBA Values**

This part of the report shows data concerning significant relative byte address (RBA) values.

**Table 79  Significant RBA Values fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Root Anchor Point</td>
<td>The RBA of the first root anchor point in the area.</td>
</tr>
<tr>
<td>First IOVF CI</td>
<td>The RBA of the first IOVF CI in the area.</td>
</tr>
<tr>
<td>REORG UOW</td>
<td>The RBA of the REORG UOW of the area.</td>
</tr>
<tr>
<td>First SDEP CI</td>
<td>The RBA of the first SDEP CI in the area.</td>
</tr>
<tr>
<td>End of AREA</td>
<td>The RBA of the end of the area.</td>
</tr>
</tbody>
</table>

**Segment Output Summary**

This part of the report provides a summary of the total number of segments selected and written to the area data set.

**Table 80  Segment Output Summary fields (part 1 of 2)**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment Name</td>
<td>The name of the segment as defined in the DBD.</td>
</tr>
<tr>
<td>Segment Code</td>
<td>The segment code of the segment.</td>
</tr>
</tbody>
</table>
The Reload Database Summary Report (Figure 31) provides basic information about the totals for each segment that has been reloaded for all areas processed in the job. For an additional example of the report, see the PFPRELD member in the REPORTS data set.

### Table 80  Segment Output Summary fields (part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment Type</td>
<td>The type of segment.</td>
</tr>
<tr>
<td>Compress Routine</td>
<td>The name of the compression routine defined for the segment in the DBD.</td>
</tr>
<tr>
<td>Bytes</td>
<td>The maximum and minimum segment length (in bytes) that are defined in the DMB for this segment type:</td>
</tr>
<tr>
<td></td>
<td>▪ Max—The maximum segment length that is defined in the DMB for this segment type.</td>
</tr>
<tr>
<td></td>
<td>▪ Min—The minimum segment length that is defined in the DMB for this segment type.</td>
</tr>
<tr>
<td></td>
<td>▪ Len—The length format defined for this segment type. The segment type can be fixed length (FL) or variable length (VL).</td>
</tr>
<tr>
<td>Segment Counts</td>
<td>The total number of segments selected and written for the output area:</td>
</tr>
<tr>
<td></td>
<td>▪ Selected—The total number of segments selected for the output area.</td>
</tr>
<tr>
<td></td>
<td>▪ Written—The total number of segments written for the output area.</td>
</tr>
</tbody>
</table>

### Reload Database Summary Report

The Reload Database Summary Report (Figure 31) provides basic information about the totals for each segment that has been reloaded for all areas processed in the job. For an additional example of the report, see the PFPRELD member in the REPORTS data set.

**Figure 31  Reload Database Summary Report**

<table>
<thead>
<tr>
<th>Date: 03/09/2007</th>
<th>FAST PATH REORG/EP</th>
<th>(3.8.00)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time: 07:10AM</td>
<td>RELOAD Database Summary Report</td>
<td>Page 1</td>
</tr>
<tr>
<td>Database: DP$10</td>
<td>Area: <strong>ALL</strong></td>
<td>DDname: <strong>ALL</strong></td>
</tr>
<tr>
<td></td>
<td>DSname: N/A</td>
<td></td>
</tr>
<tr>
<td>Segment Name</td>
<td>Segment Code</td>
<td>Type</td>
</tr>
<tr>
<td>SEGA</td>
<td>1</td>
<td>ROOT</td>
</tr>
<tr>
<td>SDEP</td>
<td>2</td>
<td>SDEP</td>
</tr>
<tr>
<td>SEGB</td>
<td>3</td>
<td>DDEP</td>
</tr>
<tr>
<td>SEGC</td>
<td>4</td>
<td>DDEP</td>
</tr>
<tr>
<td>SEGQ</td>
<td>5</td>
<td>DDEP</td>
</tr>
<tr>
<td>SEGE</td>
<td>6</td>
<td>DDEP</td>
</tr>
<tr>
<td>Totals</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 81 lists the fields that are displayed in this report.

**Table 81  Reload Database Summary Report**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>The database name.</td>
</tr>
<tr>
<td>Area</td>
<td>This field denotes all areas processed.</td>
</tr>
<tr>
<td>ddname</td>
<td>This field denotes all ddnames processed for the area.</td>
</tr>
<tr>
<td>dsname</td>
<td>The dsname is not applicable for this report.</td>
</tr>
<tr>
<td>Segment Name</td>
<td>The name of the segment as defined in the DBD.</td>
</tr>
<tr>
<td>Segment Code</td>
<td>The segment code of the segment.</td>
</tr>
<tr>
<td>Segment Type</td>
<td>The type of segment.</td>
</tr>
<tr>
<td>Compress Routine</td>
<td>The name of the compression routine defined for the segment in the DBD.</td>
</tr>
<tr>
<td>Bytes</td>
<td>The maximum and minimum segment length (in bytes) that are defined in the DMB for this segment type:</td>
</tr>
<tr>
<td></td>
<td>■ Max—The maximum segment length that is defined in the DMB for this segment type.</td>
</tr>
<tr>
<td></td>
<td>■ Min—The minimum segment length that is defined in the DMB for this segment type.</td>
</tr>
<tr>
<td></td>
<td>■ Len—The length format defined for this segment type. The segment types can be fixed length (FL) or variable length (VL).</td>
</tr>
<tr>
<td>Segment Counts</td>
<td>The total number of segments read, selected, and written for all areas processed during the reload process:</td>
</tr>
<tr>
<td></td>
<td>■ Read—The total number of segments read for all areas processed during the reload process.</td>
</tr>
<tr>
<td></td>
<td>■ Selected—The total number of segments selected for all areas processed during the reload process.</td>
</tr>
<tr>
<td></td>
<td>■ Written—The total number of segments written for all areas processed during the reload process.</td>
</tr>
</tbody>
</table>

**DEDB Extend Report**

Fast Path Reorg/EP produces only one extend report, the Extend Area Summary Report. This report is generated automatically any time the Fast Path Online Reorg/EP or Fast Path Reorg/EP EXTEND command is executed.
The Extend Area Summary Report (Figure 32) provides basic information about each area that has been extended (increased). It provides information about the size of IOVF and SDEP portions of a DEDB area before an extend has been executed and after an extend has been executed. For an additional example of the report, see the PFPXTND member in the REPORTS data set.

Figure 32  Extend Area Summary Report

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD Name</td>
<td>The database description (DBD) name of the database.</td>
</tr>
<tr>
<td>Area ddname</td>
<td>The ddname of the area.</td>
</tr>
<tr>
<td>Area dsname</td>
<td>The dsname of the area.</td>
</tr>
<tr>
<td>Randomizing Module Name</td>
<td>The value that is defined in the DBD.</td>
</tr>
</tbody>
</table>
Table 82  Extend Area Summary Report fields (part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOW</td>
<td>The first value is the number of blocks in each unit of work (UOW); the second value is the number of dependent overflow (DOVF) blocks in each UOW, as defined in the DBD.</td>
</tr>
<tr>
<td>CISIZE</td>
<td>The Control Interval Size (CISIZE) value that is defined in the DBD.</td>
</tr>
<tr>
<td>ROOT</td>
<td>The first value is the number of UOWs in the area; the second value is the number of overflow UOWs in the area, as defined in the DBD.</td>
</tr>
</tbody>
</table>

Root Addressable Area Portion

This part of the report shows data concerning the root addressable (RAA) portion of the area.

Table 83  Root Addressable Area Portion fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOWs in RAA</td>
<td>The number of UOWs in the root addressable part of the area.</td>
</tr>
<tr>
<td>Total CIs per UOW</td>
<td>The number of CIs for each UOW.</td>
</tr>
<tr>
<td>RAP CIs per UOW</td>
<td>The number of root anchor point (RAP) CIs for each UOW.</td>
</tr>
<tr>
<td>DOVF CIs per UOW</td>
<td>The number of DOVF CIs in each UOW.</td>
</tr>
<tr>
<td>Total Root Anchor Points</td>
<td>The number of root addressable blocks in the area.</td>
</tr>
<tr>
<td>Total Dependent Overflow CIs</td>
<td>The number of DOVF CIs in the area.</td>
</tr>
</tbody>
</table>

Independent Overflow Portion

This part of the report shows data concerning the independent overflow (IOVF) portion of the area before (Original column) and after (Extended column) an area extension.

Table 84  Independent Overflow Portion fields (part 1 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Independent Overflow CIs</td>
<td>The number of independent overflow (IOVF) CIs in the area.</td>
</tr>
<tr>
<td>IOVF Space Map CIs</td>
<td>The number of IOVF CIs that are used to map free space.</td>
</tr>
<tr>
<td>IOVF Data CIs</td>
<td>The number of IOVF CIs that are usable for data storage.</td>
</tr>
</tbody>
</table>
Sequential Dependent Portion

This part of the report shows data concerning the sequential dependent (SDEP) portion of the area before (Original column) and after (Extended column) an area extension.

Table 85  Sequential Dependent Portion fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Sequential</td>
<td>The number of CIs that are available for sequential dependents.</td>
</tr>
<tr>
<td>Dependent CIs</td>
<td></td>
</tr>
<tr>
<td>SDEP Logical Begin</td>
<td>The location of the first sequential dependent (SDEP) segment occurrence in</td>
</tr>
<tr>
<td></td>
<td>the area. The first value is the cycle count, and the second value is the</td>
</tr>
<tr>
<td></td>
<td>relative byte address (RBA).</td>
</tr>
<tr>
<td>SDEP Logical End</td>
<td>The location of the last SDEP segment occurrence in the area. The first</td>
</tr>
<tr>
<td></td>
<td>value is the cycle count, and the second value is the RBA.</td>
</tr>
<tr>
<td>SDEP Logical Begin</td>
<td>The time of the first valid sequential dependent (SDEP) segment occurrence</td>
</tr>
<tr>
<td>Time Stamp</td>
<td>in the area.</td>
</tr>
<tr>
<td>DMACLBTS</td>
<td>This field is the data management area control block logical begin time</td>
</tr>
<tr>
<td></td>
<td>stamp (DMACLBTS). It contains the hexadecimal date and time that is found</td>
</tr>
<tr>
<td></td>
<td>in the DMAC and represents the time of the first valid sequential dependent</td>
</tr>
<tr>
<td></td>
<td>segment occurrence in the area expressed in hexadecimal format.</td>
</tr>
<tr>
<td>Used SDEP CI’s</td>
<td>The number of SDEP CIs that are used in the area. This number is also</td>
</tr>
<tr>
<td></td>
<td>expressed as a percentage.</td>
</tr>
<tr>
<td>Free SDEP CI’s</td>
<td>The number of SDEP CIs that are available (not used) in the area. This</td>
</tr>
<tr>
<td></td>
<td>number is also expressed as a percentage.</td>
</tr>
</tbody>
</table>
**Significant RBA Values**

This part of the report shows data concerning significant relative byte address (RBA) values.

**Table 86  Significant RBA Values fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Root Anchor Point</td>
<td>The RBA of the first root anchor point in the area.</td>
</tr>
<tr>
<td>First IOVF CI</td>
<td>The RBA of the first IOVF CI in the area.</td>
</tr>
<tr>
<td>REORG UOW</td>
<td>The RBA of the REORG UOW of the area.</td>
</tr>
<tr>
<td>First SDEP CI</td>
<td>The RBA of the first SDEP CI in the area.</td>
</tr>
<tr>
<td>End of AREA</td>
<td>The RBA of the end of the area.</td>
</tr>
</tbody>
</table>
Supporting utilities

The Fast Path/EP Series products offer several supporting utilities. This chapter describes the benefits they provide as well as how to set up and execute them. This chapter discusses the following topics:

Available utilities ............................................. 442
Area Change Modeling Utility ................................ 443
   Utility program modules .................................. 444
   Input requirements ........................................ 445
   Randomizer considerations ............................... 446
   Process flow .............................................. 447
   PFMD0100 JCL requirements ............................. 449
   PFMD0100 control statements ........................... 451
   PFMD0100 return codes .................................. 455
   SORT13 .................................................... 455
   PFMD0300 JCL requirements ............................. 455
   PFMD0300 return codes .................................. 457
   SORT12 .................................................... 457
   SORT3 ..................................................... 458
   PFMD0500 JCL requirements ............................. 459
   PFMD0500 return codes .................................. 460
   Area Change Modeling Utility sample JCL ......... 460
   User procedures .......................................... 462
   User procedures for full-function databases .......... 463
   Comparing randomizing routines ....................... 464
   Area Change Modeling Utility sample scenarios ... 465
DMAC Print utility (DMAC_PRINT command) ............ 465
   Specifying the database and areas ..................... 466
   Offline and online examples ............................ 467
Control Interval Dump and Modification Utility
   (PROCESS_AREA command) .............................. 468
   PROCESS_AREA keywords and subcommands ....... 468
   Selecting the database and areas ...................... 469
   Offline considerations .................................. 470
   Offline control statements ............................. 472
   Online control statement ................................ 472
   Available functions ...................................... 473
   Syntax requirements ..................................... 473
Available utilities

Fast Path/EP Series products provides several supporting utilities for DEDB maintenance:

- The Area Change Modeling Utility models changes to an area to evaluate the performance and space utilization consequences of a proposed change to an existing DEDB, or of converting an IMS full-function database to the DEDB format.

- The DMAC Print Utility (DMAC_PRINT command) prints contents of the DMAC block for an area.
The Control Interval Dump and Modification Utility (PROCESS_AREA command) provides access to the contents of a database control interval for verification, inspection, repair, and modification.

The File Sort Utility (PFPSORT command) provides a method for invoking your installation’s sort utility to perform a sort of an input file in either root key or randomized sequence.

The Randomizer Interface Routine (PFUT0B50) provides an easy-to-use facility for invoking a DEDB randomizing routine from a high-level programming language.

The SDEP Space Utilization Utility reports on SDEP space utilization, extracts and updates SDEP history files, reformats SDEP space utilization records, and generates SDEP space utilization reports.

Area Change Modeling Utility

The Area Change Modeling Utility can be used to evaluate potential benefits or effects of changes to an existing DEDB before implementation. This utility provides a comprehensive DEDB modeling and prototyping facility. Information derived from using this utility can be used to help you better meet your application performance or space utilization requirements.

The Area Change Modeling Utility helps you to:

- determine database attribute and parameter value selection
- identify and select the physical database attributes that meet the desired performance and space utilization requirements (optimal values for the ROOT and UOW parameters)
- eliminate database maintenance required when a database does not meet performance and/or space utilization expectations after reorganization
- simplify the evaluation and selection of suitable randomizing routines
- evaluate the potential benefits or effect of database reorganization
- easily perform multiple iterations of the database modeling process, letting you select physical database attributes that meet the desired performance and space utilization requirements
- model the potential space utilization and performance effects of converting a full-function IMS database to the DEDB format
With the Area Change Modeling Utility, you can change and model any or all of the following database specifications:

- randomizing module (change in type of randomizer used)
- UOW parameter values
- ROOT parameter values
- number of database areas
- CI sizes
- changes to hierarchical database structure

The Area Change Modeling Utility produces a series of reports that provide detailed information about the performance space utilization effects of the modeled database changes.

### Utility program modules

Table 87 shows the program modules and intermediate sort steps that comprise the Area Change Modeling Utility:

<table>
<thead>
<tr>
<th>Stepname</th>
<th>Program</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFMD0100</td>
<td>PFMD0100</td>
<td>Extract model data</td>
</tr>
<tr>
<td>SORT13</td>
<td>SORT</td>
<td>Sort extracted records</td>
</tr>
<tr>
<td>PFMD0300</td>
<td>PFMD0300</td>
<td>Build model data</td>
</tr>
<tr>
<td>SORT12</td>
<td>SORT</td>
<td>Sort area and freespace information records</td>
</tr>
<tr>
<td>SORT3</td>
<td>SORT</td>
<td>Sort the segment information records</td>
</tr>
<tr>
<td>PFMD0500</td>
<td>PFMD0500</td>
<td>Generate Fast Path Analyzer/EP reports</td>
</tr>
</tbody>
</table>

**Extract data for model (PFMD0100)**

Module PFMD0100 extracts the data to be modeled from the database and areas based on the control statement input.

**Model database changes (PFMD0300)**

Module PFMD0300 sets up an area model, pseudo-inserts all segments into the model, and generates all the area description, freespace and segment information records required by the reporting module. Two output files are created. The first contains the area and freespace information. The second contains the segment information records.
Input requirements

The default value for BLKSIZE is the maximum block size of the output device (or half-track for 3380s). Standard labels must be used.

Region size requirements vary depending on the number of DOVF CIs in a UOW and the number of IOVF CIs in the area.

**Print modeler reports (PFMD0500)**

Module PFMD0500 formats the data from the PFMD0300 process and generates the Fast Path Analyzer/EP reports.

**Input requirements**

Input data to the Area Change Modeling Utility can be derived from one of the following sources:

- The sorted data set of segment information records that is generated by the Fast Path Online Analyzer/EP or Fast Path Analyzer/EP analysis function by using the MODEL_DDNAME keyword. The MODEL_DDNAME keyword is used to specify the ddname of a DD statement to which the information is to be written. Because this type of file consists only of concatenated keys, it is smaller and typically quicker to process than an unload file.

- A DEDB unload file in HD Unload format that is created by executing the UNLOAD command or the FABEUR6 utility

- An unload file in HD Unload format that is unloaded from an IMS full-function database. This input enables you to model the potential space utilization and performance effects of converting a full-function IMS database to the DEDB format.

---

**NOTE**

You should retain all Area Change Modeling Utility input data until database prototyping is completed.

The number of areas for which data is required depends on the database specifications being changed and the characteristics of the randomizing routine. Table 88 defines the input data requirements according to various database specification changes you can model.
Randomizer considerations

Randomizers are classified as follows:

- Nonspecific
  A randomizer where area selection is not controlled by key values. DBFHDC40, as delivered with IMS Fast Path, is nonspecific.

- Area-specific
  A user-written/modified randomizer that controls area (or area group) selection based on root segment key ranges or some similar technique.

--- EXAMPLE ---
Keys A through L are randomized across areas 1 through 5. Keys M through Z are randomized across areas 6 through 10.

The randomizer module interface that is created by module PFMD0100 conforms to the standard IMS 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 addresses of the ESCD. Some user-written or user-customized randomizers have been designed to use these addresses to gain access to IMS control blocks other than those passed as part of the published interface.

  When PFMD0100 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.

- Some user-written or user-customized randomizers are designed to use the pre-chained save-area set that is provided by IMS. The depth of this pre-chained save-area set is variable across IMS releases and is subject to the local modifications that are applied by a site’s IMS support personnel.
When module PFMD0100 issues a call to a randomizing module, a save-area set with a depth of 2 is provided. On entry to the randomizer, R13 points to a save-area for use by the randomizer with one more save-area chained below it.

**Process flow**

*Figure 33* shows the process flow of the three modules and interim sort steps for the Area Change Modeling Utility. Step names used in *Figure 33* correspond to those used in the JCL example shown on *Figure 35*.

For large databases, BMC Software recommends that you request the output for only one of the areas being modeled. This is primarily due to the potential size of the files created and the sort work data set requirements. If the modeled area meets requirements, the procedure can be repeated for the remaining areas.

---

**NOTE**

If Area Change Modeling Utility output is requested for one area, the SORT step for segment information records will be bypassed automatically.
Figure 33  Area Change Modeling Utility process flow
PFMD0100 JCL requirements

Execute PFMD0100 as a standard MVS job step. An EXEC statement and DD statements that define inputs and outputs are required. Table 89 lists the DD statements that are required depending on the type of input. Detailed descriptions of all available DD statements follow.

Table 89  PFMD0100 DD statements that are required depending on file type

<table>
<thead>
<tr>
<th>Input file type</th>
<th>Required DD statements for input file type</th>
<th>Required DD statements for all input file types</th>
</tr>
</thead>
<tbody>
<tr>
<td>data set that is generated by the Fast Path Online Analyzer/EP or Fast Path Analyzer/EP analysis function with the MODEL_DDNAME keyword</td>
<td>MEDARI</td>
<td>STEPLIB SYSIN SYSPRINT ACBLIB MERRORC MDARO</td>
</tr>
<tr>
<td>DEDB unload file in HD Unload format that is created by executing the Fast Path Reorg/EP UNLOAD command or the FABEUR6 utility</td>
<td>UR7DATA UR7RPT</td>
<td></td>
</tr>
<tr>
<td>unload file in HD Unload format from an IMS full-function database</td>
<td>DBDLIB UR7DATA UR7RPT UR7CTL</td>
<td></td>
</tr>
</tbody>
</table>

EXEC

This statement must be in the following format:

```
//PFMD0100 EXEC PGM=PFMD0100,REGION=0M
```

STEPLIB DD

Required. The following data sets must be concatenated to this DD statement:

- IMS RESLIB
- the load library in which the Fast Path/EP programs reside
- the load library containing the randomizer used for the database

SYSPRINT DD

Required. Defines the output message and statistics data set. The data set can reside on a direct access device or printer, or be routed through the output stream. Use RECFM=FBA and LRECL=121.
ACBLIB DD

Required. Defines the library containing the DMB that describes the database configuration to be modeled. This would normally be a testing library (not the production IMSVS.ACBLIB).

DBDLIB DD

Required when the UR7DATA DD statement is referring to an unload file in HD Unload format from an IMS full-function database. Defines the library containing the DBD that describes the IMS full-function database to be modeled. The full-function DBD name that is contained in the DBDLIB DD statement must be the same as the DEDB DBD name that is contained in the ACBLIB DD statement.

SYSIN DD

Required. Defines the input control statement data set. This data set can reside on a direct access device or card reader, or be routed through the input stream.

MEDARI DD

Optional. Defines the input data set that is generated by the Fast Path Online Analyzer/EP or Fast Path Analyzer/EP analysis function with the MODEL_DDNAME keyword. Multiple data sets can be concatenated in any order.

UR7DATA DD

Optional. Defines one of the following unload file types:

- a DEDB unload file in HD Unload format that is created by executing the Fast Path Reorg/EP UNLOAD command or the FABEUR6 utility
- an unload file in HD Unload format from an IMS full-function database.

Multiple data sets can be concatenated in any order.

UR7RPT DD

Required with UR7DATA. Defines the audit report output data set. Use RECFM=FBA and LRECL=121.

UR7CTL DD

Required when the UR7DATA DD statement is referring to an unload file in HD Unload format from an IMS full-function database. Defines the input control statement specifying the DBD name. In addition to the SYSIN control cards, you must also specify the following on the UR7CTL DD statement:
MESORTCD DD

Required. Defines the output data set containing the sort control statements required for the SORT13 step. This data set must reside on a direct access device. Standard labels must be used for this data set. Use RECFM=FB, LRECL=80, and BLKSIZE=800.

MEDARO DD

Defines the output data set for the segment information records used as input to the SORT13 step. The data set can reside on either a direct access device or tape. Standard labels must be used for this data set. Space requirements vary depending on the number of segments in the modeled area, and the length of the root key.
(Approximation: size in bytes = number of segments x (26 + root key length + 4 x no. hierarchical levels in DMB)).

PFMD0100 control statements

Use a mandatory control statement (CTL) to specify the name of the database and areas to be modeled. Optionally, provide the name of the randomizing routine to be used. An optional control statement (SEG) can be provided to specify the insert limit count for each segment type.

The format of the control statements for PFMD0100 is shown in the syntax diagram in Figure 34. Required elements are underlined.
You should code keywords and their associated parameter values in free format (columns 1-72) in the form of 80-byte statement images. Make certain that the following syntax rules are observed:

- Start all control statement type keywords (CTL and SEG) in column 1. Multiple lines are allowed if each line begins in column 1, begins with CTL, and is followed by additional keywords.

- You can provide multiple SEG control statements.

- The keyword=value specifications cannot span control statements.

- Separate all keywords from their associated parameter values by an equal sign. The equal sign must not be preceded by blanks, but can be followed by one or more blanks.
Separate parameter values from the next keyword by one or more blanks, a comma, or a comma followed by one or more blanks.

If duplicate keywords are used, the value specified on the last one is the value that will be used.

**CTL control statement**

This is a *required* control statement type identifier. This value must start in column 1.

**DBDNAME**

Required. Specifies the DBD name of the DEDB being modeled.

**AREA**

Required. Specifies the number of the areas to be modeled. Data analysis records and reports will be generated only for the areas specified.

Only one format of the AREA keyword can be specified per execution. The area numbers must be within the range 1 to 2048 and must exist within the DMB. Available parameters for the AREA keyword are described in Table 90. For detailed syntax requirements for the AREA keyword, refer to Figure 34.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREA=ALL</td>
<td>specifies that the data analysis records and reports are to be generated for all areas of the database</td>
</tr>
<tr>
<td>AREA=nnnn</td>
<td>specifies the area number of the requested area</td>
</tr>
<tr>
<td>AREA=(aaaa,bbbb,cccc)</td>
<td>specifies the area numbers of specific requested areas. Area numbers do not have to be contiguous. Parentheses are required.</td>
</tr>
<tr>
<td>AREA=(aaaaa-zzzz)</td>
<td>specifies a contiguous range of specific requested areas. Parentheses are required.</td>
</tr>
<tr>
<td>CISIZE</td>
<td>Optional second subparameter for AREA keyword. Specifies the control interval size. This subparameter can be specified for all areas processed, or for each individual area.</td>
</tr>
</tbody>
</table>
Table 90  PFMD0100 control statement AREA keyword parameters (part 2 of 2)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOW=(x,y)</td>
<td>Optional second subparameter for AREA keyword. The first value (x) specifies the number of blocks in each unit of work (UOW); the second value (y) specifies the number of dependent overflow (DOVF) blocks in each UOW. This subparameter can be specified for all areas processed, or for each individual area.</td>
</tr>
<tr>
<td>ROOT=(x,y)</td>
<td>Optional second subparameter for AREA keyword. The first value (x) specifies the number of UOWs in the area. The second value (y) specifies the number of independent overflow (IOVF) UOWs in the area. This subparameter can be specified for all areas processed, or for each individual area.</td>
</tr>
</tbody>
</table>

**RMOD**

Optional. Specifies the name of the randomizing routine to be used when it differs from that specified in the DMB. It is typically used when various randomizers are being tested to avoid the need for DBD and ACB generation for each of the randomizers. The randomizer is always loaded from STEPLIB.

**NOTE**

This RMOD name will always override the name specified in the DMB.

**RPT**

Optional. Specifies that the detailed record count report is to be printed. Valid values are YES and NO. (Default is NO.)

**SEG control statement**

Optional control statement type indicator. This value must start in column 1.

**NAME**

Required. Specifies the name of the segment to which the insert limit count is to be applied.

**LIMCT**

Specifies the insert limit count, which is the number of segments for each area that will be placed in root-addressable storage before the utility will place segments in other storage locations.
The value specified can be up to 4 digits with a maximum value up to 9999, and is not required to be right-justified.

**PFMD0100 return codes**

PFMD0100 writes numbered messages to the SYSPRINT data set that more fully explain the result of program execution. The return codes are described in Table 91.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The requested operation completed successfully.</td>
</tr>
<tr>
<td>4</td>
<td>Warning messages were issued, but the requested operation was completed.</td>
</tr>
<tr>
<td>8</td>
<td>Severe errors causing job termination occurred.</td>
</tr>
</tbody>
</table>

**SORT13**

This step sorts the extracted records from program module PFMD0100. Input is the data set created from MEDARO DD. Use the sort control cards created from MESORTCD DD.

The following DD statements are required to define input and output:

**SORTIN DD**

Input data set from MEDARO DD in PFMD0100.

**SORTOUT DD**

Same data set name (sort is executed in-place), passed to MEDARI DD in PFMD0300.

**SYSIN DD**

Sort control cards from MESORTCD DD in PFMD0100.

**PFMD0300 JCL requirements**

PFMD0300 is executed as a standard MVS job step. An EXEC statement and DD statements that define inputs and outputs are required.
EXEC

This statement must be in the following format:

```
//PFMD0300  EXEC  PGM=PFMD0300,REGION=0M,COND=(4,LT),
    PARM='mode'
```

Valid values for PARM are L and R. L (the default) indicates that load-mode emulation is desired. R specifies the emulation of a reorganization. If PARM is not specified or if any value other than L or R is specified, a default specification of L is will be used.

STEPLIB DD

Defines the load library in which the Fast Path Analyzer/EP programs reside.

SYSPRINT DD

Defines the output message and statistics data set.

MEDARI DD

Defines the input data set containing the sorted area and segment information records generated by PFMD0100 and sorted by step SORT13.

MEDAR12 DD

Defines the output data set for the area and freespace information records, and will be input to step SORT12. The data set can reside on either a direct access device or tape. Standard labels must be used for this data set.

Space requirements vary depending on the number of CIs in the modeled areas and the number of intersection references. (Approximation: number of CIs x .75 x 32 bytes.)

**NOTE**

Number of CIs = UOW part 1 x ROOT part 1

SORT12 DD

Defines the output data set containing the sort control statements required for Step SORT12. This data set must reside on a direct access device. Standard labels must be used for this data set. Use RECFM=FB, LRECL=80, and BLKSIZE=800.
MEDAR3 DD

Defines the output data set for the segment information records, and can be input optionally to the SORT3 step. Standard labels must be used for this data set.

Space requirements vary depending on the hierarchical structure of the database, and the number of segments in the modeled areas. (Approximation: number of segments x (26 + 4 x no. hierarchical levels in DMB).)

SORT3 DD

Defines the output data set containing the sort control statements required for Step SORT3. This data set must reside on a direct access device. Standard labels must be used for this data set. Use RECFM=FB, LRECL=80, and BLKSIZE=800.

PFMD0300 return codes

This program writes numbered messages to the SYSPRINT data set that more fully explain the result of program execution. The return codes are described in Table 92.

Table 92 PFMD0300 return codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The program completed successfully. SORT3 is not required.</td>
</tr>
<tr>
<td>2</td>
<td>The program completed successfully. SORT3 is required.</td>
</tr>
<tr>
<td>4</td>
<td>Warning messages were issued, but the requested operation was completed. SORT3 is not required.</td>
</tr>
<tr>
<td>6</td>
<td>Warning messages were issued, but the requested operation was completed. SORT3 is required.</td>
</tr>
<tr>
<td>8</td>
<td>Errors causing job termination occurred.</td>
</tr>
</tbody>
</table>

SORT12

The SORT12 step sorts the area and freespace information records produced by the MEDAR12 DD in PFMD0300. Use the sort control cards created from SORT12 DD.

The following DD statements are required to define input and output:

SORTIN DD

Input data set from MEDAR12 DD in PFMD0300.
SORTOUT DD

Same data set name (sort is executed in-place), passed to DAR12 DD in PFMD0500.

SYSIN DD

Sort control cards from SORT12 DD in PFMD0300.

SORT3

The SORT3 step sorts the segment information records data set only when multiple areas are processed. When only one area is processed, the data is sorted internally in module PFMD0300 and this step will be bypassed if you code the COND parameter on the EXEC statement as shown in the following example.

Code the EXEC statement in the following format:

```
//SORT3    EXEC PGM=SORT,REGION=0M,PARM='CORE=MAX',
         // COND=((0,EQ,PFMD0300),(4,EQ,PFMD0300),
         // (7,LE,PFMD0300))
```

The return codes from PFMD0300 determine whether this step is necessary. For consistency, BMC Software recommends that you always code this step in the JCL, thereby making its execution dependent on the condition (return) codes from PFMD0300.

The following DD statements are required to define input and output:

SORTIN DD

Input data set from MEDAR3 DD in PFMD0300.

SORTOUT DD

Same data set name, passed to DAR13R DD in PFMD0500.

SYSIN DD

Sort control cards from SORT3 DD in PFMD0300.
PFMD0500 JCL requirements

PFMD0500 is executed as a standard MVS job step. An EXEC statement and DD statements that define inputs and outputs are required.

EXEC

This statement must be in the following format:

```
//PFMD0500 EXEC PGM=PFMD0500,REGION=0M,COND=(8,LT)
```

STEPLIB DD

Defines the load library in which the Fast Path Analyzer/EP programs reside.

DAR12 DD

Defines the input data set containing freespace and area information for report generation that was sorted by step SORT12.

DAR13R DD

Defines the input data set containing segment information for report generation. This data set must be sorted prior to execution of PFMD0500. If a single area is used as input, then PFMD0300 will sort the data set. If multiple areas are used as input, then SORT3 will sort the data set.

SYSPRINT DD

Defines the output message and statistics data set. The data set can reside on tape, direct access device or printer, or be routed through the output stream. Use RECFM=FBA and LRECL=121.

REPORTS DD

Defines the output analysis reports data set. The data set can reside on tape, direct access device or printer, or be routed through the output stream. Use RECFM=FBA and LRECL=133.

UOWRPT DD

Defines the optional output data set for the UOW report. This data set can reside on tape, direct access device or printer, or be routed through the output stream. Use RECFM=FBA and LRECL=133.
PFMD0500 return codes

PFMD0500 writes numbered messages to the SYSPRINT data set that more fully explain the results of program execution. The return codes are described in Table 93.

Table 93 PFMD0500 return codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The requested operation completed successfully.</td>
</tr>
<tr>
<td>4</td>
<td>There are no database records in the area. The free space report generation completed successfully.</td>
</tr>
</tbody>
</table>

Area Change Modeling Utility sample JCL

Figure 35 shows sample JCL that executes the three modules and sort steps for the Area Change Modeling Utility. This JCL is included in the Fast Path/EP sample library in the member named PFUMODEL.

Figure 35 Sample Area Change Modeling Utility JCL (part 1 of 3)

```plaintext
//PFPMODEL JOB (ACCT),'PFP',MSGCLASS=A,CLASS=A        <== CHANGE
//PFMD0100 EXEC PGM=PFMD0100,REGION=0M
//************************************
//** REPLACE AREA AND RAP NUMBERS **
//************************************
//STEPLIB  DD DSN=BMC.PFP.LOAD,DISP=SHR           <== CHANGE
//         DD DSN=IMSVS.RESLIB,DISP=SHR           <== CHANGE
//SYSPRINT DD SYSOUT=*                             <== CHANGE
//ACBLIB   DD DSN=BMC.PFP.ACBLIB,DISP=SHR         <== CHANGE
//************************************
//**SPECIFY THE FOLLOWING DD FOR INPUT FROM FAST PATH ANALYZER/EP
//************************************
//MEDARI   DD DSN=BMC.PFP.DA13R,DISP=SHR           <== CHANGE
//************************************
//**SPECIFY THE FOLLOWING DDs FOR INPUT FROM FAST PATH UNLOAD FILE
//************************************
//UR7DATA DD DSN=BMC.PFP.UNLOAD,DISP=SHR           <== CHANGE
//UR7RPT  DD SYSOUT=*                             <== CHANGE
//************************************
//MESORTCD DD DSN=&&SORT13,DISP=(NEW,PASS),       <== CHANGE
//        UNIT=SYSDA,SPACE=(TRK,(1,1))
//MEDARO  DD DSN=BMC.PFP.ME13,                    <== CHANGE
//        DISP=(NEW,CATLG),
//        UNIT=SYSDA,SPACE=(CYL,(35,5),RLSE)
//SYSIN    DD *,DCB=BLKSIZE=80                    <== CHANGE
/*
```
Figure 35  Sample Area Change Modeling Utility JCL (part 2 of 3)

```
//SORT13 EXEC PGM=SORT,REGION=0M,PARM='CORE=MAX',
   COND=(4,LT)
******************************************************************************
//** SORT ME13 RECORDS **
******************************************************************************
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=* 
//SYSIN DD DSN=&&SORT13,DISP=(OLD,DELETE)
//SORTIN DD DSN=BMC.PFP.ME13,DISP=SHR  
发出 CHANGEM
//SORTOUT DD DSN=BMC.PFP.ME13,DISP=SHR  
发出 CHANGEM
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(2,3))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(2,3))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(2,3))
/*
//PFMD0300 EXEC PGM=PFMD0300,REGION=0M,COND=(4,LT)
******************************************************************************
//** BUILD AREA MODEL **
******************************************************************************
//STEPLIB DD DSN=BMC.PFP.LOAD,DISP=SHR  
发出 CHANGEM
//SYSPRINT DD SYSOUT=* 
//MEDARI DD DSN=BMC.PFP.ME13,
   DISP=(OLD,DELETE,KEEP)
//MEDAR12 DD DSN=BMC.PFP.DA12,
   DISP=(NEW,CATLG,DELETE),
   UNIT=SYSDA,
   SPACE=(CYL(15,2),RLSE)
//MEDAR3 DD DSN=BMC.PFP.DA3,  
发出 CHANGEM
   DISP=(NEW,CATLG,DELETE),
   UNIT=SYSDA,
   SPACE=(CYL(50,5),RLSE).
//SORT12 DD DSN=&SORT12,  
发出 CHANGEM
   DISP=(NEW,PASS),
   UNIT=SYSDA,
   SPACE=(TRK,(1,1))
//SORT3 DD DSN=&SORT3,DISP=(NEW,PASS),
   UNIT=SYSDA,SPACE=(TRK,(1,1))
/*
//SORT12 EXEC PGM=SORT,REGION=0M,PARM='CORE=MAX',
   COND=(4,LT)
******************************************************************************
//** SORT MODELER DA12 RECORDS **
******************************************************************************
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=* 
//SYSIN DD DSN=&SORT12,DISP=(OLD,DELETE)
//SORTIN DD DSN=BMC.PFP.DA12,DISP=SHR  
发出 CHANGEM
//SORTOUT DD DSN=BMC.PFP.DA12,DISP=SHR  
发出 CHANGEM
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(2,3))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(2,3))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(2,3))
/*
```
User procedures

To use the Area Change Modeling Utility, perform the following steps:

1 Using the Fast Path Analyzer/EP analysis reports for the existing database, determine the new database specifications to be used.

2 Code the appropriate control cards. When executing the Area Change Modeling Utility, a new DMB is not required to model changes that are made to the randomizing routine, CI size, UOW or root parameters.

3 Determine the required input for the Area Change Modeling Utility.

**NOTE**
Unless an exact model is required, it does not matter whether the data for all required areas is created at the same time.
4 Execute the Area Change Modeling Utility to request the areas of interest to be modeled.

5 Evaluate the results.

6 Repeat steps 1 through 5 until the desired space utilization and performance characteristics are obtained.

7 As a final check, execute the Area Change Modeling Utility to request a sampling of other areas or all areas.

For sample scenarios that show the use of the Area Change Modeling Utility, see Appendix H, “Sample utility and command scenarios.”

User procedures for full-function databases

When an IMS full-function database is used as input to the Area Change Modeling Utility, the procedure is similar to modeling DEDB areas.

To use full-function databases

1 Create DBD and ACB members for the DEDB. The full-function DBD name must be the same as the DEDB name. It is recommended that you use a test DBD and ACB libraries while modeling your DEDB.

   TIP
   Additional space will be required for your DEDB areas, which require more space for faster processing.

   TIP
   For a DEDB, the number of RAPs per CI in RAA is always 1.

2 Code the appropriate control cards. A new DMB is not required to model changes that are made to the randomizing routine, CI size, UOW, or root parameters.

3 Provide the full-function HD Unload file as input to the Area Change Modeling Utility.

4 Execute the Area Change Modeling Utility to request the areas of interest to be modeled.
Comparing randomizing routines

5 Evaluate the results. The Freespace Analysis information that is contained in the PFMD0500 report output can help you to determine if your data is stored in the manner that you expected. A good practice is to have as much data as possible residing in RAA base storage, and less data residing in overflow (DOVF and IOVF).

6 Repeat steps 1 through 5 until the desired space utilization and performance characteristics are obtained.

For sample scenarios that show the use of the Area Change Modeling Utility, see Appendix H, “Sample utility and command scenarios.”

Comparing randomizing routines

To effectively compare randomizing routines, none of the other database specifications should be changed when executing the Area Change Modeling Utility.

To compare randomizing routines

1 Determine the required input for the Area Change Modeling Utility.

NOTE

Unless an exact model is required, it does not matter if the data for the areas being modeled is created at the same time.

2 Execute the modeling procedure to request the areas of interest to be modeled.

3 Evaluate the results.

4 Repeat steps 2 and 3 until the desired space utilization and performance characteristics are obtained.

5 As a final check, execute the modeling procedure requesting a sampling of other areas, or all areas.

For sample scenarios that show the use of the Area Change Modeling Utility, see Appendix H, “Sample utility and command scenarios.”
Area Change Modeling Utility sample scenarios

Table 94 lists several sample scenarios that are available in Appendix H, “Sample utility and command scenarios.” These scenarios show how to code control statements for the PFMD0100 control program to perform various modeling functions.

<table>
<thead>
<tr>
<th>Utility/ scenario task</th>
<th>Control statement/keyword</th>
<th>Concept/process</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area Change Modeling Utility (PFMD0100 Program)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model Effects of Changing Randomizer for All Areas</td>
<td>CTL control statement:</td>
<td>■ specify a different randomizer than the randomizer</td>
<td>639</td>
</tr>
<tr>
<td></td>
<td>DBDAME / AREA RPT RMOD</td>
<td>currently specified in DBD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ request report reflecting detailed record count</td>
<td></td>
</tr>
<tr>
<td>Area Change Modeling Utility (PFMD0100 Program)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model Effects of Changing UOW and Root Values for</td>
<td>CTL control statement:</td>
<td>■ increase ROOT and UOW values for specific areas</td>
<td>640</td>
</tr>
<tr>
<td>Selected Areas</td>
<td>DBDNAME / AREA ROOT UOW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area Change Modeling Utility (PFMD0100 Program)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model Effect of Changing Control Interval Size for an Area</td>
<td>CTL control statement:</td>
<td>■ change size of control interval for one area</td>
<td>641</td>
</tr>
<tr>
<td></td>
<td>DBDNAME / AREA CISIZE</td>
<td>■ limit the number of segments to be placed in RAA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SEG control statement:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NAME LIMITCT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area Change Modeling Utility (PFMD0100 Program)</td>
<td>DBD control statement:</td>
<td>■ change ROOT and UOW values for selected area</td>
<td>642</td>
</tr>
<tr>
<td>Model Effects of Converting from Full-Function to DEDB</td>
<td>DBDNAME / AREA UOW ROOT</td>
<td>■ change control interval size for two areas</td>
<td></td>
</tr>
<tr>
<td>Format</td>
<td>CISIZE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DMAC Print utility (DMAC_PRINT command)

The DMAC Print utility is a command-driven function used to print the contents of the DMAC block for an area. The function can be used in either online or offline mode.
When DMAC_PRINT executes in offline mode, the input to the function can be the area data set or an image copy. Specify the image copy or area data set name on the areaname DD statement. The DMAC_PRINT utility determines which input is specified. The utility prints the contents of the DMAC constructed by merging the values from the DMB (from the ACB library) with the DMAC block from the area data set or image copy.

**Specifying the database and areas**

The DBD keyword identifies the name of the DEDB (DBD name) for which a DMAC block or blocks will be printed. The DBD keyword is required when executing the DMAC Print utility in offline mode.

For online execution, 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 execution of the DMAC Print utility, 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 when executing the DMAC Print utility in online mode, an error message is issued and the function terminates.

The IAREA keyword can be used to select specific areas for which DMAC blocks will be printed. If you omit the IAREA keyword, DMAC blocks will be printed for all areas 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=*** – Specify all areas of the DEDB.
- **IAREA=areaname** – Specify 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** – Specify 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))** – Use this syntax to specify 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.

Offline and online examples

In the example in Figure 36, the area data sets and ACB library are dynamically allocated. DMAC blocks will be printed for all areas in the DEDB.

**Figure 36 Using the offline DMAC_PRINT command to print DMAC blocks for all areas**

```
//PFP EXEC PGM=PFPMAIN,REGION=0M, REGION=OM
//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
// DD DISP=SHR,DSN=IMS.RESLIB
//PFPSYSIN DD *
 DMAC_PRINT DBD=dbdname
/*
```

When DMAC_PRINT executes in online mode, the input to the function is the online area data set. The utility prints the contents of the DMAC in use by the IMS control region. This DMAC is often referred to as “the in-core DMAC.”

An example of an online DMAC_PRINT is shown in Figure 37. This control statement uses a combination of area names and area numbers to request that DMAC blocks are to be printed for each of the specified areas. The DBD keyword is not specified (not required) because the DBD name is specified as an execution parameter. DMAC blocks will be printed for the areas named AREANAM1 and AREANAM3, and for all consecutive areas from area number 5 to area number 8 (area5, area6, area7 and area8).

**Figure 37 Using the online DMAC_PRINT command to print DMAC blocks for specified areas**

```
//PFP EXEC PGM=DFSRRC00,REGION=0M,
// PARM=(IFP,dbdname,DISCON,DISCON)
//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
// DD DISP=SHR,DSN=IMS.RESLIB
//PFPSYSIN DD *
 DMAC_PRINT IAREA=(AREANAM1,AREANAM3,RANGE=(5,8))
/*
```
Control Interval Dump and Modification Utility (PROCESS_AREA command)

The Control Interval Dump and Modification Utility is a command-driven utility that provides an enhanced, Fast Path-customized version of a conventional VSAM snap/dump utility. This utility provides a quick, efficient means to accomplish the following tasks:

- dump (snap), inspect, or modify the entire contents of a database control interval (CI)
- verify and repair pointer errors and VSAM control information, even when standard VSAM utilities do not provide access to the data
- print the contents of the DMAC block or DMCB block for an area

When the online or offline analysis function is executed, segment pointers, FSE chains, IMS and VSAM control fields, and space utilization are verified and discrepancies are reported. If discrepancies are found, the Control Interval Dump and Modification utility can be used to dump the CIs involved so that segments in error can be located and the proper values can be determined. When you are ready to repair the pointer values, the utility can be used to make and commit the required changes.

The Control Interval Dump and Modification Utility is executed by specifying the PROCESS_AREA command and one or more PERFORM subcommands. A script can be specified on each PERFORM subcommand to indicate the action desired. A script consists of one or more function calls, each which performs a specific desired task.

The PROCESS_AREA command can be used in either online or offline mode. When PROCESS_AREA executes in offline mode, the input to the function can be the area data set or an image copy. Specify the image copy or area data set name on the areaname DD statement. The utility determines which input is specified.

**PROCESS_AREA keywords and subcommands**

The keywords and subcommands available for the PROCESS_AREA command are shown in Table 95.
Selecting the database and areas

The DBD keyword identifies the name of the DEDB (DBD name) to be dumped, modified or printed. The DBD keyword is required when executing the Control Dump and Modification utility in offline mode.

For online execution, 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 execution of the Control Interval Dump and Modification utility, 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 when executing the Control Interval Dump and Modification utility in online mode, an error message is issued and the function terminates.

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

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=*** – Specify all areas of the DEDB.
- **IAREA=areaname** – Specify 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** – Specify 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.

### Table 95  PROCESS_AREA 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>PROCESS_AREA</td>
<td>DBD, IAREA</td>
</tr>
<tr>
<td>allocating the area data set</td>
<td>PROCESS_AREA</td>
<td>INPUT_DSN_MASK</td>
</tr>
<tr>
<td>specifying a script to execute a control interval dump, modification, or print function</td>
<td>PERFORM</td>
<td>SCRIPT with function(s)</td>
</tr>
</tbody>
</table>

![Table 95 PROCESS_AREA command keywords and subcommands](image)
IAREA=(RANGE=(startarea, endarea)) – Use this syntax to specify 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 process an entire DEDB, use a command set like the example shown in Figure 38.

**Figure 38  Sample control statement for processing all areas**

```
PROCESS_AREA DBD=dbdname,IAREA=ALL
```

To process specific areas within a DEDB, use a command set like the example shown in Figure 39.

**Figure 39  Sample control statement for processing specific areas**

```
PROCESS_AREA 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 processed. The UOWs in areas named AREANAM1 and AREANAM3 will be processed. Also, all consecutive areas from area number 5 to area number 8 (area5, area6, area7 and area8) will be processed.

---

**Offline considerations**

The following considerations outline specific considerations related to executing the Control Interval Dump and Modification Utility in offline mode.

**DBRC considerations**

When DBRC is active during the execution of the Control Interval Dump and Modification Utility in offline mode, and the area is registered with DBRC, then the data set name for the area data set must match the name registered with DBRC. If any script contains update processing, then an exclusive (EX) authorization level is requested. If no update processing is specified, then protected read (RD) authorization level is requested.
**MADS considerations**

When executing the Control Interval Dump and Modification Utility in offline mode, multiple area data sets (MADS) are not supported. The function searches the ADS (area data set) list 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 that will be processed. If no area data set is marked as available, or if all available area data sets contain one or more EQEs, the Control Interval Dump and Modification function will not be performed for that area.

If the script contains any update processing, all other area data sets are marked unavailable. After the function executes, use the IBM Online MADS Create utility to resynchronize the other (unavailable) area data sets.

**Allocating the area data set**

The areaname DD statement identifies the area data set to be processed by the utility. The areaname DD statement can be the area data set or an image copy. If the areaname DD statement is omitted from the JCL, Fast Path Analyzer/EP tries to dynamically allocate it.

When a compressed image copy is used as input, it will be decompressed automatically. BMC Software recommends that processing be performed in ascending RBA sequence. If not, the utility will rewind the image copy input file as needed. Updates cannot be made to image copy data sets; processing is read-only.

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

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

1. If the INPUT_DSN_MASK keyword is specified, it is used to generate the data set name. The data set name can specify an image copy.

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 DBRC is active, the area is registered with DBRC, and the data set to be processed is not an image copy, the allocated data set name must match the registered data set name, regardless of how it is allocated.
Offline control statements

The PFPSYSIN control statements include the command set necessary to run the Control Interval Dump and Modification Utility. A sample offline command set is shown in Figure 40. In this example, the area data set and ACB library are accessed using dynamic allocation. Processing is limited only to the area specified on the IAREA keyword.

Figure 40  JCL example for offline area data set input

```
//PFP      EXEC PGM=PFPMAIN,REGION=0M
//STEPLIB  DD   DSN=BMC.PFP.LOAD,DISP=SHR
//         DD   DSN=IMS.RESLIB,DISP=SHR
//PFPSYSIN DD   *
PROCESS_AREA DBD=dbname, IAREA=areaname
   PERFORM SCRIPT={ script }
/*
```

The JCL for executing the utility against an image copy is shown in Figure 41. In this example, the input image copy and ACB library are accessed using dynamic allocation. Processing will be limited only to the area(s) specified on the IAREA keyword.

Figure 41  JCL example for image copy input

```
//PFP  EXEC  PGM=PFPMAIN,REGION=0M
//STEPLIB  DD  DSN=BMC.PFP.LOAD,DISP=SHR
//         DD  DSN=IMS.RESLIB,DISP=SHR
//PFPPRINT DD  SYSOUT=* 
//PFPSYSIN DD  *
PROCESS_AREA DBD=dbname, IAREA=areaname,
   INPUT_DSN_MASK='PFP.ICOPY.&DBD.&AREA'
   PERFORM SCRIPT={ script }
/*
```

Online control statement

The PFPSYSIN control statements include the command set necessary to run the Control Interval Dump and Modification Utility. A sample online command set is shown in Figure 42. Processing will be limited only to the area(s) specified on the IAREA keyword.

Figure 42  JCL example for online area data set input (part 1 of 2)

```
//PFP EXEC PGM=DFSRRC00,REGION=0M,
   PARM=(IFP,dbname,DBF#FPU0)
//STEPLIB  DD  DSN=BMC.PFP.LOAD,DISP=SHR
```
Available functions

The Control Interval Dump and Modification Utility can perform several tasks that are typically available in a conventional snap/dump utility. The SCRIPT keyword is coded on the PERFORM subcommand to control the processing that is performed to examine and modify data within the area data set or to print information. Each task is executed by specifying one or more functions on the SCRIPT keyword. Available functions are shown in Table 96.

Table 96  Available functions for control interval dump and modification

<table>
<thead>
<tr>
<th>Function</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNAP</td>
<td>dump (snap) the contents of a single RBA or range of RBAs.</td>
</tr>
<tr>
<td>VER</td>
<td>read the requested control interval (CI) and verify the data within the CI as specified on the control statement.</td>
</tr>
<tr>
<td>REP</td>
<td>replace data in the CI. The utility produces audit information when data modification occurs.</td>
</tr>
<tr>
<td>DMAC_PRINT</td>
<td>snap the contents from the in-core DMAC in hexadecimal, and print a formatted listing of the individual fields.</td>
</tr>
<tr>
<td>DMCB_PRINT</td>
<td>print a formatted listing of the fields within the in-core DMCB.</td>
</tr>
<tr>
<td>COMMIT</td>
<td>immediately write all updates to disk, and continue processing.</td>
</tr>
<tr>
<td>ROLLBACK</td>
<td>discard all updates (do not write to disk), and continue processing.</td>
</tr>
</tbody>
</table>

Syntax requirements

Scripts used to perform dump, print, or control interval modifications must be specified using detailed syntax. To clarify syntax requirements, Figure 43 shows a syntax diagram for functions that can be specified on a SCRIPT keyword. The functions shown in this table are discussed on page 474. See page 476 for a listing of sample scenarios that use these functions.
Dump and print functions

This section discusses required parameters for dump and print functions that are available with the Control Interval Dump and Modification Utility.

**SNAP (rba1 [,rba2 ])**

The SNAP function snaps the contents of a single control interval or a range of control intervals. The rba1 and rba2 values must be specified as hexadecimal numbers. If specified, rba2 must be greater than or equal to rba1.
DMAC_PRINT ( )

The DMAC_PRINT function snaps the contents of the DMAC in hexadecimal, and prints a formatted list of the individual fields. This list is the same report that is printed when you execute the DMAC_PRINT command under PFPMAIN. No parameters exist for this function; however, the ( ) must be specified.

DMCB_PRINT ( )

The DMCB_PRINT function prints a formatted list of the individual fields within the DMCB. No parameters exist for this function; however, ( ) must be specified.

Control Interval Modification functions

The required parameters for control interval modification functions available with the Control Interval Dump and Modification Utility follow.

VER ( rba , value )

The VER function verifies the contents of the area at rba using value. The rba1 and value must be specified as hexadecimal numbers. The length of value determines the length of the verification. If a mismatch occurs, the script terminates with an error.

REP ( rba , value )

The REP function is used to replace the contents of the area at rba using value. The rba1 and value must be specified as hexadecimal numbers. The length of value determines the length of the data replaced.

Two additional functions, COMMIT and ROLLBACK, are coded in conjunction with specified REP functions to either commit or discard changes.

COMMIT ( )

The COMMIT function causes all updates to be written to disk immediately, and allows processing to continue. If a script completes without an error, this action occurs automatically. No parameters exist for this function; however, the ( ) must be specified.

NOTE

If the specified RBA is not at a CI boundary, the utility rounds the RBA argument down to the first lower boundary and dump the entire CI.
ROLLBACK ( )

The ROLLBACK function causes all updates to be discarded without writing to disk and allows processing to continue. If a script terminates because of any error, this action occurs automatically. No parameters exist for this function; however, the ( ) must be specified.

Verify and replace locking considerations

Unlike other online commands that are performed by the Fast Path Online Suite products, the PROCESS_AREA command uses CI level locking. When the VER or REP functions are executed, a lock is placed on the control interval that is being processed. All locks are held until the next commit or rollback. Using VER or REP on fields within the DMAC places a lock on the in-core DMAC, which locks the entire area until the next commit or rollback.

**NOTE**

VER or REP functions work normally for SDEP data, but locks are not placed on the control interval. If block level sharing is used for the area, you cannot use the VER or REP functions for SDEP data. If you attempt to use either of these functions for SDEP data in a block level sharing environment, the script will terminate abnormally with an RBA-read/RBA-write exception.

Control Interval Dump and Modification Utility sample scenarios

Table 97 lists several sample PROCESS_AREA command scenarios that are available in Appendix H, “Sample utility and command scenarios.” These scenarios show how to use the PROCESS_AREA command to perform various functions.

<table>
<thead>
<tr>
<th>Scenario task</th>
<th>Subcommand/keyword</th>
<th>Concept/process</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESS_AREA</td>
<td>PERFORM</td>
<td>snap dump a specified control</td>
<td>636</td>
</tr>
<tr>
<td>Use the SNAP Function to Diagnose Pointer Problem</td>
<td>SNAP function</td>
<td>interval</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 5  Supporting utilities

Fast Path Online Reorg/EP and Fast Path Reorg/EP incorporate the File Sort Utility. This command-driven utility provides a method for invoking your installation’s sort utility to perform a sort of an input file in either root key or randomized sequence. This utility provides an efficient means to accomplish the following tasks:

- sort an unload file in root key sequence, which cannot be accomplished with a standard utility unload
- specify sort processing for any SDEPs contained in the unload file
- sort a file in randomized sequence in a job step that is independent of the file unload

The input to the File Sort Utility is a sequential input file for each area to be sorted. The sort function will accept any of the supported unload formats as input, and will automatically determine which format it is processing. The output will be written in HD Unload format only. The File Sort Utility can be used in offline mode only.

The File Sort Utility is executed by specifying the PFPSORT command and related keywords on the PFPSYSIN DD statement. The PFPSORT command can accept only one unload file as input and will produce only one unload file as output. If you want to sort more than one unload file in one step, you must concatenate the files on the SORTIN DD statement.
PFPSORT keywords

NOTE
In addition to the File Sort Utility, you can also use the LGBSORT feature, which offers full 64-bit storage and zIIP processor utilization.

To use LGBSORT, you must add the data set containing LGBSORT to the STEPLIB concatenation; otherwise, Fast Path/EP uses a standard sort, such as DFSORT or SYNCSORT. For more information about the STEPLIB requirements, see Chapter 1, “JCL statements.”

PFPSORT keywords

The keywords available for the PFPSORT command are shown in Table 98.

Table 98  PFPSORT command keywords and subcommands

<table>
<thead>
<tr>
<th>Function</th>
<th>Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>selecting the database and areas</td>
<td>DBD</td>
</tr>
<tr>
<td></td>
<td>OAREA</td>
</tr>
<tr>
<td>allocating the input area data set</td>
<td>INPUT_DSN_MASK</td>
</tr>
<tr>
<td>allocating the output area data set</td>
<td>OUTPUT_DSN_MASK</td>
</tr>
<tr>
<td>customizing sorting of output file</td>
<td>SORT_SEQUENCE</td>
</tr>
<tr>
<td>specifying optional sort tuning parameters</td>
<td>SORT_OPTION</td>
</tr>
</tbody>
</table>

Selecting the database and areas

The DBD keyword identifies the name of the DEDB (DBD name) to be sorted.

The OAREA keyword identifies the names of the areas to be written to the output unload file. You can specify one or more area names. If you omit the OAREA keyword, the File Sort Utility will attempt to process all areas that are defined in the DBD.

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

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

- OAREA=(RANGE=(startarea,endarea)) specifies a consecutive range of areas using either areaname or areanumber parameters. The area 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.

**Dynamically allocating input and output**

You can dynamically allocate the input for the File Sort Utility by specifying the INPUT_DSN_MASK keyword. Because the PFPSORT command can only accept one file as input, however, only one input file will be allocated. If substitution characters are used in the data set name that you specify on the INPUT_DSN_MASK keyword, the File Sort Utility will substitute the first area name or area number that is defined in the DBD.

If you want to use multiple unload files as input, do not specify the INPUT_DSN_MASK keyword. Instead, you must concatenate the files on the SORTIN DD statement.

You can dynamically allocate the output from the File Sort Utility by specifying the OUTPUT_DSN_MASK keyword. If the SORTOUT DD statement is not specified from the JCL, Fast Path Reorg/EP uses the value that you specify on the OUTPUT_DSN_MASK keyword to allocate the data set name.

**Specifying the sort sequence**

The required SORT_SEQUENCE keyword is specified under the PFPSORT command to request up to three parameters to customize the sorting of the output file. To clarify syntax requirements, “PFPSORT command” on page 571 shows the syntax that is used for the SORT_SEQUENCE keyword in a PFPSORT command set. For a description of the parameters that are used with the SORT_SEQUENCE keyword, see “SORT_SEQUENCE” on page 315.
Using **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 used at your site.

**File Sort Utility DD statements**

Several DD statements can be used with the File Sort Utility.

**IMSACB DD**

Required. Defines the library containing the DMB for the database. This DD statement is required unless the **MODSTAT DD**, **MODSTAT2 DD**, or **OLCSTAT DD** online IMS statement is provided.

**MODSTAT / MODSTAT2**

Optional. Used to identify the active online IMS ACB data set (IMSACBA or IMSACBB). Dynamic allocation can be used for these DD statements.

When one of these DD statements is present, the **MODSTAT** data set is interrogated to determine whether IMSACBA or IMSACBB is the active library. If the IMSACBA or IMSACBB DD statement is not present in the JCL, the STEPLIB/LINKLIST are searched for a DFSMDA member.

If MODSTAT2 DD is present, the active MODSTAT data set is determined prior to ACBLIB selection. The **MODSTAT2** data set is interrogated to determine whether IMSACBA or IMSACBB is the active library. If the IMSACBA or IMSACBB DD statement is not present in the JCL, the STEPLIB/LINKLIST are searched for a DFSMDA member.

If both OLCSTAT and MODSTAT DD statements are present, then OLCSTAT will be used and MODSTAT will be ignored.

**OLCSTAT DD**

Optional. Used to identify the active online IMS ACB data set (IMSACBA or IMSACBB) in an IMS global online change environment. Dynamic allocation can be used for this DD statement.
When this DD statement is present, the OLCSTAT data set is interrogated to determine whether IMSACBA or IMSACBB is the active library. If the IMSACBA or IMSACBB DD statement is not present in the JCL, the STEPLIB/LNKLIST are searched for a DFSMDA member.

If both OLCSTAT and MODSTAT DD statements are present, then OLCSTAT will be used and MODSTAT will be ignored.

**SORTIN DD**

Required. Defines the input data sets that contains the records that are to be processed by the PFPSORT command. The data set can be created in any of the following ways:

- Fast Path Reorg/EP UNLOAD command with any value specified on the FORMAT keyword
- IBM Fast Path Basic Tools for OS/390 DEDB Unload Utility with any format specified for unload file
- Fast Path Reorg/EP FABEUR6 Unload File Create Utility with any value specified on the FORMAT control statement
- Fast Path Analyzer/EP or Fast Path Online Analyzer/EP EXTRACT command with the EXTRACT_FORMAT=HDUNLOAD value specified

The PFPSORT command can accept only one unload file as input. If you want to sort more than one unload file in one step, must concatenate the files on the SORTIN DD statement. If the SORTIN DD statement is omitted from the JCL, the File Sort Utility uses the value that you specify on the INPUT_DSN_MASK keyword to allocate the data set name.

**SORTOUT DD**

Required. Defines the output data set to contain the sorted records. The output is written in HD Unload format. You can use this data set as input to any of the following:

- Fast Path Reorg/EP RELOAD command set
- Fast Path Reorg/EP FABEUR7 Read Unloaded Database Utility
- any IMS full-function utility

If the SORTOUT DD statement is omitted from the JCL, the File Sort Utility uses the value that you specify on the OUTPUT_DSN_MASK keyword to allocate the data set name.
SORTWKnn DD

Optional. Defines the sort work files that your installation’s sort/merge utility uses to sort the records. For more information, see the documentation for your installation’s sort/merge utility.

SORTMSG DD

Optional. Defines the data set where any error or warning messages will be written.

File Sort Utility sample scenarios

Table 99 lists two sample PFPSORT command scenarios that are available in Appendix H, “Sample utility and command scenarios.” These scenarios show how to use the PFPSORT command to specify customized file sorting options.

<table>
<thead>
<tr>
<th>Scenario task</th>
<th>Keyword</th>
<th>Concept/process</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFPSORT</td>
<td>SORT_SEQUENCE=KEY_ASCEND, SORT_OPTION</td>
<td>■ specify customized sort sequence for file with root key values as the sort key</td>
<td>643</td>
</tr>
<tr>
<td>Sort File in Ascending Order by Root Key</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PFPSORT</td>
<td>SORT_SEQUENCE=RAP_ASCEND, SDEPSEQ SORT_OPTION</td>
<td>■ specify customized sort sequence for file with RAP values as the sort key</td>
<td>644</td>
</tr>
<tr>
<td>Sort File in Ascending Order by RAP</td>
<td></td>
<td>■ specify sort sequence for associated SDEPs</td>
<td></td>
</tr>
<tr>
<td>Sort Logical SDEPs in Reverse Sequence</td>
<td></td>
<td>■ specify dynamic allocation of sort work files</td>
<td></td>
</tr>
</tbody>
</table>

Randomizer Interface Subroutine

The Fast Path/EP Randomizer Interface Subroutine (PFUT0B50) provides an easy to use facility for invoking a DEDB randomizing routine from a high-level language in an offline environment.

PFUT0B50 can be used to determine the area number and RAP RBA values for a root key when a user-written program generates the segment data records to load a DEDB initially. It can also be used to add physical sequence data to transactions that are to be applied in batch (BMP) mode.
Up to 16 different randomizers can be invoked when the optional fifth parameter (DBD name) is specified on CALC calls.

PFUT0B50 has been designed and written to let users migrate to new releases without modifying or re-link-editing user programs.

If PFUT0B50 is link-edited into a user program, modify the STEPLIB DD data sets to reference the data set containing the new release modules, remove any references to the old release data set, and execute the program. The new version of PFUT0B60 (the I/O module) is invoked.

**PFUT0B50 JCL requirements**

The randomizer interface subroutine is a called module. It requires the following DD statements to be included in the job step.

**STEPLIB DD**

Required. The following data sets must be concatenated to this DD statement:

- IMS RESLIB
- the program load library containing the PFUT0B50 calls
- the load library containing the randomizer module used for the database

**ACBLIB DD**

Required. Defines the library containing the DMB for the database. This DD statement is required unless the MODSTAT DD, MODSTAT2 DD, or OLCSTAT DD online IMS statement is provided.

**MODSTAT / MODSTAT2**

Optional. Used to identify the active online IMS ACB data set (IMSACBA or IMSACBB). Dynamic allocation can be used for these DD statements.

When one of these DD statements is present, the MODSTAT data set is interrogated to determine whether IMSACBA or IMSACBB is the active library. If the IMSACBA or IMSACBB DD statement is not present in the JCL, the STEPLIB/LINKLIST are searched for a DFSMDA member.

If MODSTAT2 DD is present, the active MODSTAT data set is determined prior to ACBLIB selection. The MODSTAT2 data set is interrogated to determine whether IMSACBA or IMSACBB is the active library. If the IMSACBA or IMSACBB DD statement is not present in the JCL, the STEPLIB/LINKLIST are searched for a DFSMDA member.
If both OLCSTAT and MODSTAT DD statements are present, then OLCSTAT will be used and MODSTAT will be ignored.

**OLCSTAT DD**

Optional. Used to identify the active online IMS ACB data set (IMSACBA or IMSACBB) in an IMS global online change environment. Dynamic allocation can be used for this DD statement.

When this DD statement is present, the OLCSTAT data set is interrogated to determine whether IMSACBA or IMSACBB is the active library. If the IMSACBA or IMSACBB DD statement is not present in the JCL, the STEPLIB/LNKLIST are searched for a DFSMDA member.

If both OLCSTAT and MODSTAT DD statements are present, then OLCSTAT will be used and MODSTAT will be ignored.

**PFUT0B50 parameter lists**

PFUT0B50 is called with a parameter list consisting of a function code and one or more other data items. The function code is used to indicate which function to perform.

**PFUT0B50 initialization function**

The initialization function sets up the required environment for subsequent calculation calls. The INIT function must be performed prior to a CALC call. Up to 16 INIT calls can be done for 16 different DBD names.

For COBOL, define the following in WORKING-STORAGE:

```
77 FUNCINIT PIC X(4) VALUE 'INIT'.
77 DBDNAME PIC X(8) VALUE 'aaaaaaaa'.
```

where aaaaaaaaa is the DBD name. In the PROCEDURE DIVISION, code the following:

```
call 'PFUT0B50' using funcinit,
       dbd name.
```
PFUT0B50 calculation function

The calculation function invokes the randomizing routine using the specified root key value, calculates the RAP RBA, and returns both the RAP RBA value and the area number to the caller.

For COBOL, define the following in WORKING-STORAGE:

```
77   FUNCCAL PIC X(4) VALUE 'CALC'.
77   AREANO PIC S9(4) COMP.
77   RAPRBA PIC S9(8) COMP.
77   DBDCALC PIC X(8) VALUE 'bbbbbbbb'
01   SEG-RECORD.
05   SEG-ROOTKEY PIC X(nn).
```

where:

- \( nn \) is the root key length.
- \( bbbbbbbbb \) is the DBD name for the CALC call.

In the PROCEDURE DIVISION, code the following:

```
CALL 'PFUT0B50' USING FUNCCALC,
    SEG-ROOTKEY,
    AREANO,
    RAPRBA.
    (DBDCALC).
```

**NOTE**

If an INIT call was performed for only one DBD name, the DBDCALC parameter can be omitted.

Randomizer module interface exceptions

The randomizer module interface environment created by PFUT0B50 conforms to the published interface with the following exceptions:

- When a randomizer is invoked by IMS, registers 10 and 11 contain the addresses of the EPST and ESCD, respectively. Some user-written or user-customized randomizers have been designed to use these addresses to gain access to IMS/VSE control blocks other than those passed as part of the published interface.
When Fast Path Reorg/EP issues a call to a randomizing module registers 10 and 11 will be set to -1 and 0, respectively, to indicate that the call is not being issued in a live IMS environment.

Some user-written or user-customized randomizers are designed to use the pre-chained save-area set provided by IMS/VS. The depth of this pre-chained save-area set is variable across IMS/VS releases and subject to the local mods applied by an installations IMS support personnel.

When PFUT0B50 issues a call to a randomizing module, a save-area set with a depth of 2 is provided; on entry to the randomizer, R13 points to a save-area for use by the randomizer with one more save-area chained below it.

The randomizer interface module is linked in 31-bit mode (AMODE=31). If the calling program runs in 24-bit mode addressing, you must complete an additional step to relink the randomizing module. Fast Path/EP provides sample link-edit JCL to relink the module to run in AMODE=24. This JCL is stored in the Fast Path/EP sample library in the member named #LKEDCAL.

---

**Randomizer calculations**

The IMS interface defined for DEDB randomizing routines specifies that, upon return from the randomizer:

- Register 1 will contain the address of the DMAC for the area selected.
- Register 0 will contain the relative RAP number within that area to which the root segment is assigned.

The PFUT0B50 randomizer interface module uses the RAP number and information from the DMAC to calculate the RAP RBA value, which is returned to a user-written program.

---

**NOTE**

The information in this section is for informational purposes only. You do not need to manually calculate the RAP RBA value because this value is calculated in the PFUT0B50 randomizer interface module.
The formula to convert the relative RAP number to the RBA of that RAP is shown in Figure 44:

\[
\text{RAP-RBA} = \text{CI-size} \times \left\{ \text{INT}(\frac{\text{RAP-No}}{\text{(UOW1 - UOW2)}}) + \text{REM}(\frac{\text{RAP-No}}{\text{(UOW1 - UOW2)}}) + 2 \right\}
\]

Table 100 describes the elements used in the conversion formula.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAP-No</td>
<td>relative RAP number. (from randomizer)</td>
</tr>
<tr>
<td>UOW1</td>
<td>UOW part 1</td>
</tr>
<tr>
<td>UOW2</td>
<td>UOW part 2</td>
</tr>
<tr>
<td>INT(x/y)</td>
<td>the integer value of the quotient from the division of x by y</td>
</tr>
<tr>
<td>REM(x/y)</td>
<td>the remainder from the division of x by y</td>
</tr>
</tbody>
</table>

Randomizer interface subroutine program example

The sample COBOL program shown in Figure 45 uses the PFUT0B50 subroutine to calculate the RAP RBA, and determine the Area No. and RAP RBA values for a root key. This subroutine can also be used to add physical sequence data to transactions that are to be applied in batch (BMP) mode.

Figure 45  Randomizer interface subroutine sample program (part 1 of 3)

```
IDENTIFICATION DIVISION.
PROGRAM-ID. SAMPOB50.
AUTHOR. BMC.

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

INPUT-OUTPUT SECTION.
FILE-CONTROL.
SELECT DATAIN ASSIGN TO UT-S-DATAIN.

DATA DIVISION.
```
FILE SECTION.

FD DATAIN
   BLOCK CONTAINS 0 RECORDS
   RECORD CONTAINS 60 CHARACTERS
   RECORDING MODE IS F
   LABEL RECORDS STANDARD.
01 DATA-IN.
   05 SEG-NAME PIC X(8).
   05 IN-KEY PIC X(10).
   05 FILLER PIC X(42).

WORKING- storage SECTION.

77 FUNC-INIT PIC X(4) VALUE 'INIT'.
77 DBDNAME PIC X(8) VALUE SPACES.
77 FUNC-CALC PIC X(4) VALUE 'CALC'.
77 AREANO PIC S9(4) COMP.
77 RAPRBA PIC S9(8) COMP.
77 DBDCALC PIC X(8) VALUE SPACES.
77 IN-REC-EOF PIC X VALUE SPACES.
01 SEG-RECORD.
   05 SEG-ROOTKEY PIC X(10) VALUE SPACES.
01 WS-CONTROL-CARD.
   05 WS-DBDNAME PIC X(8) VALUE SPACES.
   05 FILLER PIC X(72) VALUE SPACES.

PROCEDURE DIVISION.

A100.

OPEN INPUT DATAIN.

ACCEPT WS-CONTROL-CARD FROM SYSIN.

MOVE WS-DBDNAME TO DBDCALC.

CALL 'PFUTOB50' USING FUNC-INIT.

PERFORM B100 UNTIL IN-REC-EOF = 'Y'.

CLOSE DATAIN.

GOBACK.

B100.

READ DATAIN AT END MOVE 'Y' TO IN-REC-EOF.

IF SEG-NAME = 'AZS3801'
   PERFORM C100.
The Fast Path Analyzer/EP SDEP Space Utilization Utility provides the following functions:

- extracting SDEP history data
- updating SDEP history files
- reformatting SDEP utilization records
- generating SDEP space utilization reports

Utility program modules

The SDEP Space Utilization Utility consists of program modules and an intermediate sort step:

- PFSD0700 – extract and journal SDEP space utilization data
- PFSD0800 – reformat data
- SORTXSDP – sort data
- PFSD0900 – update SDEP history file

Extract and journal SDEP space utilization data (PFSD0700)

Module PFSD0700 extracts and journals SDEP space utilization data. The extracted data is written to an OS file by a routine that executes as an exit routine which is invoked by the online DEDB Sequential Dependent Scan utility (DBFUMSC0). It accesses and analyzes the in-core DMAC and writes a data record to a file that is allocated with DISP=MOD defined by the SCANCOPY DD statement.
During each invocation of the scan utility, space utilization data can be extracted from all (or specified) areas of a DEDB.

Most application systems that use sequential dependents run the SCAN utility to copy the sequential dependent segments to a sequential data set, followed by the Delete Utility to logically delete the segments.

PFSD0700 is intended to be run just before the delete utility, so that the high-water space utilization mark can be captured. However, this utility can be run as often as desired; the space utilization graph will always reflect the highest amount of space that is used.

**Reformat SDEP utilization data records (PFSD0800)**

Module PFSD0800 reformats the SDEP utilization data records and resets the end-of-file marker on the journal data set.

**Generate SDEP space utilization reports (PFSD0900)**

Module PFSD0900 updates the SDEP history file using sorted records and generates space utilization reports for all or specified areas of any number of databases. The report will reflect the previous 31 calendar days beginning with the current day.

**Creating and initializing the SDEP history VSAM data set**

The SDEP history file is stored in a VSAM KSDS data set. You must create and initialize this data set prior to running program PFSD0900 the first time. After this data set is initialized, it will be used each time you execute the SDEP Space Utilization Utility to accumulate SDEP information and build a historical record of SDEP utilization.

Use the JCL shown in Figure 46 to create the SDEP history VSAM data set. This JCL is stored in the Fast Path/EP sample library in the member named PFUSDEP0.

**Figure 46  Create SDEP history VSAM data set JCL (part 1 of 2)**

```plaintext
//PFUSDEP0 JOB (ACCT),"PFP",MSGCLASS=A,CLASS=A <= CHANGE
//********************************************
//**  SAMPLE JCL: ALLOCATE SDEP HISTORY FILE  **
//********************************************
//** ALLOCATE KSDS AND REPRO IN A RECORD TO INITIALIZE  **
//** USING MEMBER PFUSDEPX TO INITIALIZE THE VSAM DSN  **
//********************************************
//S01B    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*```
The VSAM data set is initialized with the two dummy records that are contained in the Fast Path/EP sample library member named PFUSDEPX. The dummy records consist of the following:

- 16 bytes of low values - X'00', followed by a literal of BMC Software Inc.
- 16 bytes of high values - X'FF', followed by a literal of BMC Software Inc.

**Process flow**

Figure 47 shows the process flow of the three modules and interim sort steps for the SDEP Space Utilization Utility.

Module PFSD0700 is usually executed as a stand-alone process. This is because PFSD0700 executes as an exit routine in an IFP region under the IBM Fast Path SDEP Scan Utility DBF#FPUO.

Modules PFSD0800 and PFSD0900 are usually executed in the same procedure to reformat the data and update the SDEP history records using the data records that were journalled by executing module PFSD0700. This procedure can be executed as often as required to update the SDEP history file.
**Figure 47**  SDEP update and reporting process flow

```
DDEB or area

PFSD0700
Extract and Journal

SDEP Journal

PFSD0800
Reformat

SDEP Data

Sort SORTXSDP

Sorted SDEP

PFSD0900
Update and Report

Control Card

SDEP History

Reports

Messages

Messages
```
PFSD0700 JCL requirements

Use the standard online SDEP scan utility JCL as described in the IMS Utilities Reference Manual. The JCL shown in Figure 48 is contained in the Fast Path/EP sample library in the member named PFUSDEP1.

PFSD0700 control statements

Use standard scan utility commands. BMC Software recommends that only the following command keywords be used:

- TYPE
- ERROR
- NOSORT
- AREA
- EXIT
- BUFNO
- GO

In Figure 48, SDEP space utilization data is extracted from area PFPDBM01 of database PFPDBM0.

Figure 48  Data extracted from PFPDBM01

```plaintext
//PFSD0700 EXEC FPUTIL,DBD=PFPDBM0,
  //              REST=00,REGION=0M,IMSID=????
//FPU.STEPLIB DD
//   DD DSN=BMC.PFP.LOAD,DISP=SHR
//SCANCOPY DD DSN=BMC.PFP.SDEP,
//   DISP=(MOD,KEEP,KEEP),
//   UNIT=SYSDA,SPACE=(TRK,(5,2),RLSE)
//SYSIN    DD *
  TYPE   SCAN
  ERROR  SCAN
  NOSORT
  EXIT   PFSD0700
  AREA   PFPDBM01
  GO
  //*
```
PFSD0800 JCL requirements

PFSD0800 is executed as a standard MVS job step. An EXEC statement and DD statements that define inputs and outputs are required.

EXEC

This statement must be in the following format:

```//PFSD0800 EXEC PGM=PFSD0800,REGION=0M```

STEPLIB DD

Defines the load library in which the Fast Path Analyzer/EP programs reside.

SYSPRINT DD

Defines the output message data set.

DADARI DD

Defines the input data set containing the SDEP space utilization data records created by the SCANCOPY DD statement in the PFSD0700 module. This DD statement should always specify DISP=OLD.

DADARO DD

Defines the output data set containing the reformatted SDEP space utilization data records. BLKSIZE must be a multiple of 72. This data set is input to the SORTXSDP sort step.

SORTXSDP

The SORTXSDP step sorts the data set, which is required prior to its input to module PFDS0900.

The following DD statements are required to define input and output:

SORTIN DD

Output from SCANCOPY DD in PFSD0700.
PFSD0900 JCL requirements

PFSD0900 is executed as a standard MVS job step. An EXEC statement and DD statements that define inputs and outputs are required.

EXEC

This statement must be in the following format:

```
//PFSD0900 EXEC PGM=PFSD0900,REGION=0M,COND=(4,LT),
PARM=(THR=35)
```

THR=nn on the EXEC statement specifies the threshold space utilization percentage value. Valid values are 00 to 99. (If omitted, the default is 85.)

If the space utilization exceeds the threshold percentage value, a special return code is set (99) and a report is generated even when update only is specified.

STEPLIB DD

Defines the load library in which the Fast Path Analyzer/EP programs reside.

MSGOUT DD

Defines the output message data set.

DADARI DD

Defines the input data set containing the SDEP space utilization data records.

DASDHIO DD

Defines the SDEP history file (VSAM KSDS).
**RPTOUT DD**

Defines the output SDEP utilization report data set. Use RECFM=FBA and LRECL=121.

**SYSIN DD**

Defines the input control statement data set. The sample SDEP update and report generation JCL updates the history file with all SDEP space utilization data journaled to date, and generates SDEP Utilization reports for all areas of the specified database.

The control statement must be in the following format, as shown in the following example:

```
//SYSIN DD *
  DBDNAME=dbdname, AREA=areaname
/*
```

**PFSD0900 control statements**

A control statement is used to specify the name of the database, and optionally the area, for which a utilization graph is to be generated.

**DBDNAME**

Required. Specifies the name of the database for which SDEP space utilization reports are to be generated.

**AREA**

Optional. Specifies the DDNAME of the specific area for which an SDEP space utilization report is to be generated. If an AREA keyword is not present, reports will be produced for all areas of the database.

**PFSD0900 return codes**

This program writes numbered return code messages as shown in Table 101 to the SYSPRINT data set. These messages more fully explain the results of program execution.
Table 101  PFSD0900 return codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The requested operation completed successfully.</td>
</tr>
<tr>
<td>4</td>
<td>Warning messages were issued, but the requested operation completed.</td>
</tr>
<tr>
<td>99</td>
<td>Threshold values were exceeded and warning messages were generated. The</td>
</tr>
<tr>
<td></td>
<td>requested operation completed.</td>
</tr>
</tbody>
</table>

PFDS0800 and PFDS0900 sample JCL

Figure 49 shows sample JCL that executes the formatting and updating modules for the SDEP Space Utilization Utility. This JCL is contained in the Fast Path/EP sample library in the member named PFUSDEP2.

Figure 49  Sample SDEP update and report generation JCL (part 1 of 2)

```plaintext
//*******************************************
//** SAMPLE JCL: SDEP UPDATE AND REPORTING **
//*******************************************
//** REFORMAT SDEP DATA **
//*************************
//PFSD0800 EXEC PGM=PFSD0800,REGION=0M
//STEPLIB DD DSN=BMC.PFP.LOAD,DISP=SHR
//SYSPRINT DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//DADARI DD DSN=BMC.PFP.SDEP,DISP=OLD 
//DADARO DD DSN=BMC.PFP.XSDEP,
//         DISP=(NEW,CATLG,DELETE),
//         UNIT=SYSDA,
//         SPACE=(TRK,(5,2),RLSE),
//         DCB=BLKSIZE=5976
//****************************
//** SORT SDEP DATA RECORDS **
//*****************************
//SORTXSDP EXEC PGM=SORT,COND=(4,LT)
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=* 
//SYSIN DD *,DCB=BLKSIZE=80 
SORT FIELDS=(1,24,CH,A) 
//SORTIN DD DSN=BMC.PFP.XSDEP,
//         DISP=(OLD,DELETE,KEEP) 
//SORTOUT DD DSN=BMC.PFP.SORTED.XSDEP,
//         DISP=(NEW,CATLG,DELETE),
//         UNIT=SYSDA,
//         SPACE=(TRK,(5,2),RLSE) 
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(1,1)) 
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(1,1)) 
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(1,1)) 
```
Generating SDEP utilization reports only

By executing a subset of the SDEP update and report generation JCL, you can generate SDEP Utilization reports without executing the update phase. Complete the following steps.

1. Omit the DADARI DD statement in PFSD0900 or change the DADARI DD statement as follows:

   ```
   //DADARI DD DUMMY,DCB=BLKSIZE=72
   ```

2. Add the appropriate control statements.

3. Execute module PFSD0900 only.

In Figure 50, SDEP utilization reports are generated for all areas of PFPDBM01.

---

**Figure 49  Sample SDEP update and report generation JCL (part 2 of 2)**

```plaintext
//*******************************************************
//** UPDATE SDEP HISTORY VSAM DSN AND GENERATE REPORTS **
//*******************************************************
//PFSD0900 EXEC PGM=PFSD0900,REGION=0M,COND=(4,LT)
//STEPLIB DD DSN=BMC.PFP.LOAD,DISP=SHR
//MSGOUT DD SYSOUT=* 
//RPTOUT DD SYSOUT=* 
//DADARI DD DSN=BMC.PFP.SORTED.XSDEP, 
//       DISP=(OLD,DELETE,KEEP)
//DASDHIO DD DSN=BMC.SDEPHIST,DISP=OLD 
//SYSIN DD *,DCB=BLKSIZE=80
//DADNAME=PFPDBM01
```
By executing a subset of the SDEP update and report generation JCL, you can update the SDEP History File without generating reports.

1. Omit the SYSIN DD statement in PFSD0900 or change the SYSIN DD statement to:

```
//SYSIN DD DUMMY,DCB=BLKSIZE=80
```

2. Add the appropriate control statements.

3. Execute modules PFDS0800 and PFSD0900.

**WARNING**

If you follow this procedure to update the SDEP History File without generating reports, the utility will still generate reports automatically if the default (or specified) SDEP space utilization threshold percentage is exceeded.
Updating the SDEP History File only
Fast Path/EP Series offers program extensions FABEUR6, FABEUR7 and FABGXDR. These enable you to customize the functionality of the Fast Path/EP Series products. For each of these program extensions, this chapter discusses their function(s), JCL requirements, and how to code control statements.

FABEUR6 Reload File Create utility ................................................. 502
  FABEUR6 overview ............................................................... 502
  FABEUR6 parameter lists ...................................................... 503
  FABEUR6 JCL requirements ................................................... 507
  FABEUR6 utility control statements ....................................... 509
FABEUR7 Read Unloaded Database utility ...................................... 514
  FABEUR7 overview ............................................................... 514
  FABEUR7 parameter lists ...................................................... 515
  FABEUR7 JCL requirements ................................................... 521
  FABEUR7 utility control statement for HD Unload input ............... 522
  FABEUR7 utility control statements for non-HD Unload input ....... 523
DL/I segment edit/compression routine ........................................ 526
  Convention of registers on entry .......................................... 526
  Entry code convention ....................................................... 527
FABGXDR DEDB data extractor I/O module .................................... 527
  JCL requirements for FABGXDR ............................................. 527
  Parameter lists for FABGXDR ................................................. 528
  Initialization function ....................................................... 529
  Get data function ............................................................. 529
  End-of-file function ......................................................... 530
  Sample COBOL program using FABGXDR .................................. 531
Callable module considerations .................................................. 533
FABEUR6 Reload File Create utility

This section describes the utility that formats and writes an unload file of segment records. It also provides the JCL requirements and its CALL interfaces.

FABEUR6 overview

The Reload File Create utility (FABEUR6) can be used to format and write the segment data records in several different formats that can be input to the reload processor in Fast Path Reorg/EP. This action significantly simplifies the generation of unload file records (produced by an application program) by eliminating the requirement to:

- invoke the randomizer routine for each root segment
- format the segment data record prefix
- write the segment records to an output file

FABEUR6 can be used also to:

- add segments to, or delete segments from, an unloaded database
- add fields to a segment between the unload and reload of a database

FABEUR6 utility has been designed and written to allow users to migrate to new Fast Path/EP releases without modifying or re-link-editing user programs.

If FABEUR6 is link-edited into a user program, you should complete the following tasks:

- modify the STEPLIB DD data sets to reference the data set containing the new Fast Path Reorg/EP release modules
- remove any references to the OLD release data set
- execute the program

When these tasks are completed, the new version of FABEUR6X (the I/O module for FABEUR6) will be invoked. This allows changes to the Reload File Create process without impacting the user’s application program.
FABEUR6X can also be used to create a complete set of TRIMAR reload files. If you include the UR6DBDFN DD and DURSzzzO DD statements in the execution JCL, FABEUR6X will write the unload file to the DURDzzzO DD in TRIMAR format and create the DURDBDFN and DURSzzzO files.

If FABEUR6 is *not* link-edited into a user program, but is invoked via an ATTACH, LINK, or by DYNAMIC calls, see “Callable module considerations” on page 533.

---

**FABEUR6 parameter lists**

FABEUR6 is called with a parameter list consisting of a function code and one or more other data items. The function code is used to indicate which function to perform.

**FABEUR6 initialization function**

The FABEUR6 initialization function performs the following:

- sets up the required environment for subsequent randomizer calls
- edits and parses the control statements
- opens all required output files

---

**NOTE**

INIT function must be performed prior to any other function call.

---

For COBOL, define in WORKING-STORAGE the following:

```
77 FUNCINIT PIC X(4) VALUE 'INIT'.
77 DBDNAME PIC X(8) VALUE 'dbdname'.
```

where `dbdname` is the DBD name.
In the PROCEDURE DIVISION, add the following:

```
CALL 'FABEUR6' USING FUNCINIT, DBDNAME.
```

**FABEUR6 build and write function**

If the segment being written is a root segment, the FABEUR6 build and write function invokes the randomizing routine. The segment data and the record prefix is then written to the appropriate output file.

**NOTE**

Segments within a database record must be presented to FABEUR6 in hierarchical order.

For COBOL, define the following in WORKING-STORAGE:

```
77 FUNCPUT PIC X(4) VALUE 'PUT '.
01 SEG-IOAREA.
   05 SEG-NAME PIC X(8).
   05 SEG-SSPTRS PIC X OCCURS 8 TIMES.
   05 SEG-DATA PIC X (nnn).

   05 ROOT-SEG REDEFINES SEG-DATA.
   10 ROOT-LENGTH PIC S9(4) COMP.
   10 ROOT-KEY  PIC X(??).
   .

   05 DIR1-SEG REDEFINES SEG-DATA.
   10 DIR1-LENGTH PIC S9(4) COMP.
   10 DIR1-KEY  PIC X(??).
   .
```

where:

- *nnn* is the length of the longest segment definition (excluding IMS prefix).

- SEG-SSPTRS is eight 1-byte positional flags set left to right for pointers 1-8 that indicate whether or not this segment is the target of a subset pointer. Adding a Y in one of the flag bytes will cause the appropriate subset pointer to be set.
In the PROCEDURE DIVISION, code the following:

```
CALL 'FABEUR6' USING FUNCPUT, SEG-IOAREA.
```

**FABEUR6 end-of-file function**

The FABEUR6 end-of-file function is invoked after all segments for a database have been processed. All output files are closed.

For COBOL, define the following in WORKING-STORAGE:

```
77 FUNCEOF PIC X(4) VALUE 'EOF'.
```

In the PROCEDURE DIVISION, code the following:

```
CALL 'FABEUR6' USING FUNCEOF.
```

**FABEUR6 example 1**

The example COBOL program shown in Figure 51 uses the FABEUR6 utility to format and write the segment data records in the format required by Fast Path Reorg/EP. This sample program is included in the Fast Path/EP sample library member name PFUEUR6 on the product distribution tape.

**Figure 51   Reload File Create user exit sample (part 1 of 3)**

```
IDENTIFICATION DIVISION.
PROGRAM-ID. SAMPUR6.
AUTHOR. BMC.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
   SELECT DATALN ASSIGN TO UT-S-DATAIN.

DATA DIVISION.
FILE SECTION.

FD DATAIN
   BLOCK CONTAINS 0 RECORDS
   RECORD CONTAINS 60 CHARACTERS
   RECORDING MODE IS F
   LABEL RECORDS STANDARD.

01 DATA-IN.
   05 SEG-NAME PIC X(8).
```
**Figure 51  Reload File Create user exit sample (part 2 of 3)**

```
05 IN-KEY            PIC X(52).
WORKING-STORAGE SECTION.
77 FUNC-INIT            PIC X(4)    VALUE 'INIT'.
77 FUNC-PUT            PIC X(4)    VALUE 'PUT'.
77 FUNC-EOF            PIC X(4)    VALUE 'EOF'.
77 DATAIN-EOF            PIC X    VALUE 'N'.
77 DBDNAME            PIC X(8)    VALUE SPACES.
01 WS-CONTROL-CARD.
  05 WS-DBDNAME         PIC X(8)    VALUE SPACES.
  05 FILLER             PIC X(72)   VALUE SPACES.
01 WS-IOAREA.
  05 IO-SEGNAME         PIC X(8)    VALUE SPACES.
  05 IO-SSPTRS          PIC X(8)    VALUE SPACES.
  05 IO-SEGDATA.
    10 IO-SEG-LENGTH  PIC S9(4)   COMP VALUE +12.
    10 IO-SEG-DATA    PIC X(50)   VALUE SPACES.
PROCEDURE DIVISION.

A100.

    PERFORM B100.
    PERFORM C100 UNTIL DATAIN-EOF = 'Y'.
    PERFORM E100.
    GOBACK.

B100.

    OPEN INPUT DATAIN.
    ACCEPT WS-CONTROL-CARD FROM SYSIN.
    MOVE WS-DBDNAME TO DBDNAME.
    CALL 'FABEUR6' USING FUNC-INIT DBDNAME.

C100.

    READ DATAIN AT END MOVE 'Y' TO DATAIN-EOF.
    IF SEG-NAME = 'AZS3801'
      MOVE +12 TO IO-SEG-LENGTH
    ELSE
      MOVE +52 TO IO-SEG-LENGTH.
    PERFORM D100.

D100.

    MOVE SEG-NAME TO IO-SEGNAME.
    MOVE IN-KEY TO IO-SEG-DATA.
    CALL 'FABEUR6' USING FUNC-PUT WS-IOAREA.

E100.
```
FABEUR6 JCL requirements

FABEUR6 is a called module. A diagram of a user program and the subroutine is shown in Figure 52.

This module requires the following DD statements to be included in the job step:

**ACBLIB DD**

Defines the library containing the DMB for the database. This DD statement is required unless the MODSTAT DD, MODSTAT2 DD, or OLCSTAT DD statement is provided.

**MODSTAT / MODSTAT2**

Optional. Used to identify the active online IMS ACB data set (IMSACBA or IMSACBB). Dynamic allocation can be used for these DD statements.
When one of these DD statements is present, the MODSTAT data set is interrogated to determine whether IMSACBA or IMSACBB is the active library. If the IMSACBA or IMSACBB DD statement is not present in the JCL, the STEPLIB/LINKLIST are searched for a DFSMDA member.

If MODSTAT2 DD is present, the active MODSTAT data set is determined prior to ACBLIB selection. The MODSTAT2 data set is interrogated to determine whether IMSACBA or IMSACBB is the active library. If the IMSACBA or IMSACBB DD statement is not present in the JCL, the STEPLIB/LINKLIST are searched for a DFSMDA member.

If both OLCSTAT and MODSTAT DD statements are present, then OLCSTAT will be used and MODSTAT will be ignored.

**OLCSTAT DD**

Optional. Used to identify the active online IMS ACB data set (IMSACBA or IMSACBB) in an IMS global online change environment. Dynamic allocation can be used for this DD statement.

When this DD statement is present, the OLCSTAT data set is interrogated to determine whether IMSACBA or IMSACBB is the active library. If the IMSACBA or IMSACBB DD statement is not present in the JCL, the STEPLIB/LNKLIST are searched for a DFSMDA member.

If both OLCSTAT and MODSTAT DD statements are present, then OLCSTAT will be used and MODSTAT will be ignored.

**STEPLIB DD**

Required. The following data sets must be concatenated to this DD statement:

- IMS RESLIB
- the program load library containing the FABEUR6 calls
- the load library containing the randomizer module used for the database

**UR6DBDFN DD**

Optional. Defines an output data set for the database definition record (the formatted DMB) used by TRIMAR-compatible processing. The output file is blocked to the maximum size for the output device (unless overridden in the execution JCL). Because the blocking factor is determined at execution time, standard labels must be used. If you specify the UR6DBDFN DD statement, you must also specify the DURSzzzO DD statement to identify the sort control card output.
UR6RPT DD

Required. Defines the messages and audit report output data set.

- RECFM=FBA
- LRECL=121

UR6FCTL DD

Optional. Defines the file control statement input data set. This data set can reside on a direct access device, or be routed through the input stream.

DURDzzzO DD

Required. Defines an output data set for all database segment records for one or more areas with the relative area numbers between 001 and 999, as defined in the DMB. The rules for supplying DURDzzzO data sets are discussed with the FILECTL statement.

XDyyyyyO DD

Required. Defines an output data set for all database segment records for one or more areas with the relative area numbers between 00001 and 02048, as defined in the DMB. The rules for supplying XDyyyyyO data sets are discussed with the FILECTL statement.

DURSzzzO DD

Optional. Defines an output data set for sort control statements used for TRIMAR compatible processing only. This DD is associated with the segment output data set for relative area numbers 001 through 999. The output data set that is specified on the DURSzzzO DD statement will be written only if you include the UR6DBDFN DD statement in the JCL.

FABEUR6 utility control statements

The following control statements are available for FABEUR6:

- FILECTL
- DLICOMP
- FORMAT

All statements must begin in column 1. Sample FABEUR6 control statements are shown in Figure 53.
FILECTL statement

FILECTL controls the optional grouping of multiple areas segment data into a single output file:

- There must be a DURDzzzO or XDyyyyyO DD statement in the JCL stream for each file specified on a FILECTL control statement (where zzz is the three-digit file number from 001 through 999, or yyyyy is the five-digit file number from 00001 through 02048 specified on the control statement).

- A value for area number of * indicates that the segment data records for all areas defined in the output DMB that have not yet been specified in a FILECTL statement are to be written to this file.

- If FILECTL statements are not specified, FABEUR6 expects to write the segment data records for a given area defined in the output DMB to the file associated with a DURDzzzO DD or XDyyyyyO DD statement, where zzz is the three-digit area number from 000 through 999 or yyyyy is the five-digit area number from 00001 through 02048 that is assigned to that area during ACBGEN processing. Consequently, an output file is required for each area defined in the output DMB.

- Duplicate references to an area or File in the FILECTL specifications will be flagged with an error message and will cause program termination.

DLICOMP statement

DLICOMP is an optional control statement that denotes if the DL/I segment edit/compression routine will be invoked with a COMPRESS (entry code 0) call, for candidate segments. The default is NO.

FORMAT statement

FORMAT is an optional control statement that specifies the format of the unload output file. The following values are available:

- HDUNLOAD – (the default) specifies the file to be written in a format compatible with the IBM HD Reorganization Unload Utility.
TFMT – specifies the file to be written in a format compatible with the TRIMAR FAST PATH UNLOAD/RELOAD product. If you want to generate a complete set of TRIMAR-compatible reload files, you should include the UR6DBDFN DD statement and the DURSzzzO DD statement in the JCL.

DBT – specifies the file to be written in a format compatible with the DEDB Reload Utility component of the IBM Fast Path Basic Tools for OS/390.

See Appendix C, “DEDB unload/reload record layouts” for the record layout when TFMT or DBT is specified as the unload format.

**Figure 53  Sample FABEUR6 utility control statements**

| FILECTL=  | nn,ALL  | nn : output file no. |
|          | nn,x    | x : single area       |
|          | nn,(x,x..x) | x,x..x : selected areas |
|          | nn,(x-x) | x-x : group of areas  |
|          | nn,(*)  | * : all “not yet assigned” areas |
| DLICOMP= | NO, CMP |
| FORMAT=  | HDUNLOAD, TFMT, DBT |

**NOTE**

Refer to the Fast Path Offline Suite User Guide for instructions on reloading the unload file into a DEDB.

**FABEUR6 example 2**

In the example shown in Figure 54, the COBOL program reads a variable length record consisting of a 200-byte fixed portion, the last two bytes of which indicate the number of occurrences of a 150-byte trailer portion.

The database to be loaded consists of a 200-byte root segment (ROOTSEG) and 152-byte (length + data) direct dependent segments (TRAILSEG).

**Figure 54  COBOL Example (part 1 of 4)**

<table>
<thead>
<tr>
<th>IDENTIFICATION DIVISION.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGRAM-ID. TESTUR6.</td>
</tr>
<tr>
<td>AUTHOR. BMC SOFTWARE.</td>
</tr>
<tr>
<td>INSTALLATION. BMC SOFTWARE.</td>
</tr>
<tr>
<td>DATE-COMPILED.</td>
</tr>
<tr>
<td>REMARKS.</td>
</tr>
<tr>
<td>CREATE FILE FOR DATABASE LOAD.</td>
</tr>
</tbody>
</table>
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-370.
OBJECT-COMPUTER. IBM-370.

INPUT-OUTPUT SECTION.
FILE-CONTROL.
   SELECT DATAIN ASSIGN TO UT-S-DATAIN.

DATA DIVISION.
FILE SECTION.
FD DATAIN
   BLOCK CONTAINS 0 RECORDS
   RECORD CONTAINS 0 CHARACTERS
   RECORDING MODE IS V
   LABEL RECORDS STANDARD.

01 DATA-REC.
   05 FIXED-PART.
      10 FIXED-DATA          PIC X(198).
      10 NO-TRAILERS         PIC S9(4)   COMP.
   05 TRAILER             OCCURS 0 TO 20 TIMES
                        DEPENDING ON NO-TRAILERS
                        PIC X(150).

WORKING-STORAGE SECTION.
77 FUNC-INIT               PIC X(4)    VALUE 'INIT'.
77 FUNC-PUT                PIC X(4)    VALUE 'PUT'.
77 FUNC-EOF                PIC X(4)    VALUE 'EOF'.
77 DATAIN-EOF              PIC X       VALUE 'N'.
77 REC-CTR                 PIC S9(4)   COMP  VALUE +0.
77 SUB1                    PIC S9(4)   COMP.

01 WS-CONTROL-CARD.
   05 WS-DBDNAME          PIC X(8).
   05 FILLER              PIC X(72).

01 WS-IOAREA.
   05 IO-SEGNAME          PIC X(8).
   05 IO-SSPTRS           PIC X(8)    VALUE SPACES.
   05 IO-SEGDATA.
      10 IO-SEG-LENGTH       PIC S9(4)   COMP.
      10 IO-SEG-DATA         PIC X(300).

PROCEDURE DIVISION.
A100-MAINLINE.
Figure 54  COBOL Example (part 3 of 4)

PERFORM Y100-INIT THRU Y100-INIT-X.

PERFORM B100-PROCESS THRU B100-PROCESS-X
UNTIL DATAIN-EOF = 'Y'.

PERFORM Z100-EOJ THRU Z100-EOJ-X.

GOBACK.

B100-PROCESS.

READ DATAIN
  AT END MOVE 'Y' TO DATAIN-EOF
  GO TO B100-PROCESS-X.

ADD +1 TO REC-CTR.
MOVE 'ROOTSEG' TO IO-SEGNAME.
MOVE +200 TO IO-SEG-LENGTH.
MOVE FIXED-DATA TO IO-SEG-DATA.
CALL 'FABEUR6' USING FUNC-PUT
           WS-IOAREA.

MOVE +1 TO SUB1.
MOVE 'TRAILSEG' TO IO-SEGNAME.
MOVE +152 TO IO-SEG-LENGTH.
PERFORM B120-BUILD-TRAILER THRU B120-BUILD-TRAILER-X
UNTIL SUB1 > NO-TRAILERS.

B100-PROCESS-X.
  EXIT.

B120-BUILD-TRAILER.

MOVE TRAILER (SUB1) TO IO-SEG-DATA

CALL 'FABEUR6' USING FUNC-PUT.
           WS-IOAREA.

ADD +1 TO SUB1.

B120-BUILD-TRAILER-X.
  EXIT.

Y100-INIT.

OPEN INPUT DATAIN.

ACCEPT WS-CONTROL-CARD FROM SYSIN.

CALL 'FABEUR6' USING FUNC-INIT
FABEUR7 Read Unloaded Database utility

This section describes the subroutine that extracts segment data from an unloaded database. It also provides JCL requirements and the CALL interfaces. BMC recommends that you use this subroutine to read the unload database file rather than reading it directly. Using this subroutine will ensure that changes in the file format will not impact the user program.

For specific application processing needs, you can use the Fast Path/EP File Sort Utility to sort the unload file prior to application program execution. This command-driven utility provides a method for invoking your installation’s sort utility to perform a customized sort of the unload input file. In addition to providing other sorting alternatives, the File Sort utility provides a method of pre-sorting a file in RAP or load sequence.

For detailed information on using the File Sort Utility, see Chapter 5, “Supporting utilities.”

FABEUR7 overview

The Read Unloaded Database (FABEUR7) utility provides an easy method for retrieving database segments from an unloaded database file created by Fast Path Reorg/EP. Segments within a database record are returned to the caller in hierarchical order.
FABEUR7 has the following features:

- retrieves database records for off-line report generation
- reads and presents an application view of the segment data
- permits users to read two unloaded area data sets from the same DEDB through a dual processing feature
- accepts HD Unload, TRIMAR unload, or IBM Fast Path unload files as input

This utility has been designed and written to allow users to migrate to new releases without modifying or re-link-editing user programs.

If FABEUR7 is link-edited into a user program, modify the STEPLIB DD data sets to reference the data set containing the new release modules, remove any references to the old release data set, and execute the program. The new version of FABEUR7X (the I/O module) will be invoked.

As a conversion option, the Fast Path/EP version of FABEUR7X can also read an unload file in TRIMAR or IBM DBT format. If you include the TRIMAR or DBD-format unload file on the UR7DATA DD statement and add the DURDBDFN file to the execution JCL on a UR7DBDFN DD statement, FABEUR7X will process the unload file.

If FABEUR7 is not link-edited into a user program, but invoked via an ATTACH, LINK, or by DYNAMIC CALLS, see “Callable module considerations” on page 533.

**FABEUR7 parameter lists**

FABEUR7 is called with a parameter list consisting of a function code and one or more other data items. The function code is used to indicate which function to perform.

**FABEUR7 initialization function**

The initialization function INIT opens the input data sets. It parses any control cards.
For COBOL, define the following in WORKING-STORAGE:

```cobol
77 FUNCINIT  PIC X(4)     VALUE 'INIT'.
```

In the PROCEDURE DIVISION, code the following:

```cobol
CALL 'FABEUR7' USING FUNCINIT.
```

**FABEUR7 dual mode initialization**

For COBOL, define the following in WORKING-STORAGE:

```cobol
77 FUNCINIT  PIC X(4)     VALUE 'INID'.
```

In the PROCEDURE DIVISION, code the following:

```cobol
CALL 'FABEUR7' USING FUNCINIT.
```

**FABEUR7 get segment data function**

The GET DATA function reads the next sequential record. The segment name and data are returned to the caller.

For COBOL, define the following in WORKING-STORAGE:

```cobol
77 FUNCGET   PIC X(4)     VALUE 'GET '.
77 STATUS   PIC XX       VALUE SPACES.
01 IO-AREA.
  05 IO-SEGNAME  PIC X(8).
  05 IO-SSPTRS  PIC X      OCCURS 8 TIMES.
  05 IO-SEGDATA  PIC X(nnn).  
```

where

- *nnn* is the length of the longest segment.
- STATUS is set to GB at end-of-file.
- IO-SSPTRS is eight 1-byte positional flags set left-to-right for pointers 1 - 8 that indicate whether or not this segment is the target of a subset pointer. Adding a Y in one of the flag bytes indicates that this segment is pointed to by the corresponding subset pointer.
In the PROCEDURE DIVISION, code the following:

```
CALL 'FABEUR7' USING FUNCGET,
       STATUS,
       IO-AREA.
```

**FABEUR7 get segment (dual processing mode)**

For COBOL, define in WORKING-STORAGE the following:

```
77   FUNCGET1          PIC X(4)     VALUE 'GET1'.
77   FUNCGET2          PIC X(4)     VALUE 'GET2'.
77   STATUS            PIC XX       VALUE SPACES.
01   IO-AREA.
   05  IO-SEGNAME    PIC X(8).
   05  IO-SSPTRS     PIC X        OCCURS 8 TIMES.
   05  IO-SEGDATA    PIC X(nnn).
```

where

- `nnn` is the length of the longest segment.
- `STATUS` is set to GB at end-of-file.
- `IO-SSPTRS` is eight 1-byte positional flags set left-to-right for pointers 1-8 that indicate whether or not this segment is the target of a subset pointer. Adding a Y in one of the flag bytes indicates that this segment is pointed to by the corresponding subset pointer.

In the PROCEDURE DIVISION, code the following:

```
CALL 'FABEUR7' USING FUNCGET1,
       STATUS,
       IO-AREA.

** PROCESS ***

CALL 'FABEUR7' USING FUNCGET2,
       STATUS,
       IO-AREA.
```

**FABEUR7 end-of-file function**

The end-of-file function closes the files. The INIT function must be performed prior to an EOF call.
For COBOL, define the following in WORKING-STORAGE:

```
77 FUNCEOF PIC X(4) VALUE 'EOF'.
```

In the PROCEDURE DIVISION, code the following:

```
CALL 'FABEUR' USING FUNCEOF.
```

**FABEUR7 retrieving extended root/segment information**

By including one more I/O area on the GET or GET1/GET2 parameter lists, a user can retrieve more information about the returned segment. FABEUR7 will return an extended information block in the second I/O area with the following format:

<table>
<thead>
<tr>
<th>Fieldname</th>
<th>ASM Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAP RBA</td>
<td>F</td>
</tr>
<tr>
<td>Root Sequence</td>
<td>PL4</td>
</tr>
<tr>
<td>Reserved</td>
<td>F</td>
</tr>
<tr>
<td>Root Key Length</td>
<td>H</td>
</tr>
<tr>
<td>Root Key Data</td>
<td>XL256</td>
</tr>
<tr>
<td>Curr. Seg. Type</td>
<td>CL4</td>
</tr>
<tr>
<td>&quot; &quot; Seg-Code</td>
<td>H</td>
</tr>
<tr>
<td>&quot; &quot; Hierarchical Level</td>
<td>H</td>
</tr>
<tr>
<td>&quot; &quot; Key Length</td>
<td>H</td>
</tr>
<tr>
<td>&quot; &quot; Key Data</td>
<td>XL256</td>
</tr>
</tbody>
</table>

**FABEUR7 parameter lists**

For COBOL, define the following in WORKING-STORAGE:

```
01 WS-EXTENDED-IOAREA.
   05 RAP-RBA PIC S9(8) COMP.
   05 ROOT-SEQ PIC S9(7) COMP-3.
   05 FILLER PIC S9(8) COMP.
   05 ROOT-KEY-LEN PIC S9(4) COMP.
   05 ROOT-KEY-DATA PIC X(256).
   05 CURR-SEG-TYPE PIC X(4).
   05 CURR-SEG-CODE PIC S9(4) COMP.
   05 CURR-SEG-HEIR-LVL PIC S9(4) COMP.
   05 CURR-SEG-KEY-LEN PIC S9(4) COMP.
   05 CURR-SEG-KEY-DATA PIC X(256).
```
In the PROCEDURE DIVISION, code the following:

```cobol
CALL 'FABEUR7' USING FUNCGET,
     STATUS,
     IO-AREA,
     EXTENDED-IO-AREA.
```

---

**NOTE**

No special JCL is required to take advantage of the extended information block.

---

**FABEUR7 example 1**

The COBOL sample program shown in Figure 55 uses the FABEUR7 utility to provide a facility for retrieving database segments from an unloaded database file created by Fast Path Reorg/EP. This sample program is included in the Fast Path/EP sample library member name PFUEUR7 on the product distribution tape.

---

**Figure 55    Read unloaded database example user exit (part 1 of 2)**

```
IDENTIFICATION DIVISION.
PROGRAM-ID. SAMPUR7.
AUTHOR. BMC.
INSTALLATION. BMC SOFTWARE.
DATE-COMPILED.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
   SELECT DATAOUT ASSIGN TO UT-S-DATAOUT.
DATA DIVISION
FILE SECTION.
FD   DATAOUT
   BLOCK CONTAINS 0 RECORDS
   RECORD CONTAINS 3986 CHARACTERS
   RECORDING MODE IS F
   LABEL RECORDS STANDARD.

01   DATA-OUT.
   05  SEGNAME-OUT     PIC X(8).
   05  SEGLEN-OUT      PIC S9(4) COMP.
   05  SEGDATA-OUT     PIC X(3976).

WORKING-STORAGE SECTION.
77  FUNC-INIT         PIC X(4) VALUE 'INIT'.
77  FUNC-GET          PIC X(4) VALUE 'GET'.
77  FUNC-EOF          PIC X(4) VALUE 'EOF'.
```
Figure 55  Read unloaded database example user exit (part 2 of 2)

```plaintext
77   FUNC-STATUS        PIC X(2)    VALUE SPACES.
01   WS-IOAREA.
  05  IO-SEGNAM        PIC X(8).
  05  IO-SSPTRS        PIC X       OCCURS 8 TIMES.
  05  IO-SEGDATA       PIC X(3976).
  05  IO-DATA          REDEFINES IO-SEGDATA.
 10   ROOT-LL          PIC S9(4) COMP.
 10   SEGDATA.
 15   ROOT-KEY         PIC X(12).
 15   FILLER           PIC X(3962).

PROCEDURE DIVISION.
A100.
  OPEN OUTPUT DATAOUT.
  CALL 'FABEUR7' USING FUNC-INIT.
  PERFORM B100 THRU B100-EXIT UNTIL FUNC-STATUS = 'GB'.
  CLOSE DATAOUT.
  CALL 'FABEUR7' USING FUNC-EOF.
  GOBACK.
B100
  CALL 'FABEUR7' USING FUNC-GET
     FUNC-STATUS
     WS-IOAREA.
  IF FUNC-STATUS = 'GB'
     GO TO B100-EXIT.
  MOVE SPACES TO DATA-OUT.
  MOVE IO-SEGNAM TO SEGNAM-OUT.
  MOVE ROOT-LL TO SEGLEN-OUT.
  MOVE SEGDATA TO SEGDATA-OUT.
  WRITE DATA-OUT.
B100-EXIT.
  EXIT.
```
**FABEUR7 JCL requirements**

FABEUR7 is a called module. It requires the following DD statements to be included in the job step:

**ACBLIB DD**

Required if input data set is not in HD Unload format. Defines the library containing the DMB for the database. This DD statement is required unless the MODSTAT DD, MODSTAT2 DD, or OLCSTAT DD statement is provided.

**MODSTAT / MODSTAT2**

Optional if input data set is not in HD Unload format. Used to identify the active online IMS ACB data set (IMSACBA or IMSACBB). Dynamic allocation can be used for these DD statements.

When one of these DD statements is present, the MODSTAT data set is interrogated to determine whether IMSACBA or IMSACBB is the active library. If the IMSACBA or IMSACBB DD statement is not present in the JCL, the STEPLIB/LINKLIST are searched for a DFSMDA member.

If MODSTAT2 DD is present, the active MODSTAT data set is determined prior to ACBLIB selection. The MODSTAT2 data set is interrogated to determine whether IMSACBA or IMSACBB is the active library. If the IMSACBA or IMSACBB DD statement is not present in the JCL, the STEPLIB/LINKLIST are searched for a DFSMDA member.

If both OLCSTAT and MODSTAT DD statements are present, then OLCSTAT will be used and MODSTAT will be ignored.

**OLCSTAT DD**

Optional if input data set is not in HD Unload format. Used to identify the active online IMS ACB data set (IMSACBA or IMSACBB) in an IMS global online change environment. Dynamic allocation can be used for this DD statement.

When this DD statement is present, the OLCSTAT data set is interrogated to determine whether IMSACBA or IMSACBB is the active library. If the IMSACBA or IMSACBB DD statement is not present in the JCL, the STEPLIB/LNKLIST are searched for a DFSMDA member.

If both OLCSTAT and MODSTAT DD statements are present, then OLCSTAT will be used and MODSTAT will be ignored.
**STEPLIB DD**

Required. Defines the library containing the FABEUR7X load module. The IMS RESLIB must be concatenated to this DD statement.

**UR7CTL DD**

Required if the input data set is not in HD Unload format. Defines the control statement input data set. This data set can reside on a direct access device or be routed through the input stream.

**UR7DATA DD**

Required. Defines the DEDB unload file (the segment data records generated by Fast Path Reorg/EP).

**UR7DATA1 DD/ UR7DATA2 DD**

Required if you want to read two unload files from the same database (dual processing). Defines the pair of unload files that should be read.

**UR7RPT DD**

Required. Defines the audit report output data set.

- RECFM=FBA
- LRECL=121

---

**FABEUR7 utility control statement for HD Unload input**

When input to FABEUR7 is coming from an input data set in HD Unload format, the only valid UR7CTL control statement is DLICOMP as shown in Figure 56.

**DLICOMP**

DLICOMP is an optional statement that denotes if the DL/I segment edit/compression routine will be invoked with an EXPAND call (entry code 4), for candidate segments. (Default is NO.)
Figure 56  Sample FABEUR7 utility control statement for HD Unload input

![DLICOMP statement](image)

FABEUR7 example for HD Unload input

The COBOL example program shown in Figure 60 is extracting data for a trial balance report for branch 01234. The database structure consists of a root segment, and five direct dependent segment types. All data to be extracted is in the root segment. Input is an unloaded database in HD Unload format that was created by Fast Path Reorg/EP. Sample JCL is shown in Figure 57.

Figure 57  Sample JCL for FABEUR7 (with HD Unload input)

```
//EXTRACT   JOB ....etc
//*
//STEP1    EXEC PGM=userpgm
//UR7RPT   DD SYSOUT=A
//UR7DATA  DD DSN=TSS.UR.DURD001,DISP=SHR
//*
//DATAOUT  DD DSN=USER.EXTRACT.DATA,  
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(CYL,(5,2),RLSE)),
//          DCB=(RECFM=FB,LRECL=49,BLKSIZE=21952)
```

FABEUR7 utility control statements for non-HD Unload input

When FABEUR7 is accepting an input data set that is not in HD Unload format (i.e., the input data set was created using FORMAT=TFMT or FORMAT=DBT), valid UR7CTL control statements are shown in Figure 58.

DLICOMP statement

DLICOMP is an optional statement that denotes if the DL/I segment edit/compression routine will be invoked with an EXPAND call (entry code 4), for candidate segments. (Default is NO.)

DBDNAME statement

DBDNAME is a required control statement when the data in the unload input file is not in HD Unload format. This control statement identifies the DBD to be processed.
The COBOL example program shown in Figure 60 is extracting data for a trial balance report for branch 01234. The database structure consists of a root segment, and five direct dependent segment types. All data to be extracted is in the root segment. Input is an unloaded database in TFMT format that was created by the BMC Software TRIMAR FAST PATH UNLOAD/RELOAD product. Sample JCL is shown in Figure 59.

Figure 59  Sample JCL for FABEUR7 (non-HD Unload input)

```plaintext
//EXTRACT JOB ...
//*
//STEP1 EXEC PGM=userpgm
//ACBLIB DD DSN=IMSVS.ACBLIB,DISP=SHR
//UR7CTL DD *
//     DBDNAME=dbdname
//UR7RPT DD SYSOUT=A
//UR7DATA DD DSN=TSS.UR.DURD001,DISP=SHR
//*
//DATAOUT DD DSN=USER.EXTRACT.DATA,
//     DISP=(NEW,CATLG,DELETE),
//     UNIT=SYSDA,SPACE=(CYL,(5,2),RLSE)),
//     DCB=(RECFM=FB,LRECL=49,BLKSIZE=21952)
```

Figure 60 shows the COBOL coding for the examples shown in Figure 57 and Figure 59.

Figure 60  Sample COBOL for extracting date for trial balance report (part 1 of 3)
Figure 60  Sample COBOL for extracting date for trial balance report (part 2 of 3)

```cobol
OBJECT-COMPUTER. IBM-370.

INPUT-OUTPUT SECTION.
FILE-CONTROL.
  SELECT DATAOUT ASSIGN TO UT-S-DATAOUT.

DATA DIVISION.
FILE SECTION.

FD DATAOUT
  BLOCK CONTAINS 0 RECORDS
  RECORD CONTAINS 49 CHARACTERS
  RECORDING MODE IS F
  LABEL RECORDS STANDARD.

01 DATA-REC.
  05 OUT-TRANNO          PIC X(5).
  05 OUT-ACCTNO          PIC X(8).
  05 OUT-SHORTNAME       PIC X(20).
  05 OUT-BALANCE         PIC S9(9)V99.

WORKING-STORAGE SECTION.

77 FUNC-INIT            PIC X(4)    VALUE 'INIT'.
77 FUNC-GET             PIC X(4)    VALUE 'GET'.
77 FUNC-EOF           PIC X(4)    VALUE 'EOF'.
77 STATUS             PIC XX      VALUE  SPACES.

01 WS-IOAREA.
  05 IO-SEGNAM      PIC X(8).
  05 IO-SSPTRS      PIC X       OCCURS 8 TIMES.
  05 IO-SEGDATA     PIC X(3976).

  05 ROOT-SEG       REDEFINES IO-SEGDATA.
  10 ROOT-LL        PIC S9(4)   COMP.
  10 ROOT-KEY.
  15 TRAN-NO        PIC X(5).
  15 ACCT-NO        PIC X(8).
  .
  10 SHORTNAME      PIC X(20)
  .
  10 BALANCE        PIC S9(9)V99.
  .

PROCEDURE DIVISION.

A100-MAINLINE.
  OPEN OUTPUT DATAOUT.
  CALL 'FABEUR7' USING FUNC-INIT.
```
The DL/I segment edit/compression routine that can be invoked by FABEUR6 and FABEUR7 is indicated by the DLICOMP control statement in the control card file. Refer to the IMS/ESA Customization Guide: Database and the IMS/ESA Database Administration Guide for details on the use and requirements of the routine.

Convention of registers on entry

The register interface convention as documented in the IMS/ESA Customization Guide is adhered to except for Registers 1 and 4:

- Register 1
  - documented as the address of the process scheduling table (PST)
  - the PST is unavailable in a batch environment
  - Fast Path Reorg/EP will set Register 1 to negative one
Entry code convention

Fast Path Reorg/EP follows and supports all entry codes as documented in the *IMS/ESA Customization Guide: Database*.

**FABGXDR DEDB data extractor I/O module**

FABGXDR is the I/O module called by user-written programs to retrieve the extracted data from an OS sequential file. It also generates an audit control report detailing the number of segments read from each input DD file. FABGXDR has two processing modes:

- The single-file mode allows FABGXDR to operate using function calls (INIT, GET, EOJ) and JCL compatible with previous releases.
- The multi-file processing mode allows up to nine input files to be read concurrently. This mode requires function codes with a numeric suffix in the fourth character (INIx, GETx, EOJx). The numeric suffix must correspond to an XDRDATAx DD statement where $x = \{1, 2, 3, \ldots 9\}$. Each input DD can contain data from a different database and can contain SORT-type, non-SORT-type, or Extract format records.

**JCL requirements for FABGXDR**

FABGXDR is a called module. In addition to the DD statements required by the calling program, the following DD statements are required:

**STEPLIB DD**

Required. A concatenation of the following libraries:

- The load library in which the Fast Path Online Analyzer/EP Data Extractor programs reside.
- The library where the user-written program resides.
Parameter lists for FABGXDR

XDRDATA DD / XDRDATAx DD

Required. Define the input data sets containing the data extracted by the EXTRACT command, where \( x = \{1, 2, ..., 9\} \). This data set can reside on DASD or tape.

XDRPRINT DD

Required. Defines the output data set for the messages and audit trail report. The data set can reside on DASD or printer, or it can be routed through the output stream. Use RECFM=FBA and LRECL=121.

Parameter lists for FABGXDR

FABGXDR is called with a parameter list consisting of one to three parameters. All three parameters are required for GET/GETx function calls.

Parameter 1

```
01 XDRPARM1.
   05 FUNCTION             PIC X(04).
   05 FUNC REDEFINES FUNCTION.
      10 FILLER             PIC X(03).
      10 FUNC-SUFFIX        PIC X.
   05 STATUS              PIC X(02).
   05 SEGMENT-CODE        PIC S9(4) COMP.
   05 SEGMENT-NAME        PIC X(08).
```

- Valid function codes are INIT/INIx, GET/GETx and EOJ/EOJx.
- Valid status codes are GB (end-of-file) and spaces.
- The calling program is responsible only for setting the function code to the appropriate value.

Parameter 2

```
01 XDRPARM2.
   05 KEY-LENGTH           PIC S9(4) COMP.
   05 KEY-DATA             PIC X(3840).
```

KEY-DATA is set to the concatenated key of the returned segment or data items. It must be equal or greater in length than the longest possible concatenated key.
Parameter 3

<table>
<thead>
<tr>
<th>01 XDRPARM3.</th>
</tr>
</thead>
<tbody>
<tr>
<td>05 SEGDATA-LENGTH PIC S9(4) COMP.</td>
</tr>
<tr>
<td>05 SEGDATA    PIC X(28552).</td>
</tr>
</tbody>
</table>

If SEGDATA contains a complete segment, the SEGDATA-LENGTH field and the first two bytes of SEGDATA contain the length of the segment.

Initialization function

The initialization function opens the required data sets and formats some internal tables using control information passed in the first few records. The INIT (INIx) function must be done before any GET (GETx) function.

In the COBOL PROCEDURE DIVISION, code:

```
MOVE 'INIT' TO FUNCTION.
CALL 'FABGXDR' USING XDRPARM1.
MOVE 'GET ' TO FUNCTION.
```

or

```
MOVE 'INIT' TO FUNCTION.
MOVE '1' TO FUNC-SUFFIX.
CALL 'FABGXDR' USING XDRPARM1.
MOVE 'GET ' TO FUNCTION.
MOVE '1' TO FUNC-SUFX.
```

Empty input files are indicated by a GB status code.

Get data function

The get data function reads the next sequential record in the input file and returns the data previously identified in the three parameters.
End-of-file function

In the COBOL PROCEDURE DIVISION, code:

```
CALL 'FABGXDR' USING XDRPARM1,
     XDRPARM2
     XDRPARM3.
```

End-of-file is indicated by a GB status code.

**End-of-file function**

The EOJ/EOJx function generates an audit trail report and closes the files.

If multiple input files are open, an EOJx function call will close only the corresponding XDRDATAx DD file. The XDRPRINT DD file is closed only after the last input DD is closed.

A GLOBAL close can be performed by issuing an EOJ function call. This closes all open files and generates the audit reports.

In the COBOL PROCEDURE DIVISION, code:

```
MOVE 'EOJ ' TO FUNCTION.
CALL 'FABGXDR' USING XDRPARM1.
```

**Figure 61** shows a sample JCL job stream for a user-written program.

**Figure 61  Sample JCL for user-written program calling FABGXDR**

```
//REPORT JOB ..... etc
//*
//PFDX0000 EXEC PGM=usrpgm,REGION=0M
//***************************************************************************
//** READ EXTRACTED DATA FROM TWO EXTRACT RUNS
**
***************************************************************************
//STEPLIB    DD DSN=BMC.PFP.LOAD,DISP=SHR
//
//XDRPRINT   DD SYSOUT=*  
//**
//XDRDATA1   DD DSN=BMC.DB1.ODXDATA,DISP=SHR
//XDRDATA2   DD DSN=BMC.DB2.ODXDATA,DISP=SHR
//
```
Sample COBOL program using FABGXDR

Figure 62 shows a COBOL program that uses FABGXDR. See member PFUGXDR in the Fast Path/EP sample library for a COBOL program example.

**Figure 62  Sample COBOL program using FABGXDR (part 1 of 2)**

```
IDENTIFICATION DIVISION.
PROGRAM-ID.SAMP0030 2.
AUTHOR. BMC SOFTWARE.
INSTALLATION. BMC SOFTWARE.
DATE-COMPILED.
REMARKS.
  READ EXTRACTED SEGMENT DATA FROM TWO
  DATABASES AND FORMAT A REPORT.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-370.
OBJECT-COMPUTER. IBM-370.

INPUT-OUTPUT SECTION.
FILE-CONTROL.
DATA DIVISION.
FILE SECTION.
WORKING-STORAGE SECTION.

77 FUNC-INIT PIC X(4) VALUE 'INIT'.
77 FUNC-GET PIC X(4) VALUE 'GET'.
77 FUNC-EOJ PIC X(4) VALUE 'EOJ'.

01 XDRPARM1.
  05 FUNCTION PIC X(4) VALUE SPACES.
  05 FUNC REDEFINES FUNCTION.
    10 FILLER PIC X(03).
    10 FUNC-SUFFIX PIC X.
  05 STATUS PIC X(2) VALUE SPACES.
  05 SEGMENT-CODE PIC S9(4) COMP.
  05 SEGMENT-NAME PIC X(08).
01 XDRPARM2.
  05 KEY-LENGTH PIC S9(4) COMP.
  05 KEY-DATA PIC X(3840).
01 XDRPARM3.
  05 SEGDATA-LENGTH PIC S9(4) COMP.
  05 SEGDATA PIC X(28552).

PROCEDURE DIVISION.

A100-MAINLINE
*******************************************************************************
** INITIALIZATION SECTION **
```
Sample COBOL program using FABGXDR

** MAIN PROCESSING SECTION  **

** END-OF-JOB SECTION **

** PROCESSING SECTION **

CALL 'FABGXDR' USING XDRPARM1, XDRPARM2, XDRPARM3.
IF STATUS = 'GB'
GO TO B100-PROCESS-X.
.
. report formatting
.
.
B100-PROCESS-X.
EXIT.
Callable module considerations

FABEUR6, FABEUR7, FABGXDR utilities are distributed with AMODE ANY and RMODE ANY attributes. This distribution with attributes is done to prevent a residency mode change to the calling program when the callable module is link-edited into the caller (which can happen if the calling program can reside above the 16Mb line).

Problems can arise if the callable modules are not link-edited into the calling program, but invoked via an ATTACH or LINK, or DYNAMIC CALLS. (Some high-level languages can do any or all of these.)

If any of the callable modules are not link-edited into the calling program, these modules should be link-edited with the attributes of AMODE 24 and RMODE 24. This can be accomplished by executing the JCL shown in Figure 63, which also can be found in the Fast Path/EP sample library member name #LKEDCAL on the product distribution tape.

Figure 63  Sample JCL (#LKEDCAL)

```
//#LKEDCAL JOB ETC...
/*
//** JCL TO LINK-EDIT CALLABLE MODULES AS AMODE/RMODE 24
/*
//LKED1 EXEC PGM=IEWL,REGION=0M,
 //  PARM='SIZE=(300K,72K),LIST,MAP,XREF,AMODE=24,RMODE=24'
 //SYSOUT DD SYSOUT=*  
 //SYSPRINT DD SYSOUT=* 
 //SYSLIB DD DSN=BMC.PFP.LOAD,DISP=SHR
 //SYSLMOD DD DSN=your dataset name,DISP=SHR
 //SYSLIN DD *,DCB=BLKSIZE=80
 INCLUDE SYSLIB(FABEUR6)
 ENTRY FABEUR6
 NAME FABEUR6(R)
 INCLUDE SYSLIB(FABEUR7)
 ENTRY FABEUR7
 NAME FABEUR7(R)
 INCLUDE SYSLIB(FABGXDR)
 ENTRY FABGXDR
 NAME FABGXDR(R)
/*
```
This appendix describes the record layout of the history file. This file is generated when the HISTORY_DDNAME keyword is specified with an analysis process.

### NOTE
The history file layout is almost identical to that used by the TRIMAR FAST PATH ANALYZER product, with the exception of the “Date of Analysis” field. In the Fast Path Online Analyzer/EP and Fast Path Analyzer/EP history file, this field has been modified to support dates beginning with the year 2000.

Table 102 provides the record layout of the history file.

$DA#MHSR in the SAMPLIB data set maps the history file record.

#### Table 102  History file record layout (part 1 of 2)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Pos.</th>
<th>Size</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBD Name</td>
<td>1</td>
<td>X(8)</td>
<td></td>
</tr>
<tr>
<td>Area Name</td>
<td>9</td>
<td>X(8)</td>
<td></td>
</tr>
<tr>
<td>Date of Analysis (see note)</td>
<td>17</td>
<td>S9(7)</td>
<td>packed</td>
</tr>
<tr>
<td>filler</td>
<td>21</td>
<td>X(2)</td>
<td></td>
</tr>
<tr>
<td>Low UOW Range Number</td>
<td>23</td>
<td>S9(8)</td>
<td>COMP</td>
</tr>
<tr>
<td>High UOW Range Number</td>
<td>27</td>
<td>S9(8)</td>
<td>COMP</td>
</tr>
<tr>
<td>Number of DB Records</td>
<td>31</td>
<td>S9(9)</td>
<td>packed</td>
</tr>
<tr>
<td>Average Record Length</td>
<td>36</td>
<td>S9(7)</td>
<td>packed</td>
</tr>
<tr>
<td>Maximum Record Length(^a)</td>
<td>40</td>
<td>S9(7)</td>
<td>packed</td>
</tr>
<tr>
<td>Minimum Record length(^a)</td>
<td>44</td>
<td>S9(5)</td>
<td>packed</td>
</tr>
<tr>
<td>P/C Freespace RAABASE</td>
<td>47</td>
<td>S9(3)</td>
<td>packed</td>
</tr>
<tr>
<td>P/C Freespace DOVF</td>
<td>49</td>
<td>S9(3)</td>
<td>packed</td>
</tr>
<tr>
<td>P/C Freespace IOVF</td>
<td>51</td>
<td>S9(3)</td>
<td>packed</td>
</tr>
<tr>
<td>Average Syn Chain Length(^b)</td>
<td>53</td>
<td>S9(5)V99</td>
<td>packed</td>
</tr>
</tbody>
</table>
### Table 102  History file record layout (part 2 of 2)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Pos.</th>
<th>Size</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Syn Chain Length&lt;sup&gt;b&lt;/sup&gt;</td>
<td>57</td>
<td>S9(5)</td>
<td>packed</td>
</tr>
<tr>
<td>Average Root I/O&lt;sup&gt;b&lt;/sup&gt;</td>
<td>60</td>
<td>S9(5)V99</td>
<td>packed</td>
</tr>
<tr>
<td>Maximum Root I/O&lt;sup&gt;b&lt;/sup&gt;</td>
<td>64</td>
<td>S9(5)</td>
<td>packed</td>
</tr>
<tr>
<td>Average Record I/O&lt;sup&gt;a&lt;/sup&gt;</td>
<td>67</td>
<td>S9(5)V99</td>
<td>packed</td>
</tr>
<tr>
<td>Maximum Record I/O&lt;sup&gt;a&lt;/sup&gt;</td>
<td>71</td>
<td>S9(5)</td>
<td>packed</td>
</tr>
<tr>
<td>P/C Available IOVF Cls&lt;sup&gt;c&lt;/sup&gt;</td>
<td>74</td>
<td>S9(3)</td>
<td>packed</td>
</tr>
<tr>
<td>P/C UOWs Using IOVF</td>
<td>76</td>
<td>S9(3)</td>
<td>packed</td>
</tr>
<tr>
<td>Average IOVF Cls Used By a UOW</td>
<td>78</td>
<td>S9(6)V99</td>
<td>packed</td>
</tr>
<tr>
<td>Maximum IOVF Cls Used By a UOW</td>
<td>82</td>
<td>S9(5)</td>
<td>packed</td>
</tr>
<tr>
<td>P/C Records Using IOVF&lt;sup&gt;a&lt;/sup&gt;</td>
<td>85</td>
<td>S9(3)</td>
<td>packed</td>
</tr>
<tr>
<td>P/C RAABASE Cls Using either DOVF or IOVF</td>
<td>87</td>
<td>S9(3)</td>
<td>packed</td>
</tr>
<tr>
<td>P/C UOWs Using DOVF</td>
<td>89</td>
<td>S9(3)</td>
<td>packed</td>
</tr>
<tr>
<td>Average DOVF Cls Used By a UOW</td>
<td>91</td>
<td>S9(5)V99</td>
<td>packed</td>
</tr>
<tr>
<td>Maximum DOVF Cls Used By a UOW</td>
<td>95</td>
<td>S9(5)</td>
<td>packed</td>
</tr>
<tr>
<td>Report Heading</td>
<td>98</td>
<td>x(16)</td>
<td></td>
</tr>
<tr>
<td>P/C Usable Freespace RAABASE</td>
<td>114</td>
<td>S9(3)</td>
<td>packed</td>
</tr>
<tr>
<td>P/C Usable Freespace DOVF</td>
<td>116</td>
<td>S9(3)</td>
<td>packed</td>
</tr>
<tr>
<td>P/C Usable Freespace IOVF</td>
<td>118</td>
<td>S9(3)</td>
<td>packed</td>
</tr>
<tr>
<td>filler</td>
<td>120</td>
<td>X(1)</td>
<td></td>
</tr>
<tr>
<td>UOW-1 Value</td>
<td>121</td>
<td>S9(4)</td>
<td>COMP</td>
</tr>
<tr>
<td>UOW-2 Value</td>
<td>123</td>
<td>S9(4)</td>
<td>COMP</td>
</tr>
<tr>
<td>Root-1 Value</td>
<td>125</td>
<td>S9(8)</td>
<td>COMP</td>
</tr>
<tr>
<td>Root-2 Value</td>
<td>129</td>
<td>S9(8)</td>
<td>COMP</td>
</tr>
<tr>
<td>No. SDEP Cls&lt;sup&gt;c&lt;/sup&gt;</td>
<td>133</td>
<td>S9(8)</td>
<td>COMP</td>
</tr>
<tr>
<td>CI size&lt;sup&gt;c&lt;/sup&gt;</td>
<td>137</td>
<td>S9(4)</td>
<td>COMP</td>
</tr>
<tr>
<td>filler</td>
<td>139</td>
<td>X(35)</td>
<td></td>
</tr>
<tr>
<td>Fragmentation Factor</td>
<td>174</td>
<td>S(5)V99</td>
<td>packed</td>
</tr>
<tr>
<td>IOVF free within area</td>
<td>178</td>
<td>S9(5)</td>
<td>packed</td>
</tr>
<tr>
<td>P/C RAPs with space for Root</td>
<td>181</td>
<td>S9(3)</td>
<td>packed</td>
</tr>
<tr>
<td>P/C free space in SDEP portion&lt;sup&gt;c&lt;/sup&gt;</td>
<td>183</td>
<td>S9(3)</td>
<td>packed</td>
</tr>
<tr>
<td>Repository Group</td>
<td>185</td>
<td>X(4)</td>
<td></td>
</tr>
<tr>
<td>filler</td>
<td>188</td>
<td>X(12)</td>
<td></td>
</tr>
</tbody>
</table>

<sup>a</sup> These items are set to zero unless POINTER_VALIDATION=FULL has been specified.

<sup>b</sup> These items are set to zero unless POINTER_VALIDATION=FULL or RAP_VALIDATION=XREF has been specified.

<sup>c</sup> These items are computed for the entire area, and are not dependent on any UOW range specified.
This section describes the record layout for the DEDB Data Extract file produced using the EXTRACT command as discussed in the *Fast Path Online Suite User Guide* and the *Fast Path Offline Suite User Guide*. It also shows the default output format of the EXTRACT command, which is EXTRACT_FORMAT=EXTRACT. For more information, see the following tables:

- **Table 103** contains the record layout of the DBD Control Information Record Definition for SORT=NO.
- **Table 104** contains the record layout of the Extracted Segment Record Definition for SORT=NO.
- **Table 105** contains the record layout of the DBD Control Information Record Definition for SORT=YES.
- **Table 106** contains the record layout of the Extracted Segment Record Definition for SORT=YES.

### Table 103  DEDB Data Extract record layout 1: DBD control information record definition for SORT=NO (part 1 of 2)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XDR1RDW</td>
<td></td>
<td>RDW LL Field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RDW ZZ Field</td>
</tr>
<tr>
<td>XDR1TYP</td>
<td>X(1)</td>
<td>X’40’ – Flag indicating SORT=NO</td>
</tr>
<tr>
<td></td>
<td>X(1)</td>
<td>X’00’ – Alignment Byte</td>
</tr>
<tr>
<td>XDR1SC</td>
<td>X(2)</td>
<td>X’0000’ – Segment Code</td>
</tr>
<tr>
<td>XDR1ST#E</td>
<td>X(2)</td>
<td>Number of Segment Table Entries in this record</td>
</tr>
<tr>
<td>XDR1STE</td>
<td></td>
<td>Table Entry occurs four times</td>
</tr>
<tr>
<td>XDR1STSC</td>
<td>X(2)</td>
<td>Segment Code</td>
</tr>
<tr>
<td>XDR1STHL</td>
<td>X(2)</td>
<td>Segment Hierarchical Level</td>
</tr>
</tbody>
</table>
### Table 103 DEDB Data Extract record layout 1: DBD control information record definition for SORT=NO (part 2 of 2)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XDR1STPS</td>
<td>X(2)</td>
<td>Segment Parent Segment Code</td>
</tr>
<tr>
<td>XDR1STNM</td>
<td>CL8</td>
<td>Segment Name</td>
</tr>
</tbody>
</table>

### Table 104 DEDB Data Extract record layout 2: extracted segment record definition for SORT=NO

<table>
<thead>
<tr>
<th>Field name</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XDR1RDW</td>
<td></td>
<td>RDW LL Field</td>
</tr>
<tr>
<td>XDR1TYP</td>
<td>X(1)</td>
<td>X’40’ – Flag indicating SORT=NO</td>
</tr>
<tr>
<td>XDR1SC</td>
<td>X(2)</td>
<td>Segment Code</td>
</tr>
<tr>
<td>XDR1DCKL</td>
<td>X(2)</td>
<td>Concatenated Key Length (ckl)</td>
</tr>
<tr>
<td>XDR1DCKY</td>
<td>X(ckl)</td>
<td>Concatenated Key Data</td>
</tr>
<tr>
<td>XDR1DSDL</td>
<td>X(2)</td>
<td>Extracted Segment Data Length (sdl)</td>
</tr>
<tr>
<td>XDR1DSD</td>
<td>X(sdl)</td>
<td>Extracted Segment Data</td>
</tr>
</tbody>
</table>

### Table 105 DEDB Data Extract record layout 3: DBD control information record definition for SORT=YES

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XDR2RDW</td>
<td></td>
<td>RDW LL Field</td>
</tr>
<tr>
<td>XDR2TYP</td>
<td>X(1)</td>
<td>C’S’ – Flag indicating SORT=YES</td>
</tr>
<tr>
<td>XDR2BCF</td>
<td>X(2)</td>
<td>Base Correction Factor. Add to XDR Base Register to map data part of record</td>
</tr>
</tbody>
</table>
| XDR2SCSQ    | variable| SCSQ Table. The number of entries is one less than the number of hierarchical levels defined in the DMB. (There is no entry for the root segment.)
|             |       | ■ 1st byte is Segment Code of segment in this record.                      |
|             |       | ■ 2nd-4th bytes are a sequential counter of the occurrence of               |
|             |       | this segment type in this database record.                                 |
| XDR2DATA    |       | Data Portion of Record                                                      |
| XDR2ST#E    | X(2)  | Number of Segment Table Entries in this record                             |
| XDR2STE     |       | Table entry occurs four times                                               |
| XDR2STSC    | X(2)  | Segment Code                                                                |
| XDR2STHL    | X(2)  | Segment Hierarchical Level                                                  |
| XDR2STPS    | X(2)  | Segment Parent Segment Code                                                 |
| XDR2STNM    | X(8)  | Segment Name                                                                |
### Table 106  DEDB Data Extract record layout 4: extracted segment record definition for SORT=YES

<table>
<thead>
<tr>
<th>Field name</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XDR2RDW</td>
<td></td>
<td>RDW LL Field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RDW ZZ Field</td>
</tr>
<tr>
<td>XDR2TYP</td>
<td>X(1)</td>
<td>C’S’ – Flag indicating SORT=YES</td>
</tr>
<tr>
<td></td>
<td>X(1)</td>
<td>X’00’ – Alignment Byte</td>
</tr>
<tr>
<td>XDR2BCF</td>
<td>X(2)</td>
<td>Base Correction Factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Add to XDR Base Register to map data part of record</td>
</tr>
<tr>
<td>XDR2RKL</td>
<td>X(2)</td>
<td>Root Segment Key Length</td>
</tr>
<tr>
<td>XDR2RKV</td>
<td>variable</td>
<td>Root Segment Key</td>
</tr>
<tr>
<td>XDR2SCSQ</td>
<td>variable</td>
<td>SCSQ Table. The number of entries is one less than the number of hierarchical levels defined in the DMB. (There is no entry for the root segment.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1st byte is Segment Code of segment in this record.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2nd-4th bytes are a sequential counter of the occurrence of this segment type in this database record.</td>
</tr>
<tr>
<td>XDR2DATA</td>
<td></td>
<td>Data Portion of Record</td>
</tr>
<tr>
<td>XDR2SC</td>
<td>X(2)</td>
<td>Segment Code</td>
</tr>
<tr>
<td>XDR2DCKL</td>
<td>X(2)</td>
<td>Concatenated Key Length (ckl) excluding root key</td>
</tr>
<tr>
<td>XDR2DCKY</td>
<td>X(ckl)</td>
<td>Concatenated Key Data excluding root key</td>
</tr>
<tr>
<td>XDR2DSDL</td>
<td>X(2)</td>
<td>Extracted Segment Data Length (sdl)</td>
</tr>
<tr>
<td>XDR2DSD</td>
<td>X(sdl)</td>
<td>Extracted Segment Data</td>
</tr>
</tbody>
</table>
DEDB unload/reload record layouts

This appendix discusses the following topics:

- HD unload/reload file record layout ........................................... 541
- TFMT unload/reload file record layout ........................................... 541
- DBT unload/reload file record layout ........................................... 543

HD unload/reload file record layout

The standard IMS HD Unload data set contains one record for each database segment unloaded, plus a header and trailer record. Each record consists of a prefix (not the segment’s prefix) and segment data. The HD Unload file record layout format is contained in the IMSVS.ADFSMAC(DFSURGUF) IMS macro library.

TFMT unload/reload file record layout

The TFMT file provides compatibility with the BMC Software TRIMAR FAST PATH UNLOAD/RELOAD product. This file is generated by either of the following processes:

- when the FORMAT=TFMT keyword is specified with a Fast Path Reorg/EP unload process
- when FORMAT=TFMT is specified in the utility control statement of the Fast Path Reorg/EP FABEUR6 program extension prior to execution
Table 107 provides defines the record layout of the TFMT unload/reload file.

**Table 107  TFMT unload/reload record layout  (part 1 of 2)**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Assembler definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RDW FIELDS:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RDW</td>
<td>H</td>
<td>RDW LL field</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>RDW ZZ field</td>
</tr>
<tr>
<td><strong>SORT KEY PART OF RECORD:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USRA#</td>
<td>H</td>
<td>-area number</td>
</tr>
<tr>
<td>USRRAP</td>
<td>XL4</td>
<td>-RAP RBA</td>
</tr>
<tr>
<td>USRLCFLG</td>
<td>X</td>
<td>-Insert limit count flag</td>
</tr>
<tr>
<td></td>
<td></td>
<td>==&gt; set to X'00'</td>
</tr>
<tr>
<td>USRBCF</td>
<td>XL2</td>
<td>-Base correction factor used to reset USR base register to access data part of record.</td>
</tr>
<tr>
<td>USRRKL</td>
<td>XL2</td>
<td>-root seg key length</td>
</tr>
<tr>
<td>USRRKV</td>
<td>XL?</td>
<td>-root seg key value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>==&gt; ?? is the length of the root segment key field</td>
</tr>
<tr>
<td>USRLCG#</td>
<td>XL2</td>
<td>-Insert Limit Count group</td>
</tr>
<tr>
<td></td>
<td></td>
<td>==&gt; set to X'0000'</td>
</tr>
<tr>
<td>USRSCSQL</td>
<td>n(XL1,XL3)</td>
<td>-SCSQ table entry. The number of entries is one less than the number of hierarchical levels defined in the database. (There is no entry for the root segment.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>==&gt; 1st byte is Seg Code of segment in this record</td>
</tr>
<tr>
<td></td>
<td></td>
<td>==&gt; 2nd-4th bytes are a sequential counter of the occurrence of this segment type in this database record.</td>
</tr>
<tr>
<td><strong>DATA PORTION OF RECORD:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USRPFLG2</td>
<td>X</td>
<td>-processing flag 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Bits set if segment is target of a subset pointer.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bits are set left to right for pointers 1 - 8.</td>
</tr>
<tr>
<td>USRSEGCD</td>
<td>H</td>
<td>-segment code</td>
</tr>
<tr>
<td>USRSHLVL</td>
<td>H</td>
<td>-hierarchical level of segment</td>
</tr>
<tr>
<td>USRPSCD</td>
<td>H</td>
<td>-hierarchical parent’s seg code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>==&gt; Seg Code of the segment whose PCF or PTF pointer is used to retrieve this segment via GN processing</td>
</tr>
</tbody>
</table>
The DBT file provides compatibility with the DEDB Reload Utility component of the IBM Fast Path Basic Tools for OS/390. This file is generated by either of the following processes:

- when the FORMAT=DBT keyword is specified with a Fast Path Reorg/EP unload process
- when FORMAT=DBT is specified in the utility control statement of the Fast Path Reorg/EP FABEUR6 program extension prior to execution

Table 108 provides defines the record layout of the DBT format unload/reload file

<table>
<thead>
<tr>
<th>Field name</th>
<th>Assembler definition</th>
<th>Description</th>
</tr>
</thead>
</table>
| USRPFLG1     | H                    | -Processing Flag 1
                |                      | 0: Ignore                                                             |
                |                      | Non-zero: more segments on Twin Chain (But 'ISRT LIMCT' Reached).        |
                |                      | Contains ILCG#.                                                         |
| USRSDATA     | XL???                | -Segment Data                                                            |
                |                      | ==&gt; 1st two bytes of USRSDATA is the length of the segment            |

Table 107 TFMT unload/reload record layout (part 2 of 2)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Assembler definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;RDW&quot; FIELDS:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RDW</td>
<td>H</td>
<td>RDW LL field</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>RDW ZZ field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SORT KEY PART OF RECORD:</td>
</tr>
<tr>
<td>USRA#</td>
<td>H</td>
<td>-area number</td>
</tr>
<tr>
<td>USRRAP</td>
<td>XL4</td>
<td>-RAP RBA</td>
</tr>
<tr>
<td>USRLCFLG</td>
<td>X</td>
<td>-Insert limit count flag</td>
</tr>
<tr>
<td></td>
<td></td>
<td>==&gt; set to X'00'</td>
</tr>
<tr>
<td>USRRKL</td>
<td>XL2</td>
<td>-root seg key length</td>
</tr>
<tr>
<td>USRRKV</td>
<td>XLnn</td>
<td>-root seg key value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>==&gt; nn is the length of the root segment key field</td>
</tr>
</tbody>
</table>
Table 108  DBT unload/reload record layout  (part 2 of 3)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Assembler definition</th>
<th>Description</th>
</tr>
</thead>
</table>
| USRLCG#    | XL2                  | -Insert Limit Count group  
  == set to X'0000'  |
| USRSCSQ2   | XL1,XL3              | -Seg-Code/Seg Seq. Field for Hierarchical Level 2 Direct Dependent segments (binary zeros for Root Segment)  
  ==> first byte is seg-code of segment in this segment data record  
  ==> second-fourth bytes are a sequential counter of the occurrence of this segment type within this database record.  |
<p>| USRSCSQ3   | XL1,XL3              | -Seg-Code/Seg Seq. Field for Hierarchical Level 3 Direct Dependent segments (binary zeros for Root Segment)  |
| USRSCSQ4   | XL1,XL3              | -Seg-Code/Seg Seq. Field for Hierarchical Level 4 Direct Dependent segments (binary zeros for Root Segment)  |
| USRSCSQ5   | XL1,XL3              | -Seg-Code/Seg Seq. Field for Hierarchical Level 5 Direct Dependent segments (binary zeros for Root Segment)  |
| USRSCSQ6   | XL1,XL3              | -Seg-Code/Seg Seq. Field for Hierarchical Level 6 Direct Dependent segments (binary zeros for Root Segment)  |
| USRSCSQ7   | XL1,X3               | -Seg-Code/Seg Seq. Field for Hierarchical Level 7 Direct Dependent segments (binary zeros for Root Segment)  |
| USRSCSQ8   | XL1,XL3              | -Seg-Code/Seg Seq. Field for Hierarchical Level 8 Direct Dependent segments (binary zeros for Root Segment)  |
| USRSCSQ9   | XL1,XL3              | -Seg-Code/Seg Seq. Field for Hierarchical Level 9 Direct Dependent segments (binary zeros for Root Segment)  |
| USRSCSQA   | XL1,XL3              | -Seg-Code/Seg Seq. Field for Hierarchical Level A Direct Dependent segments (binary zeros for Root Segment)  |
| USRSCSQB   | XL1,XL3              | -Seg-Code/Seg Seq. Field for Hierarchical Level B Direct Dependent segments (binary zeros for Root Segment)  |
| USRSCSQC   | XL1,XL3              | -Seg-Code/Seg Seq. Field for Hierarchical Level C Direct Dependent segments (binary zeros for Root Segment)  |
| USRSCSQD   | XL1,XL3              | -Seg-Code/Seg Seq. Field for Hierarchical Level D Direct Dependent segments (binary zeros for Root Segment)  |
| USRSCSQE   | XL1,XL3              | -Seg-Code/Seg Seq. Field for Hierarchical Level E Direct Dependent segments (binary zeros for Root Segment)  |
| USRSCSQF   | XL1,XL3              | -Seg-Code/Seg Seq. Field for Hierarchical Level F Direct Dependent segments (binary zeros for Root Segment)  |</p>
<table>
<thead>
<tr>
<th>Field name</th>
<th>Assembler definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USRPFLG1</td>
<td>X</td>
<td>-processing flag 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X’01’: More Segs on Twin Chain</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(But ‘ISRT LIMCT’ Reached)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X’02’: Unloaded Seg is compressed</td>
</tr>
<tr>
<td>USRSEGCD</td>
<td>H</td>
<td>-Segment Code</td>
</tr>
<tr>
<td>USRSHLVL</td>
<td>H</td>
<td>-Hierarchical level of segment</td>
</tr>
<tr>
<td>USRPSCD</td>
<td>H</td>
<td>-Hierarchical parent’s Seg Code</td>
</tr>
<tr>
<td></td>
<td></td>
<td>==&gt; Seg Code of the segment whose PCF or PTF pointer is used to retrieve</td>
</tr>
<tr>
<td></td>
<td></td>
<td>this segment via “GN” processing</td>
</tr>
<tr>
<td>USRPFLG2</td>
<td>H</td>
<td>-Processing Flag 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bit(s) set if segment is target of a Subset pointer. bits are set left to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>right for pointers 1 - 8</td>
</tr>
<tr>
<td>Blank</td>
<td>X</td>
<td>1 byte filler</td>
</tr>
<tr>
<td>USRSDATA</td>
<td>XLnnn</td>
<td>-Segment Data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>==&gt; 1st two bytes of USRSDATA is the length of the segment</td>
</tr>
</tbody>
</table>
Discard file record layout

This appendix describes the record layout of the discard file. This file is generated when the EXCEPTION_LIMIT keyword and the DISCARD_FILECTL subcommand are specified with a command process.

**NOTE**

A user-written program will need to be created to process the discarded data.

- **Table 109** contains the record layout for the DMCB definition.

  The first record in this file contains database definition information.

- **Table 110** contains the record layout for the SDBF definition.

  The next set of records in this file contains segment definition information—one record for each segment type that is defined in the database.

- **Table 111** contains the record layout for the segment data.

  There is one record for each discarded segment following each of these header records.

- **Table 112** contains a list of exception codes, along with a description of each code.

**Table 109  Discard file record layout 1: DMCB definition (part 1 of 2)**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL</td>
<td>H</td>
<td>standard RECFM=V record length</td>
</tr>
<tr>
<td>ZZ</td>
<td>2X</td>
<td>standard RECFM=B record flags</td>
</tr>
<tr>
<td>exception</td>
<td>CL4</td>
<td>binary zeros (LOW-VALUES)</td>
</tr>
<tr>
<td>identifier</td>
<td>CL4</td>
<td>“DMCB”</td>
</tr>
<tr>
<td>max ckeyl</td>
<td>H</td>
<td>maximum concatenated key length for database</td>
</tr>
</tbody>
</table>
Table 109  Discard file record layout 1: DMCB definition (part 2 of 2)

<table>
<thead>
<tr>
<th>Field name</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS level</td>
<td>2X</td>
<td>IMS release level</td>
</tr>
<tr>
<td>filler</td>
<td>16X</td>
<td>binary zeros (LOW-VALUES)</td>
</tr>
<tr>
<td>DMCB</td>
<td>variable</td>
<td>IMS DMCB header for the database</td>
</tr>
</tbody>
</table>

Note: This format is defined by the DBFDMCB macro, and might change between IMS releases.

Table 110  Discard file record layout 2: SDBF definition

<table>
<thead>
<tr>
<th>Field name</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL</td>
<td>H</td>
<td>standard RECFM=V record length</td>
</tr>
<tr>
<td>ZZ</td>
<td>2X</td>
<td>standard RECFM=B record flags</td>
</tr>
<tr>
<td>exception</td>
<td>CL4</td>
<td>binary zeros (LOW-VALUES)</td>
</tr>
<tr>
<td>identifier</td>
<td>CL4</td>
<td>“SDBF”</td>
</tr>
<tr>
<td>filler</td>
<td>20X</td>
<td>binary zeros (LOW-VALUES)</td>
</tr>
<tr>
<td>SDBF</td>
<td>variable</td>
<td>IMS SDBF for the segment</td>
</tr>
</tbody>
</table>

Note: This format is defined by the DBFDMCB macro, and might change between IMS releases.

Table 111  Discard file record layout 3: segment data

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL</td>
<td>H</td>
<td>standard RECFM=V record length</td>
</tr>
<tr>
<td>ZZ</td>
<td>2X</td>
<td>standard RECFM=B record flags</td>
</tr>
<tr>
<td>exceptiona</td>
<td>4C</td>
<td>exception code</td>
</tr>
<tr>
<td>segcode</td>
<td>H</td>
<td>segment code</td>
</tr>
<tr>
<td>seglevel</td>
<td>H</td>
<td>segment hierarchical level</td>
</tr>
<tr>
<td>ckeyl</td>
<td>H</td>
<td>concatenated key length for this segment</td>
</tr>
<tr>
<td>segll</td>
<td>H</td>
<td>segment data length</td>
</tr>
<tr>
<td>segname</td>
<td>CL8</td>
<td>segment name</td>
</tr>
<tr>
<td>filler</td>
<td>8X</td>
<td>binary zeros (LOW-VALUES)</td>
</tr>
<tr>
<td>ckeyb</td>
<td>CLn</td>
<td>concatenated key</td>
</tr>
<tr>
<td>segdatac</td>
<td>variable</td>
<td>segment data</td>
</tr>
</tbody>
</table>

a For a list of exception codes and descriptions, see Table 112.

b The size of the ckey field is fixed based on the maximum length for the concatenated key in the database. For more information, see the max ckeyl field in the DMCB definition record in Table 109. The length of the ckey field is the smallest multiple of eight that is greater than, or equal to the maximum length of the concatenated key for any segment in the database. When the actual length of the concatenated key is less than the maximum length, unused bytes in ckey field will be filled with binary zeros (LOW-VALUES).
The segment data is written in the same form as it appears in the I/O area of an application program. As a result, any compressed data sets will be expanded, variable length segments will contain the 2-byte field, and the fixed length segments will not contain a length field.

The exception code field (see Table 111) contains a 4-character code, indicating the type of exception that caused the segment to be discarded. Table 112 contains a list of the codes, along with descriptions.

Table 112 Exception codes

<table>
<thead>
<tr>
<th>Code name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>zeros</td>
<td>None (LOW-VALUES). This is a header record.</td>
</tr>
<tr>
<td>blanks</td>
<td>None. This segment is a dependent of a segment for which an exception has been detected.</td>
</tr>
<tr>
<td>DUP</td>
<td>Duplicate key</td>
</tr>
<tr>
<td>AREA</td>
<td>Target output area not in OAREA list (root segment only)</td>
</tr>
<tr>
<td>RAND</td>
<td>Randomizer error (root segment only)</td>
</tr>
<tr>
<td>SEQ</td>
<td>Key sequence error</td>
</tr>
<tr>
<td>MINL</td>
<td>Segment length less than the minimum</td>
</tr>
<tr>
<td>MAXL</td>
<td>Segment length greater than the maximum</td>
</tr>
</tbody>
</table>
Appendix E  Command syntax diagrams

This appendix contains syntax diagrams for the most frequently used Fast Path/EP commands. Use it as a quick reference to determine available functions, correct syntax, available keywords, and their defaults/parameters.

NOTE
You can remove this appendix from this reference manual and copy it for use as a quick desk reference.

This appendix includes the following topics:

- ANALYZE command .......................................................... 553
  Subcommands available with ANALYZE .................................. 554
- BACKOUT command .......................................................... 555
  Subcommands available with BACKOUT ................................. 555
- BUILD command .............................................................. 556
  Subcommand available with BUILD ...................................... 556
- CHANGE command ........................................................... 557
  Subcommands available with CHANGE .................................. 559
- EXTEND command ........................................................... 560
  Subcommands available with EXTEND .................................. 562
- EXTRACT command ........................................................... 563
  Subcommands available with EXTRACT .................................. 564
- GLOBAL command ............................................................ 565
  Subcommands available with GLOBAL .................................. 567
- IMAGECOPY command ....................................................... 568
  Subcommands available with IMAGECOPY .............................. 569
- INITIALIZE command ....................................................... 569
  Subcommands available with INITIALIZE .............................. 570
- PFPSORT command ........................................................... 571
- PREPARE command .......................................................... 572
  Subcommands available with PREPARE .................................. 574
- PROCESS_AREA command ................................................... 574
  Subcommand available with PROCESS_AREA ............................ 575
RELOAD command ................................................................. 576
  Subcommands available with RELOAD .............................. 578
REORGANIZE command ......................................................... 579
  Subcommands available with REORGANIZE ....................... 580
RESTART command ............................................................. 581
  Subcommand available with RESTART ............................... 581
RESTRUCTURE command ....................................................... 582
  Subcommand available with RESTRUCTURE ....................... 583
RESYNC command ............................................................... 584
  Subcommand available with RESYNC ............................... 585
RETRIEVE command ............................................................ 585
  Subcommand available with RETRIEVE ............................. 586
SHADOW_INIT command .......................................................... 586
  Subcommands available with SHADOW_INIT .................... 586
STATUS command ................................................................. 587
  Subcommand available with STATUS ............................... 587
UNLOAD command ............................................................... 588
  Subcommands available with UNLOAD ............................. 590
VERIFY command ................................................................. 591
  Subcommand available with VERIFY ............................... 592
XSCAN command ................................................................. 592
  Subcommand available with XSCAN ............................... 593
ANALYZE command

- **DBD**=*dbname***
- **IAREA**=
  - **areaname**
  - **areanumber**
- **RANGE**= ( *startarea,endarea* )
- **INPUT_DSN_MASK**=
  - *parameter*
  - 'LATEST_BATCH_IMAGECOPY'
- **ACCESS**=
  - OFFLINE
  - CONCURRENT
- **POINTER_VALIDATION**=
  - QUICK
  - FULL
  - OFF
- **RAP_VALIDATION**=
  - ( *NOXREF*, *XREF*, *NOPLACEMENT*, *PLACEMENT* )
- **SDEP_VALIDATION**=
  - QUICK
  - FULL
  - NONE
  - OFF
- **ORPHANED_SDEP_MSG**=
  - ERROR
  - NOMSG
  - INFORMATIONAL
  - WARNING
- **MODEL_DD_NAME**=
  - *ddname*
- **HISTORY_DDNAME**=
  - *ddname*
- **ICACHE**=
  - ( *IOVF*, *SDEP* )
ANALYZE command (continued)

Subcommands available with ANALYZE

NOTE
For syntax and keywords that are available for these subcommands, see Appendix F, “Subcommand syntax diagrams.”
BACKOUT command

Subcommands available with BACKOUT

**NOTE**
For syntax and keywords that are available for these subcommands, see Appendix F, “Subcommand syntax diagrams.”
BUILD command

Subcommand available with BUILD

NOTE
For syntax and keywords that are available for this subcommand, see Appendix F, “Subcommand syntax diagrams.”
CHANGE command

- **DBD**: dbdname*
- **IAREA**: ALL, (area, areanumber), RANGE= - (startarea, endarea)
- **INPUT_DSN_MASK**: parameter, 'LATEST_BATCH_IMAGECOPY'
- **OUTPUT_DSN_MASK**: 'dataset name mask'
- **OAREA**: ALL, (area, areanumber), RANGE= - (startarea, endarea)
- **EXPAND**: NO, YES, (segment name)
- **SDEP_PROCESS**: LOGICAL, PHYSICAL, VSComp, NONE
- **SUBSET_POINTERS**: NO, YES
- **ERROR_THRESHOLD**: 0-99999

Appendix E  Command syntax diagrams  557
CHANGE command (continued)

ICACHE = (ICACHE, ICACHE, ICACHE)

OCACHE = (OCACHE, OCACHE, OCACHE)

POINTER_VALIDATION = QUICK, FULL, OFF, NONE

RAP_VALIDATION = NOXREF, XREF, NOPLACEMENT, PLACEMENT

SDEP_VALIDATION = QUICK, FULL, OFF, NONE

ORPHANED_SDEP_MSG = ERROR, NOMSG, INFORMATIONAL, WARNING

BYPASS_RECORD = NO, YES

LARGEST_DATABASE_RECORDS = 0-32767

INDEX_THREADS = 1-16

IOVF_LOAD_HWM = 50-100

INPUT_THREADS = 1-240

OUTPUT_THREADS = 1-240
Subcommands available with CHANGE

NOTE
For syntax and keywords that are available for these subcommands, see Appendix F, “Subcommand syntax diagrams.”
EXTEND command

DBD=dbname**

IAREA= ALL

areaname

areanumber

RANGE= ( startarea, endarea )

INPUT_DSN_MASK= 'parameter'

LATEST_BATCH_IMAGECOPY'

POINTER_VALIDATION= QUICK

FULL

OFF

NONE

RAP_VALIDATION= ( NOXREF

XREF

NOPLEACEMENT

PLACEMENT )

SDEP_VALIDATION= QUICK

FULL

NONE

OFF

ORPHANED_SDEP_MSG= ERROR

NOMSG

INFORMATIONAL

WARNING
EXTEND command (continued)

```
EXTEND

MESSAGE_SUPPRESSION=

INFORMATIONAL=

WARNING=

ERROR=

CRITICAL=

LARGEST_DATABASE_RECORDS=0-32767

EXTEND_IOVF=  (    units, number    )

EXTEND_SDEP=  (    units, number    )

TYPE_RUN = EXECUTE SIMULATE
```
Subcommands available with EXTEND

NOTE
For syntax and keywords that are available for these subcommands, see Appendix F, “Subcommand syntax diagrams.”
EXTRACT command

For information about available extract formatting options, see the Fast Path Online Suite User Guide or the Fast Path Offline Suite User Guide.
Subcommands available with EXTRACT

NOTE
For syntax and keywords that are available for these subcommands, see Appendix F, “Subcommand syntax diagrams.”
GLOBAL command

```
GLOBAL

ACCESS = ONLINE OFFLINE

DBRC = ( YES NO gsgname )

FLOWER_BOX = YES NO

HISTORY_DDNAME = ( ddname )

POINTER_VALIDATION = QUICK FULL OFF NONE

RAP_VALIDATION = ( NOXREF XREF )

SCAN = NO YES

SDEP_VALIDATION = QUICK FULL OFF NONE

ORPHANED_SDEP_MSG = ERROR NOMSG

INFORMATIONAL WARNING
```
GLOBAL command (continued)

MESSAGE_SUPPRESSION=
INFORMATIONAL=
WARNING=
ERROR=
CRITICAL=

LARGEST_DATABASE_RECORDS=0-32767

EARLY_TERMINATION=

OUTAGE_WINDOW=

TYPE_RUN =

EXECUTE
SIMULATE
Subcommands available with GLOBAL

NOTE
For syntax and keywords that are available for these subcommands, see Appendix F, “Subcommand syntax diagrams.”
IMAGECOPY command

- **DBD=dbname**
- **IAREA=**
  - ALL
  - (areaname, areanumber)
  - RANGE= (startarea, endarea)
- **POINTER_VALIDATION=**
  - QUICK
  - FULL
  - OFF
  - NONE
- **RAP_VALIDATION=**
  - NOXREF
  - XREF
  - NOPLACEMENT
  - PLACEMENT
- **SDEP_VALIDATION=**
  - QUICK
  - FULL
  - NONE
  - OFF
- **ORPHANED_SDEP_MSG=**
  - ERROR
  - NOMSG
  - INFORMATIONAL
  - WARNING
- **LARGEST_DATABASE_RECORDS=0-32767**
Subcommands available with IMAGECOPY

INITIALIZE command

NOTE
For syntax and keywords that are available for these subcommands, see Appendix F, “Subcommand syntax diagrams.”
Subcommands available with INITIALIZE

NOTE

For syntax and keywords that are available for these subcommands, see Appendix F, “Subcommand syntax diagrams.”
**PFPSORT command**

```
PFPSORT

  DBD=dbrname

  INPUT_DSN_MASK= 'parameter'
           LATEST_BATCHE_IMAGECOPY

  SORTSEQUENCE=
     RELATIVE_RAP
     RAP_ASCEND RAP_DESCEND
     KEY_ASCEND KEY_DESCEND
     SDEPSEQ=
           FIFO LIFO

  OUTPUT_DSN_MASK='dataset name mask'

  OAREA=
     ALL
     ( areaname
     , areaname
     , areanumber
     )
     RANGE=
           ( - startarea, endarea - )

  SORT_OPTION= ( parameter )
```
PREPARE command

\[
\text{PREPARE}
\quad \text{DBD} = \text{dbname} **
\quad \text{REQUIRE\_AREA} = \text{ALL}
\quad \text{ACCESS} = \text{ONLINE}
\quad \text{POINTER\_VALIDATION} = \text{QUICK}
\quad \text{RAP\_VALIDATION} = \text{NOXREF}
\quad \text{SDEP\_VALIDATION} = \text{QUICK}
\quad \text{ORPHANED\_SDEP\_MSG} = \text{INFORMATIONAL}
\quad \text{RETIRED\_SUFFIX} = \text{retained suffix name}
\quad \text{SHADOW\_SUFFIX} = \text{shadow suffix name}
\quad \text{SHADOW2\_DSNAME} = \text{shadow2 dataset name}
\]
PREPARE command (continued)

```
PREPARE

MESSAGE_SUPPRESSION=
  INFORMATIONAL=NONE 1-32767 100
  WARNING=NONE 1-32767 100
  ERROR=NONE 1-32767 100
  CRITICAL=NONE 1-32767 100

EARLY_TERMINATION=
  NONE
  BEFORE_RENAME_ADS
  BEFORE_NOTIFY_IC
  BEFORE_START_DATABASE

SHADOW2_SUFFIX=shadow2 suffix name

OUTAGE_WINDOW=
  (start-date-time, end-date-time)

LARGEST_DATABASE_RECORDS=0-32767
```
Subcommands available with PREPARE

NOTE
For syntax and keywords that are available for these subcommands, see Appendix F, “Subcommand syntax diagrams.”

PROCESS_AREA command

```
PROCESS_AREA
  DBD=dbdname*
  IAREA= ALL
  ( areaname
    ( areanumber
      RANGE= ( startarea,endarea - )
    )
  )
  INPUT_DSN_MASK= 'parameter'
                   'LATEST_BATCH_IMAGECOPY'
```
Subcommand available with PROCESS_AREA

NOTE
For syntax and keywords that are available for this subcommand, see Appendix F, “Subcommand syntax diagrams.”
RELOAD command

RELOAD

- DBD=dbname*
- IAREA=
  - ALL
  - (areaname, areanumber)
  - RANGE= (startarea, endarea)
- INPUT_DSN_MASK='dataset name mask'
- OAREA=
  - ALL
  - (areaname, areanumber)
  - RANGE= (startarea, endarea)
- COMRESS=
  - NO
  - YES
  - (segment name)
- EXPAND=
  - NO
  - YES
  - (segment name)
- INPUT_THREADS= 1-240
- INDEX_THREADS= 1-16
- IOVF_LOAD_HWM= 50-100
- OCACHE= (IOVF, SDEP)
RELOAD command (continued)

RELOAD

POINTER_VALIDATION = QUICK, FULL, OFF, NONE

RAP_VALIDATION = NOXREF, XREF, NOPLACEMENT, PLACEMENT

LARGEST_DATABASE_RECORDS = 0-32767

SDEP_PROCESS = V5COMP, NONE

SDEP_VALIDATION = QUICK, FULL, OFF, NONE

ORPHANED_SDEP_MSG = INFORMATIONAL, WARNING

EXCEPTION_LIMIT = 0-2147483647

BYPASS_RECORD = NO, YES

SORT = NO, AUTO

SORT_OPTION = parameter

ERROR_THRESHOLD = 0-99999

OUTPUT_THREADS = 1-240

SUBSET_POINTERS = NO, YES
Subcommands available with RELOAD

NOTE
For syntax and keywords that are available for these subcommands, see Appendix F, “Subcommand syntax diagrams.”
REORGANIZE command

- **DBD=** dbdname*
- **IAREA=** ALL
  - ( areaname
    - areanumber
    - RANGE= ( startarea, endarea )
  )
- **INPUT_DSN_MASK=** 'parameter'
  - 'LATEST_BATCH_IMAGECOPY'
- **SELECT_UOW=** ( IOVF - ' 1-32767 ' 1-32767
  - ALL
  )
- **FRAGMENTATION_PERCENT=** 1-100
- **IOVF_SAVE_THRESHOLD=** 1-32767
- **COMPRESS=** NO
  - YES
  - ( segment name
    )
- **EXTEND_IOVF_#UOWS=** 0-32765
- **EXTEND_SDEP_#CIS=** 0-8338601
- **ERROR_THRESHOLD=** 0-99999
- **POINTER_VALIDATION=** QUICK
  - FULL
  - OFF
  - NONE
- **RAP_VALIDATION=**
  - NOXREF
  - XREF
  - NOPLACEMENT
  - PLACEMENT

*dbdname*
REORGANIZE command (continued)

Subcommands available with REORGANIZE

NOTE
For syntax and keywords that are available for these subcommands, see Appendix F, “Subcommand syntax diagrams.”
RESTART command

Subcommand available with RESTART

NOTE
For syntax and keywords that are available for this subcommand, see Appendix F, “Subcommand syntax diagrams.”
Subcommand available with RESTRUCTURE

NOTE
For syntax and keywords that are available for this subcommand, see Appendix F, “Subcommand syntax diagrams.”
RESYNC command

```plaintext
RESYNC

DBD=dbdname

CHECKPOINT= 10000

IAREA= ALL
   areaname
   areanumber
   RANGE= startarea,endarea

INDEX_THREADS= 1-16

INPUT_DSN_MASK= 'parameter'
   'LATEST_BATCH_IMAGECOPY'

MESSAGE_SUPPRESSION= INFORMATIONAL= NONE
   WARNING= NONE
   ERROR= NONE
   CRITICAL= NONE

SORT= NO
  YES

SORT_OPTION= parameter
```
Subcommand available with RESYNC

NOTE

For syntax and keywords that are available for these subcommands, see Appendix F, “Subcommand syntax diagrams.”

RETRIEVE command

FILE

RETRIEVE

HISTORY_DDNAME= (ddname)

SELECT_GROUP= 'group name mask'

SELECT_DB= 'dataset name mask'

SELECT_AREA= (areaname)

SELECT_LIMIT=0-256

SELECT_DATE= (start date, stop date)
Subcommand available with RETRIEVE

NOTE
For syntax and keywords that are available for this subcommand, see Appendix F, “Subcommand syntax diagrams.”

SHADOW_INIT command

NOTE
For syntax and keywords that are available for these subcommands, see Appendix F, “Subcommand syntax diagrams.”

Subcommands available with SHADOW_INIT
STATUS command

Subcommand available with STATUS

NOTE
For syntax and keywords that are available for this subcommand, see Appendix F, “Subcommand syntax diagrams.”
UNLOAD command

UNLOAD

- DBD=dbname*
- IAREA=
  - ALL
  - areaname
  - areanumber
  - RANGE= ( startarea,endarea )
- INPUT_DSN_MASK= parameter
  - LATEST_BATCH_IMAGECOPY
- OUTPUT_DSN_MASK= dataset name mask
- OAREA=
  - ALL
  - areaname
  - areanumber
  - RANGE= ( startarea,endarea )
- COMPRESS=
  - NO
  - YES
  - segment name
- SDEP_PROCESS=
  - LOGICAL
  - PHYSICAL
  - NONE
- SUBSET_POINTERS=
  - NO
  - YES
UNLOAD command

UNLOAD command (continued)

- **UNLOAD**
  - **FORMAT**
    - HDUNLOAD
    - MSDBINIT
    - TFMT
    - DBT
  - **ERROR_THRESHOLD** = 0-99999
  - **ICACHE**
    - (IOVF, SDEP)
  - **BYPASS_RECORD** = NO, YES
  - **INPUT_THREADS** = 1-240
  - **MESSAGE_SUPPRESSION**
    - (INFORMATIONAL = 1-32767, 100)
    - WARNING = 1-32767, 100
    - ERROR = 1-32767, 100
    - CRITICAL = 1-32767, 100
Subcommands available with UNLOAD

NOTE
For syntax and keywords that are available for these subcommands, see Appendix F, “Subcommand syntax diagrams.”
VERIFY command

- `DBD=dbdname*`
- `IAREA= ALL` or `areaname`
- `ICACHE= (IOVF, SDEP)`
- `INDEX_THREADS= 1-16`
- `INPUT_DSN_MASK= 'parameter'
tribe 'LATEST_BATCH_IMAGECOPY'`
- `CHECKPOINT= 10000`
- `INPUT_THREADS= 1-240`
- `RANGE= startarea,endarea`
- `IAREA= (areaname, areanumber)`
- `SORT= NO, YES`
- `SORT_OPTION= (parameter)`
- `SORT= (parameter)`
Subcommand available with VERIFY

NOTE
For syntax and keywords that are available for this subcommand, see Appendix F, “Subcommand syntax diagrams.”

XSCAN command
Subcommand available with XSCAN

NOTE
For syntax and keywords that are available for this subcommand, see Appendix F, “Subcommand syntax diagrams.”
Subcommand syntax diagrams

This appendix contains syntax diagrams for the most frequently used Fast Path/EP subcommands. Use it as a quick reference to determine which commands are supported by a subcommand, correct subcommand syntax, available keywords and keyword defaults/parameters.

NOTE
You can remove this appendix from this reference manual and copy it for use as a quick desk reference.

This appendix includes the following topics:

Subcommand functions ................................. 596
ACTIVITY_FILECTL subcommand  ..................... 598
ALLOCATE subcommand  .................................. 599
CORRECTIONS_FILECTL subcommand ..................... 600
DISCARD_FILECTL subcommand ......................... 601
EXCLUDE subcommand ................................... 602
INCLUDE subcommand ................................... 602
IC subcommand ......................................... 603
IX subcommand ......................................... 604
LOADCTL subcommand ................................... 604
OFILECTL subcommand .................................. 605
OUTPUT subcommand .................................... 606
PERFORM subcommand .................................. 606
PLAN_FILECTL subcommand ............................. 607
REGISTER subcommand .................................. 608
REPORT subcommand .................................... 609
THRESHOLD subcommand ................................ 610
USER_RECORD subcommand ............................. 611
SPACE keyword ......................................... 611
Subcommand functions

Table 113 provides a brief description of the functions that can be performed by all available Fast Path/EP subcommands, and the commands with which each subcommand can be used.

Table 113 Fast Path/EP subcommand functions (part 1 of 2)

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Function</th>
<th>Can be used with these commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVITY_FILECTL</td>
<td>Control the allocation of the Restructure Activity data set</td>
<td>PREPARE</td>
</tr>
<tr>
<td>ALLOCATE</td>
<td>Dynamically define the VSAM cluster that is used for the command’s output processing</td>
<td>CHANGE, INITIALIZE, RELOAD, SHADOW_INIT</td>
</tr>
<tr>
<td>CORRECTIONS_FILECTL</td>
<td>Request that a pointer corrections output file be created to correct invalid pointers during command processing</td>
<td>ANALYZE</td>
</tr>
<tr>
<td>DISCARD_FILECTL</td>
<td>Request that a discard file be created during command processing</td>
<td>RELOAD</td>
</tr>
</tbody>
</table>
| EXCLUDE               | ■ Identify segment(s) and their dependents to be excluded from processing by primary command  
                          ■ Specify conditional criteria for excluding segment and its dependents from processing by primary command | CHANGE, EXTRACT, RELOAD, UNLOAD |
| IC                    | Request that one or more image copies be created during command processing. The image copies requested are created for each area processed. | ANALYZE, CHANGE, EXTEND, GLOBAL, INITIALIZE, REORGANIZE, UNLOAD, RELOAD |
| INCLUDE               | ■ Identify segment(s) and their dependents to be included for processing by the primary command  
                          ■ Specify conditional criteria for including segment and its dependents for processing by primary command | CHANGE, EXTRACT, RELOAD, UNLOAD |
| IX                    | Specify the index, indexes, or XSCAN data set to be processed by the primary command | BUILD, CHANGE, INITIALIZE, RELOAD, RESYNC, VERIFY, XSCAN |
Table 113  Fast Path/EP subcommand functions (part 2 of 2)

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Function</th>
<th>Can be used with these commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOADCTL</td>
<td>Specify segment placement criteria (for placement into either IOVF or DOVF) during maintenance process specified by primary command</td>
<td>RELOAD, REORGANIZE, CHANGE</td>
</tr>
<tr>
<td>OFILECTL</td>
<td>Define physical, storage and retention attributes of output file(s) created during command processing</td>
<td>EXTRACT, UNLOAD</td>
</tr>
</tbody>
</table>
| OUTPUT           | ■ Identify segment and their dependents to be included for processing by the primary command  
                  ■ Specify conditional criteria for including segments and their dependents for output  
                  ■ Modify the format or content of selected segment data | CHANGE, EXTRACT, PREPARE, RELOAD, UNLOAD              |
| PERFORM          | Execute a script with the Control Interval Dump and Modification Utility | PROCESS_AREA                                         |
| PLAN_FILECTL     | Control the allocation of the Restructure Plan data set                  | PREPARE, RESTART, RESTRUCTURE, SHADOW_INIT            |
| REGISTER         | Register a new area with DBRC that is inserted into or appended to an area list. | PREPARE                                               |
| REPORT           | ■ Create or suppress creation of all analysis reports or specific analysis report(s)  
                  ■ Specify start and end range for UOWs to be included in report or report set  
                  ■ Specify number of lines to print on each page of report | ANALYZE, CHANGE, EXTEND, GLOBAL, IMAGECOPY, PREPARE, RELOAD, REORGANIZE, RETRIEVE |
| THRESHOLD        | Detect exception conditions related to free space, I/O, RAP usage and overflow | ANALYZE, CHANGE, EXTEND, GLOBAL, IMAGECOPY, PREPARE, RELOAD, REORGANIZE, RETRIEVE |
| USER_RECORD      | ■ Specify an output record to be written to an extract file  
                  ■ Control content of segment header/trailer records | EXTRACT (valid only when EXTRACT_FORMAT=USER)          |
ACTIVITY_FILECTL subcommand

DDNAME=ddname

DSNAME=dataset name

AVGREC=

B

U

K

M

DATACLAS=class name

DISP=

NEW

OLD

DELETE

KEEP

CATLG

UNCATLG

DELETE

KEEP

CATLG

UNCATLG

EXPDT=

yyddd

yyyy/ddd

LIKE=dataset name

MGMTCLAS=class name

RETPD=0-32767 days

SPACE=space requirements - See diagram on page 611

STORCLAS=class name

UNIT= device name

1-59

VOLCNT=0-255

VOLSER= volume ID
ALLOCATE subcommand

- **ALLOCATE**

- **OAREA=** 
  - ALL
  - (areaname)
  - areanumber
  - RANGE=
    - (startarea, endarea)

- **CONFIGURE_AREA=**
  - (VOLCNT=0-32)
  - 1-32767

- **CONFIGURE_RAA=**
  - (VOLCNT=0-32)
  - 1-32767

- **CONFIGURE_IOVF=**
  - (VOLCNT=0-32)
  - 1-32767

- **CONFIGURE_SDEP=**
  - (VOLCNT=0-32)
  - 1-32767

- **ACTUATE=**
  - DELETE
  - REUSE
  - RENAME
  - IDCAMS
  - DSN='dataset name mask'
  - EROPT=
    - ABORT
    - IGNORE
  - IDCAMS_OPTION=
    - (parameter)

- **AVGREC=**
  - B
  - U
  - K
  - M

- **DATACLAS=class name**

- **MGMTCLAS=class name**

- **REPORT_DDNAME=**
  - ()

- **SPACE=space requirements** - See diagram on page 611
ALLOCATE subcommand (continued)

ALLOCATE subcommand (continued)

CORRECTIONS_FILECTL subcommand
DISCARD_FILECTL subcommand

DDNAME=ddname

DSNAME=dataset name

AVGREC= B U K M

DATACLAS=class name

DISP= NEW OLD

DELETE KEEP CATLG UNCATLG

DELETE KEEP CATLG UNCATLG

EXPDT= yyddd yyyy/ddd

LIKE=dataset name

MGMTCLAS=class name

RETPD=0-32767 days

SPACE=space requirements - See diagram on page 611

STORCLAS=class name

UNIT= device name 1-59

VOLCNT=0-255

VOLSER= volume ID
EXCLUDE subcommand

```
EXCLUDE

   SEGMENT=segname*
   WHERE= ( Boolean expression )
   SAMPLE_LIMIT=0-2147483647
   SAMPLE_INTERVAL=0-2147483647
```

INCLUDE subcommand

```
INCLUDE

   SEGMENT=segname*
   WHERE= ( Boolean expression )
   SAMPLE_LIMIT=0-2147483647
   SAMPLE_INTERVAL=0-2147483647
```
### IC subcommand

**DDNAME**=`ddname`

**DSNAME**=`dataset name`

**AVGREC**=

- `B`
- `U`
- `K`
- `M`

**DATACLAS**=`class name`

**DISP**=

- `NEW`
- `OLD`
- `DELETE`
- `KEEP`
- `CATLG`
- `UNCATLG`

**EXPDT**=

- `yyddd`
- `yyyy/ddd`

**LIKE**=`dataset name`

**MGMTCLAS**=`class name`

**RETPD**=`0-32767 days`

**SPACE**=`space requirements` - See diagram on page 611

**STACK_NAME**=`stack group name`

**STORCLAS**=`class name`

**UNIT**=

- `device name`
- `1-59`

**VOLCNT**=`0-255`

**VOLSER**=

- `volume ID`

**NOTIFY**=

- `YES`
- `NO`

**COMPRESSION**=

- `NONE`
- `FSE`
- `CCC`
- `DPE`
**IX subcommand**

```
IX INDEX=index database name
     INPUT_DSN_MASK='parameter'
          'LATEST_BATCH_IMAGECOPY'
     OUTPUT_DSN_MASK='dataset name mask'
     DDNAME=ddname
     DSNAME=dataset name
     SORT= NO YES
     SORT_OPTION= ( parameter )
```

**LOADCTL subcommand**

```
LOADCTL INSERT_LIMIT_COUNT=0-32766
     LOCATION= IOVF DOVF
     SEGMENT= ( segname )* ( ONLY DEPENDENTS BOTH )
     WHERE= ( Boolean expression )
```
OFILECTL subcommand

```
OFILECTL

DDNAME=ddname

DSNAME=dataset name

OAREA= ALL,

( areaname, areanumber

RANGE= ( startarea, endarea )

AVGREC= B U K M

DATACLAS=class name

DISP= ( NEW, OLD )

DELETE KEEP CATLG UNCATLG

DELETE KEEP CATLG UNCATLG

EXPDT= yyddd yyyy/ddd

LIKE=dataset name

MGMTCLAS=class name

RETPD=0-32767 days

SPACE=space requirements - See diagram on page 611

STORCLAS=class name

UNIT= ( device name )

VOLCNT=0-255

VOLSER= ( volume ID )
```
OUTPUT subcommand

```
OUTPUT

SEGMENT=segname*

FIELDS=(expression)

WHERE=(Boolean expression)

PERFORM subcommand

PERFORM

SCRIPT={

DMAC_PRINT()

DMCB_PRINT()

VER(expression, expression)

REP(expression, expression)

COMMIT()

ROLLBACK()

SNAP(expression, expression)

}
PLAN_FILECTL subcommand

- DDNAME=ddname
- DSNAME=dataset name
- AVGREC=B, U, K, M
- DATACLASS=class name
- DISP=NEW, OLD
- DELETE, KEEP, CATLG, UNCATLG
- EXPDT=yyddd, yyyy/ddd
- LIKE=dataset name
- MGMTCLASS=class name
- RETPD=0-32767 days
- SPACE=space requirements - See diagram on page 611
- STORCLASS=class name
- UNIT=device name, 1-59
- VOLCNT=0-255
- VOLSER=volume ID
REGISTER subcommand

REGISTER

OAREA=(area-list)

ADDN=name

ADSN='dsname-mask'

DEFLTJCL=name

FULLSEG= YES

NO

GENMAX=number

GSGNAME=name

ICJCL=name

PREOPEN= YES

NO

RECOVJCL=name

RECOVPD=number

RECVJCL=name

REUSE= YES

NO

TRACK= DB

RCV

VSO= YES

NO

CFSTR1=name

CFSTR2=name

LKASID= YES

NO

MAS= YES

NO

PRELOAD= YES

NO
REPORT subcommand

- REPORT_DEFAULT=
  - YES
  - NO

- POINTER_ANALYSIS=
  - YES
  - NO

- RECORD_LENGTH_ANALYSIS=
  - YES
  - NO

- RECORD_LENGTH_INCREMENT=1-2000000

- RECORD_PLACEMENT_ANALYSIS=
  - YES
  - NO

- RECORD_PROFILE_ANALYSIS=
  - YES
  - NO

- SEGMENT_IO_ANALYSIS=
  - YES
  - NO

- SEGMENT_LENGTH_ANALYSIS=
  - NO
  - YES
  - (segment name)

- SEGMENT_PLACEMENT_ANALYSIS=
  - YES
  - NO

- SYNONYM_CHAIN_ANALYSIS=
  - YES
  - NO

- SYNONYM_CHAIN_INCREMENT=1-100

- UOW_DETAILED_ANALYSIS=
  - YES
  - NO

- START_UOW=UOW number

- STOP_UOW=UOW number

- REPORT_LINE_COUNT=16-32767

- REPORT_DDNAME= (ddname)

- REPORT_HEADING=title
THRESHOLD subcommand

DOVF_FREESPACE_PERCENT=1-100

FREESPACEDOVF_IOVF=(0-100,0-100)

FREESPACERAA_DOVF=(0-100,0-100)

FREESPACERAA_IOVF=(0-100,0-100)

IOVF_USED_PERCENT=1-100

RAA_FREESPACE_PERCENT=1-100

RAP_OVERFLOW_PERCENT=1-100

RECORD_IO_AVERAGE=1-32767

RECORD_IO_MAXIMUM=1-32767

RECORD_IOVF_PERCENT=1-100

ROOT_IO_AVERAGE=1-32767

ROOT_IO_MAXIMUM=1-32767

SYNONYM_CHAIN_LENGTH=1-32767

SYNONYM_CHAIN_MAXIMUM=1-32767

SYNONYM_CHAIN_PERCENT=1-100

UOW_DOVF_PERCENT=1-100

UOW_IOVF_AVERAGE=1-32767

UOW_IOVF_MAXIMUM=1-32767

UOW_IOVF_PERCENT=1-100
USER_RECORD subcommand

The diagram that follows is an expansion of the diagrams for these subcommands, which are shown on page 603 and page 605.

SPACE keyword

The diagram that follows is an expansion of the diagrams for these subcommands, which are shown on page 603 and page 605.
Appendix G  Repository maintenance command and subcommand syntax diagrams

This appendix contains syntax diagrams for the commands and subcommands that can be executed under the PFPEPR00 repository maintenance program. Use it as a quick reference to determine correct command/subcommand syntax, available keywords and keyword defaults/parameters.

NOTE
You can remove this appendix from this reference manual and copy it for use as a quick desk reference.

This appendix includes the following topics:

Subcommand functions .............................................................. 613
PROCESS_EPR command .......................................................... 615
   Subcommands available with PROCESS_EPR .............................. 615
ADD subcommand ................................................................. 616
DELETE subcommand .............................................................. 618
LIST subcommand ................................................................. 619
MODIFY subcommand ............................................................. 620
OVERRIDE subcommand .......................................................... 622
RESET subcommand ............................................................... 622
RETRIEVE subcommand .......................................................... 623
   Subcommand available with RETRIEVE ................................. 623

Subcommand functions

Table 114 provides a brief description of the functions that can be performed by all available PFPEPR00 commands and subcommands. Each of the subcommands listed in Table 114 must be specified under the PROCESS_EPR primary command.
### Table 114  PFPEPR00 command and subcommand functions

<table>
<thead>
<tr>
<th>Command or subcommand</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESS_EPR command</td>
<td>Specify the name of the repository catalog to be processed by PFPEPR00</td>
</tr>
<tr>
<td>ADD subcommand</td>
<td>Add a new allocation rule to the repository catalog at a global, group, DBD, or area level</td>
</tr>
<tr>
<td></td>
<td>Specify the details of the allocation rule, including how the statistics data set name is to be determined, details about the type of storage unit to be used, class criteria, retention period, and specific volume serial numbers</td>
</tr>
<tr>
<td></td>
<td>Add a new statistics catalog entry to the repository catalog</td>
</tr>
<tr>
<td>DELETE subcommand</td>
<td>Delete an allocation rule from the repository catalog data set</td>
</tr>
<tr>
<td></td>
<td>Delete a statistics catalog entry from the repository catalog</td>
</tr>
<tr>
<td>LIST subcommand</td>
<td>List allocation rules contained in the repository catalog data set at a global, group, DBD, or area level</td>
</tr>
<tr>
<td></td>
<td>List all allocation rules contained in the repository catalog</td>
</tr>
<tr>
<td></td>
<td>List statistics catalog entries contained in the repository catalog by group, DBD, or area</td>
</tr>
<tr>
<td></td>
<td>List statistics catalog entries contained in the repository catalog for a specified creation date or a specific date range (start date, stop date)</td>
</tr>
<tr>
<td></td>
<td>List all message customizations contained in the repository catalog</td>
</tr>
<tr>
<td>MODIFY subcommand</td>
<td>Modify data set name, details about the type of storage unit to be used, class criteria, retention period, and specific volume serial numbers for an allocation rule</td>
</tr>
<tr>
<td></td>
<td>Globally modify all allocation rules contained in the repository catalog per specified criteria</td>
</tr>
<tr>
<td>OVERRIDE subcommand</td>
<td>Dynamically specify and retain customizations of severity level suffixes or suppression threshold for eligible messages</td>
</tr>
<tr>
<td></td>
<td>Override previously stored message customizations</td>
</tr>
<tr>
<td>RESET subcommand</td>
<td>Restore all message customizations to product defaults</td>
</tr>
<tr>
<td></td>
<td>Restore customizations for selected messages to product default</td>
</tr>
<tr>
<td>RETRIEVE subcommand</td>
<td>Retrieve analysis statistics stored in the repository (in conjunction with the REPORT subcommand)</td>
</tr>
</tbody>
</table>
PROCESS_EPR command

Subcommands available with PROCESS_EPR

ADD
DELETE
LIST
MODIFY
OVERRIDE
RESET
RETRIEVE

REPOSITORY_DSNAMEx='dataset name'
ADD subcommand

ADD subcommand

GLOBAL_ALLOCATION

GROUP_ALLOCATION

DSNAME=dataset name*

UNIT= ( device name* - 1-59 )

DATACLAS=class name

MGMTCLAS=class name

STORCLAS=class name

RETPD=0-32767 days

VOLSER= ( volume ID )

EXPDT= yyddd yyyy/ddd

GROUP_KEY=groupname*

VOLSER= ( volume ID )

EXPDT= yyddd yyyy/ddd
ADD subcommand (continued)

ADD subcommand

ADD

DBD_ALLOCATION

GROUP_KEY=groupname*

DBD_KEY=database name*

DSNAME=dataset name*

UNIT=  

DATACLASS=class name

MGMTCLASS=class name

STORCLASS=class name

RETPD=0-32767 days

VOLSER=  

EXPDT= yyddd

yyyy/ddd

AREA_ALLOCATION

GROUP_KEY=groupname*

DBD_KEY=database name*

AREA_KEY=areaname*

DSNAME=dataset name*

UNIT=  

DATACLASS=class name

MGMTCLASS=class name

STORCLASS=class name

RETPD=0-32767 days

VOLSER=  

EXPDT= yyddd

yyyy/ddd

STATISTICS

DSNAME=dataset name
DELETE subcommand

DELETE subcommand
LIST subcommand

LIST — GLOBAL_ALLOCATION

GLOBAL_ALLOCATION — SELECT_GROUP=groupname*

GROUP_ALLOCATION — SELECT_GROUP=groupname*, SELECT_DATABASE=database name*

DBD_ALLOCATION — SELECT_GROUP=groupname*, SELECT_DATABASE=database name*

AREA_ALLOCATION — SELECT_GROUP=groupname*, SELECT_DATABASE=database name*, SELECT_AREA=areaname*

ALLOCATION — SELECT_GROUP=groupname, SELECT_DATABASE=database name, SELECT_AREA=areaname

STATISTICS — DSNAME=dataset name mask*, SELECT_GROUP=groupname, SELECT_DATABASE=database name, SELECT_AREA=areaname

SELECT_DATE= { start date, stop date }

HISTORY_DDNAME= { ddname }

MESSAGE_OVERRIDE
MODIFY subcommand

GLOBAL_ALLOCATION

DSNAME=dataset name mask
UNIT= ( device name*, 1-59 )
DATAclas=class name
MGMTCLAS=class name
STORCLAS=class name
RETPD=0-32767 days
VOLSER= ( volume ID )
EXPDT= yyddd yyyyy/ddd

SELECT_GROUP=groupname

GROUP_ALLOCATION

MODIFY

UNIT= ( device name*, 1-59 )
DATAclas=class name
MGMTCLAS=class name
STORCLAS=class name
RETPD=0-32767 days
VOLSER= ( volume ID )
EXPDT= yyddd yyyyy/ddd
MODIFY subcommand (continued)
OVERRIDE subcommand

```
CURSE
OVERRIDE
MESSAGE_NUMBER=number
MESSAGE_LEVEL=
  INFORMATIONAL
  WARNING
  ERROR
  CRITICAL
MESSAGE_LIMIT=0-32767
```

RESET subcommand

```
CURSE
RESET
MESSAGE_NUMBER=ALL
```

RETRIEVE subcommand

Subcommand available with RETRIEVE

NOTE
For syntax and keywords that are available for this subcommand, see Appendix F, “Subcommand syntax diagrams.”
Subcommand available with RETRIEVE
Sample utility and command scenarios

How to interpret the scenarios .......................................................... 626
JCL and control statement ............................................................... 626
Descriptive text .............................................................................. 626
Sample library JCL ......................................................................... 627
Segment hierarchy for sample DEDB .............................................. 627
DBD for sample DEDB ................................................................. 627
Scenario conceptual descriptions .................................................. 628
Expressions ................................................................................... 631
Customize unload output using expressions ............................... 631
Customize offline extract output using expressions ...................... 633
Control Interval Dump and Modification Utility ............................. 636
Use the SNAP function to dump control interval ....................... 636
Repair pointer values online using VER and REP functions .......... 637
Print fields within in-core DMAC .................................................. 638
Area Change Modeling Utility ...................................................... 638
Model effects of changing randomizer for all areas .................... 639
Model effects of changing UOW and ROOT values for selected areas 640
Model effects of changing control interval size for an area and control segment placement ........................................ 641
Model effects of converting from full-function to DEDB format .... 642
File Sort Utility ............................................................................ 643
Sort file in ascending order by root key ........................................ 643
Sort file in ascending order by RAP and sort logical SDEPs in reverse sequence . 644
How to interpret the scenarios

The scenarios in this section show how to use selected utilities and expression language that are discussed in this manual. Each scenario presents JCL and an associated control statement that combines a primary command with other elements of the Fast Path/EP 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 64  JCL and control statement

<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/EP 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>
<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 REORGANIZE. 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>
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. The following figure shows a hierarchy diagram of the segments that are defined in the DBD for each area in the PFPSAMP database.

Figure 65 Segment hierarchy for sample DEDB

The DBD for the PFPSAMP sample database is shown on page 627.

DBD for sample DEDB

Figure 66 DBD for sample DEDB (part 1 of 2)
Scenario conceptual descriptions

Table 116 provides an overview and page number reference for each of the scenarios that are provided in this appendix.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEGM NAME=SEGB, BYTES=(60,13)</td>
<td>X</td>
</tr>
<tr>
<td>COMPRTN=(DPIFPRTN,DATA,INIT)</td>
<td>X</td>
</tr>
<tr>
<td>PARENT=SEGA, TYPE=DIR</td>
<td></td>
</tr>
<tr>
<td>FIELD NAME=(SEGBKEY,SEQ,U), BYTES=8, START=3, TYPE=C</td>
<td></td>
</tr>
<tr>
<td>SEGM NAME=SEGC, BYTES=(50,13)</td>
<td>X</td>
</tr>
<tr>
<td>FIELD NAME=(SEGCCKEY,SEQ,U), BYTES=8, START=3, TYPE=C</td>
<td></td>
</tr>
<tr>
<td>SEGM NAME=SEGD, BYTES=(55,13)</td>
<td>X</td>
</tr>
<tr>
<td>FIELD NAME=(SEGDKEY,SEQ,U), BYTES=8, START=3, TYPE=C</td>
<td></td>
</tr>
<tr>
<td>SEGM NAME=SEGE, BYTES=(150,13)</td>
<td>X</td>
</tr>
<tr>
<td>FIELD NAME=(SEGEKEY,SEQ,U), BYTES=8, START=3, TYPE=C</td>
<td></td>
</tr>
<tr>
<td>SEGM NAME=SEGF, BYTES=(90,14)</td>
<td>X</td>
</tr>
<tr>
<td>FIELD NAME=(SEGFKEY,SEQ,U), BYTES=9, START=3, TYPE=C</td>
<td></td>
</tr>
<tr>
<td>SEGM NAME=SEGG, BYTES=(40,7)</td>
<td>X</td>
</tr>
<tr>
<td>FIELD NAME=(SEGGKEY,SEQ,U), BYTES=2, START=3, TYPE=C</td>
<td></td>
</tr>
</tbody>
</table>

DBDGEN
FINISH
END
### Table 116  Scenario tasks (part 1 of 3)

<table>
<thead>
<tr>
<th>Utility/command/scenario task</th>
<th>Subcommand/keyword</th>
<th>Concept/process</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UNLOAD</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customize Unload Output</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Using Expressions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>UNLOAD</strong></td>
<td>REFORMAT</td>
<td>specify output file</td>
<td>631</td>
</tr>
<tr>
<td>Extract (by default)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INCLUDE SEGMENT with WHERE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXCLUDE SEGMENT with WHERE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OUTPUT SEGMENT with WHERE and FIELDS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OUTPUT SEGMENT with FIELDS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OFILECTL with OAREA and DDNAME</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Offline EXTRACT</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customize Offline Extract</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output Using Expressions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Offline EXTRACT</strong></td>
<td>REFORMAT</td>
<td>specify output file</td>
<td>633</td>
</tr>
<tr>
<td>Extract (by default)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INCLUDE SEGMENT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAMPLE_LIMIT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INCLUDE SEGMENT with WHERE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OUTPUT SEGMENT with FIELDS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USER_RECORD with BREAK / BEFORE FIELDS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OFILECTL with OAREA and DSNAME</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 116 Scenario tasks (part 2 of 3)

<table>
<thead>
<tr>
<th>Utility/command/scenario task</th>
<th>Subcommand/keyword</th>
<th>Concept/process</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Interval Dump and Modification Utility (PROCESS_AREA command)</td>
<td>PERFORM subcommand SNAP function</td>
<td>■ snap dump a specified control interval</td>
<td>636</td>
</tr>
<tr>
<td>Use the SNAP Function to Dump Control Interval</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Control Interval Dump and Modification Utility (PROCESS_AREA command) | PERFORM subcommand VER function REP function | ■ use VERify and REPLace function to verify and replace corrupted pointer values  
■ commit repaired (replaced) pointer values to DASD | 637 |
| Repair Pointer Values Online Using VER and REP Functions | | | |
| Control Interval Dump and Modification Utility (PROCESS_AREA command) | PERFORM subcommand DMAC_PRINT function | ■ perform in-core DMAC print of specified area | 638 |
| Print Fields within In-Core DMAC | | | |
| Area Change Modeling Utility (PFMD0100 Program) | CTL control statement: DBDNAME / AREA RPT RMOD | ■ specify a different randomizer than the randomizer currently specified in DBD  
■ request report reflecting detailed record count | 639 |
| Model Effects of Changing Randomizer for All Areas | | | |
| Area Change Modeling Utility (PFMD0100 Program) | CTL control statement: DBDNAME / AREA ROOT UOW | ■ increase ROOT and UOW values for specific areas | 640 |
| Model Effects of Changing UOW and Root Values for Selected Areas | | | |
| Area Change Modeling Utility (PFMD0100 Program) | CTL control statement: DBDNAME / AREA CISIZE SEG control statement: NAME LIMITCT | ■ change size of control interval for one area  
■ limit the number of segments to be placed in RAA | 641 |
| Model Effects of Changing Control Interval Size for an Area  
Control Segment Placement | | | |
This section provides control statement examples using expressions with Fast Path/EP unload and online extract processes. Collectively, these examples show numerous elements of the Fast Path/EP expression language that are available for customizing the content or format of the output file. For purposes of exemplifying the customization of included segments, the scenarios in this section use the segment hierarchy shown on page 627.

### Customize unload output using expressions

Only the data from two root key values will be selected by this scenario, with further exclusion on SEGC. The input areas are dynamically allocated. This control statement could also be performed to customize the output of a RELOAD, CHANGE, or EXTRACT process.

Figure 67  JCL to customize unload output using expressions (part 1 of 2)

```plaintext
01 //PFPUNLD  EXEC  PGM=PFPMAIN
02 //STEPLIB  DD  DISP=SHR,DSN=BMC.PFP.LOAD
```
Customize unload output using expressions

Figure 67   JCL to customize unload output using expressions (part 2 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>03-04</td>
<td>EXEC and STEPLIB DD statements for Fast Path/EP offline execution.</td>
</tr>
<tr>
<td>05</td>
<td>The DBD from the IMSACB DD statement is used to define the input areas.</td>
</tr>
<tr>
<td>06</td>
<td>The DBD from the NEWACB DD statement is used to define the output unload file.</td>
</tr>
<tr>
<td>07-09</td>
<td>All output areas are written to one unload file, which is specified by the ULD1CPY1 DD statement.</td>
</tr>
<tr>
<td>11</td>
<td>Defines the DBD name for the unload process. All areas are unloaded.</td>
</tr>
<tr>
<td>12-13</td>
<td>Limit the output file by selecting only SEGA root segments with key values ‘4563’ or ‘4598’ and their dependents. Elements of this expression are:</td>
</tr>
<tr>
<td></td>
<td>- SEGKEY EQ C’4563’ - field variable/comparison operator/typed literal</td>
</tr>
<tr>
<td></td>
<td>- OR - Boolean operator</td>
</tr>
<tr>
<td></td>
<td>- SEGKEY EQ C’4598’ - field variable/comparison operator/typed literal</td>
</tr>
<tr>
<td>14</td>
<td>Exclude from selection all SEGc segments that have a value less than ‘31’ starting in column 10. Elements of this expression are:</td>
</tr>
<tr>
<td></td>
<td>- 10:2C - column/length/data-type (field variable)</td>
</tr>
<tr>
<td></td>
<td>- &lt; - comparison operator</td>
</tr>
<tr>
<td></td>
<td>- C’31’ - typed literal</td>
</tr>
<tr>
<td>15-17</td>
<td>OUTPUT subcommands identify the modified segments to be output. All other selected database segments are written to the output file, but are not changed.</td>
</tr>
</tbody>
</table>
Customize offline extract output using expressions

This scenario uses the EXTRACT_FORMAT=USER option to customize the output file. This scenario is executing an offline EXTRACT command, but could also be used for an online EXTRACT (if you have a license for the BMC Software Fast Path Online Analyzer/EP product). The input areas are dynamically allocated. The segments in the database are compressed, so EXPAND=YES is specified for the output files.

Table 117  Descriptive text for JCL to customize unload output using expressions (part 2 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>With the FIELDS keyword specified, the data for SEGB will be altered when written to the output file. Copy the first 5 bytes starting at column 3, then the search fields SEGBFLD2 and SEGBFLD5, and then write the remainder of the data starting at column 25 of the input to the end of the segment (as indicated by *). Elements of this expression are</td>
</tr>
<tr>
<td></td>
<td>- 3:5C - column/length/data-type (field variable)</td>
</tr>
<tr>
<td></td>
<td>- SEGBFLD2 - DBD-defined variable</td>
</tr>
<tr>
<td></td>
<td>- SEGBFLD5 - DBD-defined variable</td>
</tr>
<tr>
<td></td>
<td>- 25:* - column/length (field variable) using default character data-type</td>
</tr>
<tr>
<td>16</td>
<td>Segment SEGD will be written to the output file only when the value at column 4 is 'B' and the value at column 10 is 'R'. Elements of this expression are</td>
</tr>
<tr>
<td></td>
<td>- 4:1C and 10:1C - column/length/data-type (field variables)</td>
</tr>
<tr>
<td></td>
<td>- EQ - comparison operator</td>
</tr>
<tr>
<td></td>
<td>- AND - Boolean operator</td>
</tr>
<tr>
<td></td>
<td>- C'B' and C'R' - typed literals</td>
</tr>
<tr>
<td>17</td>
<td>With the FIELDS keyword specified, the data for SEGD will be altered when written to the output file. Copy the first 8 bytes starting at column 3, assign the area number to the next two bytes, and then write the remainder of the data starting at column 13 of the input to the end of the segment (as indicated by *). Elements of this expression are</td>
</tr>
<tr>
<td></td>
<td>- 3:8C - column/length/data-type (field variable)</td>
</tr>
<tr>
<td></td>
<td>- :2X - assignment variable</td>
</tr>
<tr>
<td></td>
<td>- := - assignment operator</td>
</tr>
<tr>
<td></td>
<td>- AREA_NUMBER - built-in variable</td>
</tr>
<tr>
<td></td>
<td>- 13:* - column/length (field variable) using default character data-type</td>
</tr>
<tr>
<td>18</td>
<td>All areas are written to a single unload output file using the provided DD name.</td>
</tr>
</tbody>
</table>

Customize offline extract output using expressions

Figure 68  JCL to customize offline extract output using expressions (part 1 of 2)

01 //PFPPEXTR EXEC PGM=PFPMAIN
02 //STEPLIB DD DISP=SHR,DSN=PFP.DIST.LOAD
03 // DD DISP=SHR,DSN=IMSVS.RESLIB
04 // DD DISP=SHR,DSN=IMSVS.DFSMDA
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

Appendix H  Sample utility and command scenarios  633
Customize offline extract output using expressions

Figure 68  JCL to customize offline extract output using expressions (part 2 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>08-04</td>
<td>EXEC and STEPLIB DD statements for Fast Path/EP offline execution.</td>
</tr>
<tr>
<td>05-07</td>
<td>DD statements that define data sets for the BMC Software DATA PACKER/IMS product. These data sets are required because all compressed data is being expanded.</td>
</tr>
<tr>
<td>08</td>
<td>The DBD from the IMSACB DD statement is used to define the areas being processed.</td>
</tr>
<tr>
<td>09</td>
<td>PFPSYSIN DD for Fast Path/EP control statements.</td>
</tr>
<tr>
<td>10</td>
<td>Defines the DEDB name for the EXTRACT process. Data from all areas is being extracted. The format of the output file is user-specified. All compressed data is expanded.</td>
</tr>
<tr>
<td>11</td>
<td>Based on the value specified on the SAMPLE_LIMIT keyword, the first 250 SEGA root segments and their dependents are selected for each area.</td>
</tr>
<tr>
<td>12</td>
<td>Only SEGB segments under the selected SEGA segments that contain two bytes of zeroes (‘00’) at column 3 are selected for output. Elements of this expression are</td>
</tr>
<tr>
<td></td>
<td>- 3:2C - column/length/data-type (field variable)</td>
</tr>
<tr>
<td></td>
<td>- EQ - comparison operator</td>
</tr>
<tr>
<td></td>
<td>- C'00' - typed literal</td>
</tr>
<tr>
<td>13</td>
<td>Only SEGE segments under the selected SEGA segments that have a ‘P’ or ‘B’ at column 3 are selected for output. Elements of this expression are</td>
</tr>
<tr>
<td></td>
<td>- 3:1C - column/length/data-type (field variable)</td>
</tr>
<tr>
<td></td>
<td>- EQ - comparison operator</td>
</tr>
<tr>
<td></td>
<td>- C'P' - typed literal</td>
</tr>
<tr>
<td></td>
<td>- OR - Boolean operator</td>
</tr>
<tr>
<td></td>
<td>- C'B' - typed literal</td>
</tr>
</tbody>
</table>

Table 118  Descriptive text for JCL to customize offline extract output using expressions (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/EP offline execution.</td>
</tr>
<tr>
<td>05-07</td>
<td>DD statements that define data sets for the BMC Software DATA PACKER/IMS product. These data sets are required because all compressed data is being expanded.</td>
</tr>
<tr>
<td>08</td>
<td>The DBD from the IMSACB DD statement is used to define the areas being processed.</td>
</tr>
<tr>
<td>09</td>
<td>PFPSYSIN DD for Fast Path/EP control statements.</td>
</tr>
<tr>
<td>10</td>
<td>Defines the DEDB name for the EXTRACT process. Data from all areas is being extracted. The format of the output file is user-specified. All compressed data is expanded.</td>
</tr>
<tr>
<td>11</td>
<td>Based on the value specified on the SAMPLE_LIMIT keyword, the first 250 SEGA root segments and their dependents are selected for each area.</td>
</tr>
<tr>
<td>12</td>
<td>Only SEGB segments under the selected SEGA segments that contain two bytes of zeroes (‘00’) at column 3 are selected for output. Elements of this expression are</td>
</tr>
<tr>
<td></td>
<td>- 3:2C - column/length/data-type (field variable)</td>
</tr>
<tr>
<td></td>
<td>- EQ - comparison operator</td>
</tr>
<tr>
<td></td>
<td>- C'00' - typed literal</td>
</tr>
<tr>
<td>13</td>
<td>Only SEGE segments under the selected SEGA segments that have a ‘P’ or ‘B’ at column 3 are selected for output. Elements of this expression are</td>
</tr>
<tr>
<td></td>
<td>- 3:1C - column/length/data-type (field variable)</td>
</tr>
<tr>
<td></td>
<td>- EQ - comparison operator</td>
</tr>
<tr>
<td></td>
<td>- C'P' - typed literal</td>
</tr>
<tr>
<td></td>
<td>- OR - Boolean operator</td>
</tr>
<tr>
<td></td>
<td>- C'B' - typed literal</td>
</tr>
</tbody>
</table>

/*

Table118 Descriptive text for JCL to customize offline extract output using expressions (part 1 of 2)
Table 118  Descriptive text for JCL to customize offline extract output using expressions (part 2 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>With the FIELDS keyword specified, the data for SEGA will be altered when written to the output file. The first 5 bytes will have a blank followed by the character value ‘SEGA’, then starting at column 1, 9 bytes will be written to the output file, then the 6-byte packed decimal field at column 4 of input will be converted to character and written to an 11-byte output field. Elements of this expression are</td>
</tr>
<tr>
<td></td>
<td>- ‘SEGA’ - literal</td>
</tr>
<tr>
<td></td>
<td>- 1:9C - column/length/data-type (field variable)</td>
</tr>
<tr>
<td></td>
<td>- :11C - assignment variable, used as length and data-type placeholder for result of data conversion (packed decimal to character)</td>
</tr>
<tr>
<td></td>
<td>- := - assignment operator</td>
</tr>
<tr>
<td></td>
<td>- 4:6P - column/length/data-type (field variable)</td>
</tr>
<tr>
<td>15-16</td>
<td>Only the previously selected SEGE segments that contain a character ‘V’, ‘A’, or ‘B’ in column 4 will be written as specified in the FIELDS statement on line 17. Elements of this expression are</td>
</tr>
<tr>
<td></td>
<td>- 4:1C - column/length/data-type (field variable)</td>
</tr>
<tr>
<td></td>
<td>- EQ - comparison operator</td>
</tr>
<tr>
<td></td>
<td>- OR - Boolean operator</td>
</tr>
<tr>
<td></td>
<td>- C’V’, C’A’, C’B’ - typed literals</td>
</tr>
<tr>
<td>17</td>
<td>With the FIELDS keyword specified, the data for SEGE will be altered when written to the output file. The first 5 bytes will have a blank followed the character value ‘SEGE’, then 2 bytes will be written to the output file starting at column 3 of the input, then followed by 10 bytes starting in column 8 of input, and then 3 bytes starting in column 31 of input. Elements of this expression are</td>
</tr>
<tr>
<td></td>
<td>- ‘SEGE’ - literal</td>
</tr>
<tr>
<td></td>
<td>- 3:2C - column/length/data-type (field variable)</td>
</tr>
<tr>
<td></td>
<td>- 8:10 - column/length (field variable) using default character data-type</td>
</tr>
<tr>
<td></td>
<td>- 31:3 - column/length (field variable) using default character data-type</td>
</tr>
<tr>
<td>18</td>
<td>The header (first) record is created by specifying the USER_RECORD subcommand. The BREAK keyword specifies that this record is to be written to the extract file before the remainder of the database is written (DATABASE,BEFORE). The FIELDS keyword specifies the content of the header record using the following expression elements:</td>
</tr>
<tr>
<td></td>
<td>- X’0000’ - literal (record identifier)</td>
</tr>
<tr>
<td></td>
<td>- DBD_NAME - built-in variable, 8 bytes = ‘PFPSAMP’</td>
</tr>
<tr>
<td>19-20</td>
<td>All areas are written to a single extract output file using the data set name created from the variables specified in the mask:</td>
</tr>
<tr>
<td></td>
<td>- DSN=BMCPFP.PFPSAMP.D01171.T103455</td>
</tr>
<tr>
<td></td>
<td>- &amp;DBD=PFPSAMP</td>
</tr>
<tr>
<td></td>
<td>- &amp;DATE=current date in format ‘DyYddd’</td>
</tr>
<tr>
<td></td>
<td>- &amp;TIME=current time in format ‘Thhmmss’</td>
</tr>
</tbody>
</table>
Control Interval Dump and Modification Utility

The scenarios in this section show how to use the Control Interval Dump and Modification Utility (PROCESS_AREA command) in conjunction with key related keywords and subcommands.

Use the SNAP function to dump control interval

Analysis of a Fast Path area has indicated an invalid physical child first pointer in a segment at RBA 01AF4C in the PFP.SAMP database. The area data set and ACB library are accessed using dynamic allocation. The command set will dump the CI containing the record at RBA 01AF4C.

Figure 69  JCL to use the SNAP function to dump control interval

```jcl
01//PFP EXEC  PGM=PFPMAIN,REGION=0M
02//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
03//      DD DISP=SHR,DSN=IMSVS.RESLIB
04//      DD DISP=SHR,DSN=IMSVS.DFSMDA
05//PFPSYSIN DD *
06//PROCESS_AREA DBD=PFPSAMP,IAREA=PFPSAMP1
07 PERFORM SCRIPT={ SNAP (X'01AF4C'); }
/*
```

Table 119  Descriptive text for JCL to use the SNAP function to dump control interval

<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 offline execution.</td>
</tr>
<tr>
<td>05</td>
<td>PFPSYSIN DD for Fast Path/EP control statements.</td>
</tr>
<tr>
<td>06</td>
<td>Defines DEDB name for PROCESS_AREA command. The SNAP function will be performed for area PFPSAMP1 only.</td>
</tr>
<tr>
<td>07</td>
<td>The CI containing the record at RBA 01AF4C is snap dumped.</td>
</tr>
</tbody>
</table>
Repair pointer values online using VER and REP functions

In this scenario, online analysis has detected problems with area PFPSAMP3. Some pointers have been corrupted at RBAs 02DF3E, 01ECA4, 0043E2, and 1E34D2. A dump of the CIs at these RBAs has shown the following pointer values: 0000000E, FF240000, 40404040 and C3C2C5E4. Analysis of the CI dump has determined that the pointers should have the following values: 0001ECA0, 00004ED8, 001E34C6, and 00000000.

Figure 70  JCL to repair pointer values online using VER and REP functions

Table 120  Descriptive text for JCL to repair pointer values online using VER and REP functions

<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/EP control statements.</td>
</tr>
<tr>
<td>06</td>
<td>Defines DEDB name for PROCESS_AREA command. The VERify and REPlace functions will be performed for area PFPSAMP3.</td>
</tr>
<tr>
<td>07-08</td>
<td>Verify the existence of pointer 0000000E at RBA 02DF3E. Replace the corrupted pointer with the value 0001ECA0.</td>
</tr>
<tr>
<td>09-10</td>
<td>Verify the existence of pointer FF240000 at RBA 01ECA4. Replace the corrupted pointer with the value 00004ED8.</td>
</tr>
<tr>
<td>11-12</td>
<td>Verify the existence of pointer 40404040 at RBA 0043E2. Replace the corrupted pointer with the value 001E34C6.</td>
</tr>
<tr>
<td>13-14</td>
<td>Verify the existence of pointer C3C2C5E4 at RBA 1E34D2. Replace the corrupted pointer with the value 00000000.</td>
</tr>
<tr>
<td>15</td>
<td>The COMMIT (); function must be specified to write (commit) the repaired pointer values to disk.</td>
</tr>
</tbody>
</table>
Print fields within in-core DMAC

The following script prints a formatted listing of the individual fields within the in-core DMAC for the area PFPSAMP2. The area data set and ACB library are accessed using dynamic allocation.

Figure 71 JCL to print fields within in-core DMAC

```
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=IMSVS.DFSMDA
06//PFPSYSIN DD *
07 PROCESS_AREA DBD=PFPSAMP,IAREA=PFPSAMP2
08 PERFORM SCRIPT={ DMAC_PRINT (); }
/*
```

Table 121 Descriptive text for JCL to print fields within in-core DMAC

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-05</td>
<td>EXEC and STEPLIB DD statements for Fast Path/EP online execution.</td>
</tr>
<tr>
<td>06</td>
<td>PFPSYSIN DD for Fast Path/EP control statements.</td>
</tr>
<tr>
<td>07</td>
<td>Defines DEDB name for PROCESS_AREA command. The DMAC print will be performed for area PFPSAMP2 only.</td>
</tr>
<tr>
<td>08</td>
<td>Snap the contents of the in-core DMAC in hexadecimal, and print a formatted list of the individual fields. The ( ); characters are required syntax elements.</td>
</tr>
</tbody>
</table>

Area Change Modeling Utility

The scenarios in this section show different ways of using the Area Change Modeling Utility based on control statements that you specify for the PFMD0100 program module. Each scenario uses a different control statement that you would specify in the JCL based on the desired performance, space, or conversion criteria.

These scenarios provide examples of control statements for the PFMD0100 program only. The following additional control programs and sort steps are required for the Area Change Modeling Utility:

- SORT13
- PFMD0300
- SORT12
- SORT3
- PFMD0500
If you use PFDM0100 control statements that are similar to the examples in these scenarios, you must still code a complete set of JCL to produce the Modeler reports. For a complete set of sample JCL, see “Sample Area Change Modeling Utility JCL” on page 460.

Model effects of changing randomizer for all areas

This scenario models the effect of changing the randomizer for all areas in the PFPSAMP database. A randomizer is specified that is different than the randomizer which is currently specified in the DBD. The resultant reports can help you determine whether the new randomizer performs on the data as expected. Input is an unload file that was created by executing the Fast Path Reorg/EP UNLOAD command.

**Figure 72** JCL to model effects of changing randomizer for all areas

```
01//PFMD0100 EXEC PGM=PFMD0100,REGION=OM
02//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
03// DD DISP=SHR,DSN=IMSVS.RESLIB
04//ACBLIB DD DISP=SHR,DSN=IMSVS.ACBLIB
05//SYSPRINT DD SYSOUT=*  
06//UR7DATA DD DSN=BMC.PFPSAMP.UNLOAD,DISP=SHR 
07//UR7RPT DD SYSOUT=*  
08//MESORTCD DD DSN=&&SORT13,DISP=(NEW,PASS),  
09// UNIT=SYSDA,SPACE=(TRK,(1,1))  
10//MEDARO DD DSN=BMC.PFPSAMP.ME13,  
11// DISP=(NEW,CATLG),  
12// UNIT=SYSDA,SPACE=(CYL,(35,5),RLSE)  
13//SYSIN DD *,DCB=BLKSIZE=80 
14CTL DBDNAME=PFPSAMP,AREA=ALL,  
15CTL RPT=YES,RMOD=DBFHDC40 */
```

**Table 122** Descriptive text for JCL to model effects of changing randomizer for all areas (part 1 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-03</td>
<td>EXEC and STEPLIB DD statements for Area Change Modeling Utility execution.</td>
</tr>
<tr>
<td>04</td>
<td>The ACBLIB DD statement contains the DMB that describes the database to be modeled.</td>
</tr>
<tr>
<td>05</td>
<td>The SYSPRINT output contains the message output.</td>
</tr>
<tr>
<td>06</td>
<td>The input unload file, which is in HD Unload format, contains data for all areas.</td>
</tr>
<tr>
<td>07</td>
<td>Audit report output, which is required when UR7DATA is specified.</td>
</tr>
<tr>
<td>08-09</td>
<td>Sort control cards are passed as a temporary file to step SORT13.</td>
</tr>
<tr>
<td>10-12</td>
<td>The output data set from this step is input to step SORT13.</td>
</tr>
<tr>
<td>13</td>
<td>SYSIN DD statement for PFMD0100 control cards.</td>
</tr>
</tbody>
</table>
Model effects of changing UOW and ROOT values for selected areas

This scenario models the effect of increasing multiple control parameters (the UOW and ROOT values) for areas 1 and 2 of the PFPSAMP database, which might soon be exceeding available space. This scenario shows how the Area Change Modeling utility can be used to help you to more accurately pre-define UOW and RAA parameters before applying the actual changes in your DBD. The input file was created by executing the Fast Path Analyzer/EP ANALYZE command with the MODEL_DDNAME keyword.

This type of scenario could be beneficial to pre-optimize a new database that is being designed to meet desired DASD or performance criteria for a new application.

Table 122  Descriptive text for JCL to model effects of changing randomizer for all areas (part 2 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>All areas are modeled for the PFPSAMP database.</td>
</tr>
<tr>
<td>15</td>
<td>(RPT=YES) requests that the detailed record count report be produced. The newly specified randomizer DBFHDC40 is used to model and produce the Modeler reports.</td>
</tr>
</tbody>
</table>

Model effects of changing UOW and ROOT values for selected areas

Table 123  Descriptive text for JCL to model effects of changing UOW and ROOT values for selected areas (part 1 of 2)

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-03</td>
<td>EXEC and STEPLIB DD statements for Area Change Modeling Utility execution.</td>
</tr>
<tr>
<td>04</td>
<td>The ACBLIB DD statement contains the DMB that describes the database to be modeled.</td>
</tr>
<tr>
<td>05</td>
<td>The SYSPRINT output contains the message output.</td>
</tr>
</tbody>
</table>
This scenario models the effect of increasing the size of the control interval (CISIZE) for area 3 in the PFPSAMP database. This type of scenario could be beneficial when planning for the transfer of data from one type of storage device to another type of storage device. The input file was created by executing the Fast Path Analyzer/EP ANALYZE command with the MODEL_DDNAME keyword.

Figure 74  JCL to model effects of changing control interval size for an area and control segment placement

```plaintext
01//PFMD0100 EXEC PGM=PFMD0100,REGION=0M
02//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
03// DD DISP=SHR,DSN=IMSVS.RESLIB
04//ACBLIB DD DISP=SHR,DSN=IMSVS.ACBLIB
05//SYSPRINT DD SYSOUT=* 
06//MEDARI DD DSN=BMC.PFPSAMP.MODELER,DISP=SHR
07//MESORTCD DD DSN=&&SORT13,DISP=(NEW,PASS), 
08// UNIT=SYSDA,SPACE=(TRK,(1,1))
09//MEDARO DD DSN=BMC.PFPSAMP.ME13,
10// DISP=(NEW,CATLG),
11// UNIT=SYSDA,SPACE=(CYL,(35,5),RLSE)
12//SYSIN DD *,DCB=BLKSIZE=80
13CTL DBDNAME=PFPSAMP,AREA=(3(CISIZE=28672))
14SEG NAME=SEGB,LIMCT=100 */
```
Model effects of converting from full-function to DEDB format

This scenario uses an input file in HD Unload format that was derived from an IMS full-function database. This file can be input to the Area Change Modeling Utility to model the space usage requirements when converting the database to DEDB format.

Figure 75  JCL to model effects of converting from full-function to DEDB format

```plaintext
01//PFMD0100 EXEC PGM=PFMD0100,REGION=0M
02//STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
03// DD DISP=SHR,DSN=IMSVS.RESLIB
04//ACBLIB DD DISP=SHR,DSN=IMSVS.TEST.ACBLIB <=== DEDB
05//DBDLIB DD DISP=SHR,DSN=IMSVS.DBDLIB <=== FULL-FUNCTION
06//SYSPRINT DD SYSOUT=* 
07//UR7DATA DD DSN=BMC.PFPSAMP.UNLOADFF,DISP=SHR
08//UR7CTL DD *
09 // DBDDNAME=PFPSAMP
10//UR7RPT DD SYSOUT=* 
11//MESORTCD DD DSN=&SORT13,DISP=(NEW, PASS),
12// UNIT=SYSDA,SPACE=(TRK,(1,1))
13//MEDARO DD DSN=BMC.PFPSAMP.ME13,
14// DISP=(NEW,CATLG),
15// UNIT=SYSDA,SPACE=(CYL,(35,5),RLSE)
16//SYSIN DD *,DCB=BLKSIZE=80
17CTL DBDDNAME=PFPSAMP,
18AREA=(1(UOW=(40,10),ROOT=(400,15)),2,3(CISIZE=8192)) /*
```
File Sort Utility

The scenarios in this section show how to use the File Sort Utility (PFPSORT command) in conjunction with key related keywords and subcommands.

Sort file in ascending order by root key

In this scenario, the file named PFPSAMP is sorted in ascending order by root key. Sorting the file in this manner can be useful for an application program that requires the data in root key sequence. The data for all three areas of PFPSAMP is contained in the input unload file.

Table 125  Descriptive text for JCL to model effects of converting from full-function to DEDB format

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-03</td>
<td>EXEC and STEPLIB DD statements for Area Change Modeling Utility execution.</td>
</tr>
<tr>
<td>04</td>
<td>The ACBLIB DD statement contains the DMB that describes the database to be modeled. Using a test library is recommended to avoid confusion with the full-function database that has the same name.</td>
</tr>
<tr>
<td>05</td>
<td>Defines the library containing the DBD that describes the IMS full-function database to be modeled. The DBDLIB DD statement is required when the UR7DATA DD statement refers to an unload file that was derived from an IMS full-function database. The DBDLIB DD statement and the ACBLIB DD statement must reference the same DBD name.</td>
</tr>
<tr>
<td>06</td>
<td>The SYSPRINT output contains the message output.</td>
</tr>
<tr>
<td>07</td>
<td>The input HD Unload file, which is from a full-function database, contains data that will be divided among the three DEDB areas in the PFPSAMP database.</td>
</tr>
<tr>
<td>08</td>
<td>Defines the input control statement data set. The UR7CTL DD statement is required when the UR7DATA DD statement refers to a full-function HD Unload file.</td>
</tr>
<tr>
<td>09</td>
<td>Because a full-function file is used as input, the DBD name must be specified on the UR7CTL DD statement. This statement must be specified in addition to the DBD name on SYSIN.</td>
</tr>
<tr>
<td>10</td>
<td>Audit report output, which is required when UR7DATA is specified.</td>
</tr>
<tr>
<td>11-12</td>
<td>Audit report output, which is required when UR7DATA is specified.</td>
</tr>
<tr>
<td>13-15</td>
<td>The output data set from this step is input to step SORT13.</td>
</tr>
<tr>
<td>16</td>
<td>SYSIN DD statement for PFMD0100 control cards.</td>
</tr>
<tr>
<td>17</td>
<td>The database to be modeled is PFPSAMP.</td>
</tr>
<tr>
<td>18</td>
<td>Area 1 is modeled with the specified UOW and ROOT parameters. Areas 2 and 3 are modeled with the specified CISIZE parameter.</td>
</tr>
</tbody>
</table>

Figure 76  JCL to sort files in ascending order by root key (part 1 of 2)

01 //PFP EXEC PGM=PFPPMAIN,REGION=0M
02 //STEPLIB DD DISP=SHR,DSN=BMC.PFP.LOAD
Sort file in ascending order by RAP and sort logical SDEPs in reverse sequence

In this scenario, the file named PFPSAMP is sorted in ascending sequence by root anchor point (RAP). Two selected areas of PFPSAMP are specified as input. Logical SDEPs are sorted in “last in, first out” order to provide a reverse chronological transaction register.

Table 126  Descriptive text for JCL to sort files in ascending order by root key

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-03</td>
<td>EXEC and STEPLIB DD statements for Fast Path/EP offline execution.</td>
</tr>
<tr>
<td>04</td>
<td>The IMSACB DD is required for the File Sort Utility in order to correctly write the new unload file.</td>
</tr>
<tr>
<td>05-07</td>
<td>Defines the sort work files that your installation’s sort/merge utility uses to sort the records.</td>
</tr>
<tr>
<td>08</td>
<td>Specifies the input unload data set that contains the records that are to be processed by the PFPSORT command. Data for all three areas is contained in the unload file.</td>
</tr>
<tr>
<td>09-11</td>
<td>Defines the new output unload data set that contains the sorted records.</td>
</tr>
<tr>
<td>12</td>
<td>PFPSYSIN DD for Fast Path/EP control statements.</td>
</tr>
<tr>
<td>13</td>
<td>Specifies that the file using the DBD named PFPSAMP is to be sorted. The data in all three areas of PFPSAMP is sorted and written out.</td>
</tr>
<tr>
<td>14</td>
<td>Specifies that the file is to be sorted in ascending sequence using root key values.</td>
</tr>
<tr>
<td>15</td>
<td>The specified parameter for the SORT_OPTION keyword will be passed to the Sort utility used at your installation. In this example, the SORT utility will use the estimated input file size.</td>
</tr>
</tbody>
</table>

Figure 76  JCL to sort files in ascending order by root key (part 2 of 2)

```plaintext
03 //         DD DISP=SHR,DSN=IMSVS.RESLIB
04 //IMSACB DD DISP=SHR,DSN=IMSVS.ACBLIB
05 //SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,75)
06 //SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,75)
07 //SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,75)
08 //SORTIN DD DSN=BMC.PFP.PFPSAMP.UNLOAD,DISP=SHR
09 //SORTOUT DD DSN=BMC.PFP.PFPSAMP.UNLOAD.NEW,
10 // DISP=(NEW,CATLG),UNIT=SYSDA,
11 // SPACE=(CYL,(500,250))
12 //PFPSYSIN DD *
13 PFPSORT DBD=PFPSAMP,OAREA=ALL,
14 SORT_SEQUENCE=KEY_ASCEND,
15 SORT_OPTION='FILSZ=E2000000'.
/*
**Figure 77** JCL to sort files in ascending order by RAP and sort logical SDEPs in reverse sequence

```
01 //PFP  EXEC  PGM=PFPMAIN,REGION=0M
02 //STEPLIB  DD  DISP=SHR,DSN=BMC.PFP.LOAD
03 // D D D D S P H R S I A N S = I M S V S . R E S L I B
08 // D D D I S P = ( N E W , C A T L G ) , U N I T = S Y S D A ,
09 // D D S P A C E = ( C Y L . ( 5 0 0 , 2 5 0 ) )
10 //PFPSYSIN DD *
11 PFPSORT DBD=PFPSAMP,OAREA=(PFPSAMP1,PFPSAMP2)
12 SORT_SEQUENCE=(RAP_ASCEND,SDEPSEQ=LIFO),
13 SORT_OPTION='DYNALLOC',
/*
```

**Table 127** Descriptive text for JCL to sort files in ascending order by RAP and sort logical SDEPs in reverse sequence

<table>
<thead>
<tr>
<th>Line no.</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-03</td>
<td>EXEC and STEPLIB DD statements for Fast Path/EP offline execution.</td>
</tr>
<tr>
<td>04</td>
<td>The IMSACB DD is required for the File Sort Utility in order to correctly write the new unload file.</td>
</tr>
<tr>
<td>05-06</td>
<td>Specifies the concatenated input unload data sets that contain the records that are to be processed by the PFPSORT command for areas PFPSAMP1 and PFPSAMP2.</td>
</tr>
<tr>
<td>07-09</td>
<td>Defines the new output unload data set to contain the sorted records</td>
</tr>
<tr>
<td>10</td>
<td>PFPSYSIN DD for Fast Path/EP control statements.</td>
</tr>
<tr>
<td>11</td>
<td>Specifies that the file using the DBD named PFPSAMP is to be sorted. The OAREA keyword specifies that the output unload file is to contain data for areas PFPSAMP1 and PFPSAMP2.</td>
</tr>
<tr>
<td>12</td>
<td>Specifies that the file is to be sorted in ascending sequence using RAP values. The last SDEP retrieved by IMS using GN processing will be the first logical SDEP that was inserted by the unload utility (the SDEPs will be sorted in reverse chronological order).</td>
</tr>
<tr>
<td>13</td>
<td>The specified parameter will be passed to the Sort utility used at your installation. In this example, the sort utility will use dynamically allocated sort work data sets so they are not hardcoded in the JCL.</td>
</tr>
</tbody>
</table>
Sort file in ascending order by RAP and sort logical SDEPs in reverse sequence
Index

A

abbreviating command names 54
abbreviating keyword names 55
ACBLIB DD statement 507, 521
ACCESS 85, 119
ACTIVITY_FILECTL 73, 87
ACTUATE 89
ADD 81, 82, 90
ADDN 91
ADSN 92
aliases
  AREA 172, 298
  DATE 299
  DBD 300
  DBDNAME 118
  DEFAULT 264
  DFP 132
  ERT 102, 141
  EXTDD 170
  EXTRACT_DDNAME 170
  FCT 220
  FDI 161
  FPA 236
  FRD 162
  FRI 163
  FSA 160
  GROUP 301
  HISDD 170
  IDM 188
  IFP 191
  ILC 190
  INTERVAL 284
  IOVF 148
  ISA 194
  ITHREADS 189
  IUP 195
  LCT 203
  LEVEL 205
  LIMIT 206, 285, 302
  LRC, KEYS 198
  MDLDD 212
  MSGLIM 206
  MSGLV L 205
  MSGSUP 209
  ODM 228
  OSM 223

ORTHREADS 229
PA 234
PREFIX 296
PTR 234, 236
RAREA 273
RECOIA 248
RECIOM 249
RFP 244
RIP 250
RLA 251
RLI 252
ROP 245
RPLA 253
RPRA 254
RPTDD 263
RPTH 265
RPTLC 266
RTIOA 282
RTHOM 283
SCA 326
SCI 327
SCL 328
SCM 329
SCP 330
SDEP 151
SIA 293
SLA 294
SORTSEQ 316
SPLA 295
SSPTR 325
SUFFIX 297
UDA 337
UDP 338
UIA 339
UIM 340
UIP 340
XIOVF 149
XSDEP 152
XTHREADS 184
ALLOCATE 94
AMSOUT DD statement 33
ANALYZE 60, 62, 81, 95, 169
AREA 172, 298
Area Change Modeling Utility
capabilities 443
function 442
input requirements 445
randomizer considerations 446
sample scenarios 465
area number 482
AREA_KEY 96
areaname DD statement 33

B
BACKOUT 73, 75, 99
BMC Software, contacting 2
BREAK 99
BUILD 60, 63, 100
building JCL 56
built-in variable 363
BYPASS_RECORD 102

C
caching,input 175
caching,output 218
CASE 57, 58, 103, 221
CHANGE 60, 63, 103
changes to product 27
character display 57
character string typed literal 361
CHECKPOINT 105
CI level locking 476
command set examples
  how to interpret examples 628
  list of tasks 628
commands 59
  See also aliases
  ANALYZE 60, 62, 81, 95
  BACKOUT 73, 75, 99
  BUILD 60, 63, 100
  CHANGE 60, 63, 103
diagnostic 83
DISPLAY 83, 129
DMAC_CLEANUP 60, 64, 130
DMAC_PRINT 60, 64, 131
DUMP 83, 136
END 60, 139
EXTEND 60, 65, 146
EXTRACT 60, 65, 152
GLOBAL 59, 165
HELP 83, 167
IMAGECOPY 60, 66, 82, 181
INITIALIZE 60, 66, 185
MODIFY 83
operator interface 83
OPTIONS 58, 220
PREPARE 73, 238
PROCESS_AREA 239, 443, 468
PROCESS_EPR 240
RELOAD 61, 69, 258
REORGANIZE 61, 70, 259
RESTART 73, 79
RESTRUCTURE 73, 79, 275
RESYNC 61, 71, 276
RETRIEVE 61, 71, 279
SET 83, 304
SHADOW_INIT 73, 79, 305
SHUTDOWN 84, 310
STATUS 73, 79, 320
UNLOAD 61, 72, 335
VERIFY 61, 72, 342
XSCAN 61, 72, 353
COMMIT function 473
COMPRESS 106, 144
COMPRESS 107
concurrent processing 188
concurrent tasks, limiting 58
CONFIGURE_AREA 108
CONFIGURE_IOVF 109
CONFIGURE_RAA 110
CONFIGURE_SDEP 111
Control Interval Dump and Modification Utility 465
  available functions 473
  capabilities 468
  control statements 472
  DBRC considerations 470
  function 443
  keywords and subcommands 468
  multiple area data sets (MADS) considerations 471
donkey offline considerations 470
  online considerations 476
  process an entire DEDB 470
  process specific areas 470
data conversions 371
data-type 363
DATE 299
DATE_TIME_FORMAT 57, 115, 221
DBD 116, 300
DBD_KEY 118
DBDNAME 118
DBFHANDLE 446
DBFUMSC0 489
DBRC 119
DD statements
ACBLIB 507, 521
AMSOUT 33
areaname 33
DURDzzzO 509
DURSzzzO 509
IAREAxxx 35
IARxxxxx 35
IMSACB 36, 37, 471
IMSACBA 36, 37, 46
IMSACBB 36, 37, 46
IMSRESLB 37
indexname 38
NEWACB 39
NEWPPXLB 39
NEWRESLB 40
OAREAxxx 40
OARxxxxx 40
OLCSTAT 41, 480, 484, 508, 521
OLDACB 42
OLDLIB 42
OLDRESLB 42
PFPPDLTS 43
PPPEPR 44
PFPOPTS 44
PFPPRINT 44
PFPPRTS 45
PFPSYSIN 45
PFPTOTAL 45
PFXLIB 46
RECON2 46
RECON3 46
RECONn 46
STEPLIB 46, 508, 522
UR6fCTL 509
UR6RPT 508, 509
UR7CTL 522
UR7DATA 522
UR7RPT 522
XDyyyyyO 509

DDNAME 120, 134
DEDDB modeling and prototyping 443
DEFAULT 264
definition for SORT=UR, table 538, 539
DEFLTjcl 121
DELETE 81, 82, 83, 122
DESC 57, 123, 221
DETAIL 124
DETAIL_LEVEL 125
DFP 132
diagnostic commands 83
DISCARD_FILECTL 64, 126
DISP 127, 134
DISPLAY 83, 129
DMAC block contents 465, 468
DMAC_CLEANUP 60, 64, 130
DMAC_PRINT 60, 64, 131
DMAC_PRINT function 473
DMAC_PRINT utility
capabilities 465
function 442
DMCB block 468
DMCB_PRINT function 473
DOVF_FREESPACE_PERCENT 132
DSNAME 92, 114, 128, 133
DUMP 83, 136
DUMP_TYPE 136
DURDzzzO DD statement 509
DURSzzzO DD statement 509

E
EARLY_TERMINATION 137
EDIT 139, 140
electronic documentation 23
END 60, 139
entry code convention 527
error queue elements (EQEs) 471
ERROR_THRESHOLD 140
ERT 102, 141
EXCEPTION_LIMIT 141
EXCLUDE 60, 65, 82, 142
EXCLUDE subcommand 142
EXEC statement 30, 31
EXPAND 106, 143
EXPDT 134, 145
eexpression language
available functions 356, 358
data conversions 371
operands 360
operators 373
sample scenarios 631
expressions 356
EXTDD 170
EXTEND 60, 65, 146
EXTEND_IOVF 147
EXTEND_IOVF_#UOWS 149, 150, 151
EXTEND_SDEP 150
EXTEND_SDEP_#CIS 148, 149, 151
extending IOVF 147, 149
extending SDEP 150, 151
EXTRACT 60, 65
EXTRACT command 152
EXTRACT_DDNAME 168, 170

F
FABEUR6 502
FABEUR6 JCL requirements 507
FABEUR6 program extension 502
FABEUR6 utility control statements 509
FABEUR7 502
FABEUR7 JCL requirements
UR7CTL DD 522
UR7DATA1 DD/UR7DATA2 DD 522
FABEUR7 program extension 514
FABEUR7 utility control statements 522
for HDUNLOAD input 522
for non-HDUNLOAD input 523
FCT 220
FDI 161
field variable 362
field-name 363
FIELDS 154, 356
File Sort Utility
available keywords 478
FLOWER_BOX 155
FORCE 156
FORMAT 157
formats, syntax diagrams 26
FP_PROCNAME 221
FPA 236
FRAGMENTATION_PERCENT 158
FRD 162
FREESPACE_ANALYSIS 159, 382
FREESPACE_DOVF_IOVF 160
FREESPACE_RAA_DOVF 161
FREESPACE_RAA_IOVF 162
FRI 163
FSA 160
full validation 290
FULLSEG 163
functions 365, 473
COMMUT 473, 475
DMAC_PRINT 473, 475
DMCB_PRINT 473, 475
IS_PAPPED 366
IS_ZONED 366
LENGTH 367
REP 473, 475
ROLLBACK 473, 476
SEGMENT_CODE 367
SEGMENT_LEVEL 368
SEGMENT_NAME 368
SEGMENT_PARENT 368
SNAP 473, 474
SYSDATE 368
SYSDATETIME 368
SYSTIME 369
VER 473, 475
G
GENMAX 164
GLOBAL 165, 169
GLOBAL command 59
GROUP 301
GROUP_KEY 167
GSNAME 164, 333
H
HADOW2_DSNNAME 345
HADOW2_SUFFIX 345
HELP 83, 167
Help, online 23
hexadecimal typed literal 362
HISDD 170
I
IAREA 117, 170, 215
IAREA keyword 466, 469, 478
IAREAxxx DD statement 35
IARxxxxx DD statement 35
IBM HD Reorganization Unload Utility 510
IC 60, 66, 172
ICACHE 174, 218, 223
ICJCL 175
ID 129, 176
IDCAMS_OPTION 177
IDM 188
IFP 191
IFP_ACCOUNT 57, 178, 221
IFP_JOBNAME 57, 178, 221
IFP_LIMIT 57, 179, 221
IFP_PROCNAME 58, 180
ILC 190
IMAGECOPY 60, 66, 82, 181
IMSACB DD statement 36, 37, 471
IMSACBA DD statement 36, 37, 46
IMSACBB DD statement 36, 37, 46
IMSRESLB DD statement 37
INCLUDE 66, 182
increasing IOVF 147, 149
increasing SDEP 150, 151
INDEX 183
INDEX_THREADS 183
indexname DD statement 38
INITIALIZE 60, 66, 185
INPUT_DSN_MASK 185
INPUT_DSN_MASK keyword 469, 478
INPUT_THREADS 188
INSERT_LIMIT_COUNT 190
INTERVAL 284
IOVF 148
IOVF, extending 147, 149
IOVF_FREESPACE_PERCENT 190
IOVF_LOAD_HWM 191
IOVF_SAVE_THRESHOLD 192
IOVF_SPACE_ANALYSIS 193, 382
IOVF_USED_PERCENT 194
ISA 194
ITHREADS 189
IX 60, 67, 195

J
JCL 29
See also DD statements
See also EXEC statement
JCL, building 56

K
keywords 61, 74, 382
See also aliases
ACCESS 85, 119
ACTUATE 89
ADDN 91
ADSN 92
AREA_KEY 96
BREAK 99
BYPASS_RECORD 102
CASE 57, 58, 103, 221
CHECKPOINT 105
COMPRESS 106, 144
COMPRESSION 107
CONFIGURE_AREA 108
CONFIGURE_IOVF 109
CONFIGURE_RAA 110
CONFIGURE_SDEP 111
DATACLAS 113, 134
DATACLASS 114, 307
DATE_TIME_FORMAT 57, 115, 221
DBD 116, 469, 478
DBD_KEY 118
DBRC 119
DDNAME 120, 134
DEFLTJCL 121
DESC 123, 221
DETAIL 124
DETAIL_LEVEL 125
DISP 127, 134
DOVF_FREESPACE_PERCENT 132
DSNAME 92, 114, 128, 133
DUMP_TYPE 136
EARLY_TERMINATION 137
EDIT 139, 140
ERROR_THRESHOLD 140
EXCEPTION_LIMIT 141
EXPAND 106, 143
EXPDT 134, 145
EXTEND_IOVF 147
EXTEND_IOVF#UOWS 149
EXTEND_IOVF_#UOWS 149, 150, 151
EXTEND_SDEP 150
EXTEND_SDEP#CIS 151
EXTEND_SDEP_#CIS 148, 149, 151
EXTEND_SDEP#CIS 148, 149, 151
EXTRACT_DDNAME 168
FIELDS 154
FLOWER_BOX 155
FORCE 156
FORMAT 157
FRAGMENTATION_PERCENT 158
FREESPASGE_ANALYSIS 159, 382
FREESPASGE_DOVF_IOVF 160
FREESPASGE_RAA_DOVF 161
FREESPASGE_RAA_IOVF 162
FULLSEG 163
GENMAX 164
GROUP_KEY 167
GSGNAME 164, 333
IAREA 117, 170, 215, 466, 469, 478
ICACHE 174, 218, 223
ICJCL 175
ID 129, 176
IDCAMS_OPTION 177
IFP_ACCOUNT 57, 178, 221
IFP_JOBNAME 57, 178, 221
IFP_LIMIT 57, 179, 221
IFP_PROCNAME 58, 180, 221
INDEX 183
INDEX_THREADS 183
INPUT_DSN_MASK 185, 469, 478
INPUT_THREADS 188
INSERT_LIMIT_COUNT 190
IOVF_FREESPASGE_PERCENT 190
IOVF_LOAD_HWM 191
IOVF_SAVE_THRESHOLD 192
IOVF_SPACE_ANALYSIS 193, 382
IOVF_USED_PERCENT 194
LANGUAGE 58, 196, 221
LARGEST_DATABASE_RECORDS 197
LIKE 134, 198
LINE_COUNT 58, 200, 221
LIST_OPTIONS 58, 201, 221
LOCATION 203
MESSAGE_LEVEL 204
MESSAGE_LIMIT 205
MESSAGE_NUMBER 206
MESSAGE_OVERRIDE 207
MESSAGE_SUPPRESSION 208
MGMTCLAS 134, 209
MGMTCLAS2 210, 307
MODEL_DDNAME 211
MONITOR 58, 213, 221
NOTIFY 212, 214
OAREA 91, 117, 221
OCACHE 174, 218, 223
ORPHANED_SDEP_MSG 222
OUTAGE_WINDOW 223, 304
OUTPUT_DSN_MASK 478
OUTPUT_DSNAME 226
OUTPUT_THREADS 229
PAGING_COUNT 221
POINTER_ANALYSIS 234, 382
POINTER_VALIDATION 235
PREOPEN 237, 346
PRODUCT 241
PRODUCT_LIMIT 58, 221, 243
RAA_FREESPACE_PERCENT 244
RAP_OVERFLOW_PERCENT 245
RAP_VALIDATION 246
RECORD_IO_AVERAGE 248
RECORD_IO_MAXIMUM 249
RECORD_IOVVF_PERCENT 250
RECORD_LENGTH_ANALYSIS 251, 382
RECORD_LENGTH_INCREMENT 252
RECORD_PLACEMENT_ANALYSIS 252, 382
RECORD_PROFILE_ANALYSIS 253, 382
RECOVJCL 254
RECVJCL 256
RELOAD 258
REPORT_DDNAME 262
REPORT_DEFAULT 263
REPORT_HEADING 264
REPORT_LINE_COUNT 265
REPOSITORY_DSNAME 58, 221, 266
REPOSITORY_GROUP 58, 221, 267
REPOSITORY_OVERWRITE 58, 221, 268
REPOSITORY_RETENTION_COUNT 221, 269
REPOSITORY_RETENTION_PERIOD 58, 221, 270
REQUIRE_AREA 117, 271
RESTART 274
RETAIENED_SUFFIX 277
RETPD 134, 276, 278
REUSE 280
ROOT_IO_AVERAGE 281
ROOT_IO_MAXIMUM 282
ROUTCDE 58, 123, 221, 283
SAMPLE_INTERVAL 143, 284
SAMPLE_LIMIT 143, 285
SCAN 286
SCRIPT 287, 469, 478
SDEP_PROCESS 287
SDEP_VALIDATION 290
SEGMENT 143, 291
SEGMENT_IO_ANALYSIS 292, 382
SEGMENT_LENGTH_ANALYSIS 293, 382
SEGMENT_PLACEMENT_ANALYSIS 294
SEGMENT_RECORD_PREFIX 295
SEGMENT_RECORD_SUFFIX 296
SELECT_AREA 297
SELECT_DATE 299
SELECT_DBD 300
SELECT_GROUP 301
SELECT_DBD 302
SELECT_UOW 302
SHADOW_SUFFIX 306
SHADOW2_DSNAME 115, 210, 307, 345
SHADOW2_SUFFIX 115, 210, 309, 345
SORT 311
SORT_NAME 58, 221, 312
SORT_OPTION 313, 478
SORT_SEQUENCE 315
SPACE 98, 134, 316
STACK_NAME 318
STARTUOW 319
STOPUOW 321
STORCLAS 134, 322, 323
STORCLAS2 307, 323
SYNONYM_CHAIN_ANALYSIS 325, 382
SYNONYM_CHAIN_INCREMENT 326
SYNONYM_CHAIN_LENGTH 327
SYNONYM_CHAIN_MAXIMUM 328
SYNONYM_CHAIN_PERCENT 329
TIMESTAMP 58, 221, 332
TRACK 332
TYPE_RUN 333
UNIT 134, 334
UOW_DETAILED_ANALYSIS 336, 382
UOW_DOVF_PERCENT 337
UOW_IOVF_AVERAGE 338
UOW_IOVF_MAXIMUM 339
UOW_IOVF_PERCENT 340
VOLCNT 134, 343, 345
VOLSER 134, 344
VOLSER2 307, 345
VSO 237, 346
WARNING 58, 221, 348
WHERE 143, 349
WORK_DATASET 58, 221, 349

L

LANGUAGE 58, 196, 221
language code 58
LARGEST_DATABASE_RECORDS 197
LCT 203
LEVEL 205
LIKE 134, 198
LIMIT 206, 285, 302
limit concurrent tasks 58
LINE_COUNT 58, 200, 221
lines per page 58
LIST 81, 82, 83, 200
list option settings 58
LIST_OPTIONS 58, 201, 221
literals
  string 361
  typed 361
    character string typed literal 361
    hexadecimal typed literal 362
    numeric typed literal 362
LOADCTL 60, 67, 81, 202
LOCATION 203

M
mask 89, 133, 134, 185, 187, 198, 226, 227, 298, 300, 301, 307, 370
MDLDD 212
message routing 58
MESSAGE_LEVEL 204
MESSAGE_LIMIT 205
MESSAGE_NUMBER 206
MESSAGE_OVERRIDE 207
MESSAGE_SUPPRESSION 208
MGMTCLAS 134, 209
MGMTCLAS2 210, 307
mode availability, command 60, 73
MODEL_DDNAME 211
MODIFY 81, 212
MODIFY operator command 83
MONITOR 58, 213, 221
MSGLIM 206
MSGLVL 205
MSGSUP 209

N
NEWACB DD statement 39
NEWPFXLB DD statement 39
NEWRESLB DD statement 40
NOTIFY 212, 214
numeric typed literal 362

O
OAREA 91, 117, 215
OAREAxxx DD statement 40
OARxxxx DD statement 40
OBJECT 129, 217
OCACHE 174, 218, 223
ODM 228
OFILECTL 219
OFILECTL subcommand 60, 61, 68, 82
OLCSTAT DD statement 41, 480, 484, 508, 521
OLDACB DD statement 42
OLDLIB DD statement 42
OLDRESLB DD statement 42
Online DEDB Data Extract Record Layout 537, 547
online Help 23
operator interface commands 83
operators 373
assignment 375
Boolean 376
comparison 376
conditional 379
numeric 377
precedence of operators 374
OPTIONS 220
OPTIONS command 58
ORPHANED_SDEP_MSG 222
OSM 223
OUTHREADS 229
OUTAGE_WINDOW 223, 304
OUTPUT 60, 68, 75, 81, 82, 225
OUTPUT_DSN_MASK keyword 478
OUTPUT_DSNNAME 226
OUTPUT_THREADS 229
OVERRIDE 230

P
PA 234
PAGING_COUNT 221
parallel processing 188
PERFORM 68, 231
PERFORM subcommand 468
PFMD0100 446
PFMD0500 460
PPFDFTLS DD statement 43
PFPEPR DD statement 44
PFPOPTS DD statement 44
PFPRINT DD statement 44
PFPRPTS DD statement 45
PFPSYSIN DD statement 45
PFPTOTAL DD statement 45
PFSD0700 489
PFUT0B50 443, 482
PFXLIB DD statement 46
physical sequence data 482
PLAN_FILECTL 73, 232
POINTER_ANALYSIS 234, 382
POINTER_VALIDATION 235
precision 373
PREFIX 296
PREOPEN 237, 346
PREPARE 73, 238
PROCESS_AREA 239
PROCESS_AREA command 443, 468
PROCESS_EPR 240
PRODUCT 241
product changes 27
product support 3
PRODUCT_LIMIT 58, 221, 243
PTR 234, 236
publications, related 23, 24

Q
quick validation 290
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

R

RAA_FREESPACE_PERCENT 244
Randomizer Interface Routine 443
Randomizer Interface Subroutine capabilities 482
  program example 487
randomizers 446
randomizing routines, testing 463
RAP 246
RAP RBA values 482
RAP_OVERFLOW_PERCENT 245
RAP_VALIDATION 246
RAREA 248
RECOIA 248
RECOIOM 249
RECON2 DD statement 46
RECON3 DD statement 46
RECONn DD statement 46
RECORD_IO_AVERAGE 248
RECORD_IO_MAXIMUM 249
RECORD_IOVFSPERCENT 249
RECORD_LENGTH_ANALYSIS 251, 382
RECORD_LENGTH_INCREMENT 252
RECORD_PLACEMENT_ANALYSIS 252, 382
RECORD_PROFILE_ANALYSIS 253, 382
RECOVJCL 254
RECOVPD 255
RECJCL 256
REGISTER 73, 257
registers on entry, convention of 526
related publications 23, 24
RELOAD 61, 69, 258
REORGANIZE 61, 70, 259
REP function 473
REPORT 60, 70, 71, 78, 82, 260
report field descriptions
  Area Summary 383, 436
  Free Space Analysis 386
  IOVF Space Analysis 393
  Pointer Analysis 396
  Record Length Analysis 404
  Record Placement Analysis 408
  Record Profile Analysis 410
  Reload Input Area Summary 427
  Reload Output Area Summary 430
  Reload Segment Summary 434
  Segment I/O Analysis 398
  Segment Length Analysis 401
  Segment Placement Analysis 403
  Synonym Chain Analysis 413
  Unload Input Area Summary 417
  Unload Output Area Summary 421
  Unload Segment Summary 425
  UOW Detailed Analysis 394
report generation keywords
  FREESPACE_ANALYSIS 159, 382

IOVF_SPACE_ANALYSIS 193, 382
POINTER_ANALYSIS 382
RECORD_LENGTH_ANALYSIS 251, 382
RECORD_PLACEMENT_ANALYSIS 252, 382
RECORD_PROFILE_ANALYSIS 253, 382
SEGMENT_IO_ANALYSIS 292, 382
SEGMENT_LENGTH_ANALYSIS 293, 382
SEGMENT_PLACEMENT_ANALYSIS 294, 382
SYNONYM_CHAIN_ANALYSIS 325, 382
UOW_DETAILED_ANALYSIS 336, 382

report lines per page 58
REPORT_DDNAME 262
REPORT_DEFAULT 263
REPORT_HEADING 264
REPORT_LINE_COUNT 265
repository allocation 58
repository data set name 58
repository group 58
repository retention 58
REPOSITORY_DSNAME 58, 123, 221, 283
REPOSITORY_GROUP 58, 123, 221, 283
REPOSITORY_OVERWRITE 58, 221, 268
REPOSITORY_RETENTION_COUNT 221, 269
REPOSITORY_RETENTION_PERIOD 58, 221, 270
REQUIRE_AREA 117, 271
RESTART 273
RESTRUCTURE 73, 79, 275
RESYNC 61, 71, 276
RETAINED_SUFFIX 277
RETPD 134, 276, 278
RETRIEVE 61, 71, 169, 261, 279
REUSE 280
RFP 244
RIP 250
RLA 251
RLI 252
ROLLBACK function 473
Root Anchor Point 246
ROOT_IO_AVERAGE 281
ROOT_IO_MAXIMUM 282
ROP 245
ROUTCDE 58, 123, 221, 283
RPLA 253
RPRA 254
RPTDD 263
RPTH 265
RPTLC 266
RTIOA 282
RTIOM 283

S

sample command scenarios
  how to interpret samples 628
  list of tasks 628
SAMPLE_INTERVAL 143, 284
SAMPLE_LIMIT 143, 285
SCA 326
SCAN 286
SCANCOPY DD statement 489
SCI 327
SCL 328
SCM 329
SCP 330
SCRIPT 287
SCRIPT keyword 469, 478
SDEP 151
  extending 150, 151
  space utilization 490
  validation 290
SDEP Space Utilization Utility
  capabilities 489
  function 443
  program module descriptions 489
SDEP_PROCESS 287
SDEP_VALIDATION 290
SEGMENT 143, 291
SEGMENT_IO_ANALYSIS 292, 382
SEGMENT_LENGTH_ANALYSIS 293, 382
SEGMENT_PLACEMENT_ANALYSIS 294, 382
SEGMENT_RECORD_PREFIX 295, 356
SEGMENT_RECORD_SUFFIX 296, 356
SELECT_AREA 297
SELECT_DATE 299
SELECT_DBD 300
SELECT_GROUP 301
SELECT_LIMIT 302
SELECT_UOW 302
SET 83, 304
SHADOW_INIT 73, 79, 305
SHADOW_SUFFIX 306
SHADOW2_DSNAME 115, 210, 307, 324
SHADOW2_SUFFIX 115, 210, 309, 324
SHUTDOWN 84, 310
SIA 293
SLA 294
SNAP function 473
SORT 311
SORT_NAME 58, 221, 312
SORT_OPTION 313
SORT_OPTION keyword 478
SORTSEQ 316
SPACE 98, 134, 316
space utilization graph 490
space utilization reports 490
SPLA 295
SSPTR 325
STACK_NAME 318
STARTUOW 319
STATUS 73, 79, 320
STEPLIB DD statement 46, 508, 522
STOPUOW 321
STORCLAS 134, 322, 323
STORCLAS2 307, 323
string literal 361
subcommands 60, 73
See also aliases
  ACTIVITY_FILECTL 73, 87
  ADD 81, 82, 90
  ALLOCATE 94
  CORRECTIONS_FILECTL 60, 112
  DELETE 81, 82, 83, 122
  DISCARD_FILECTL 64, 126
  EXCLUDE 60, 65, 82, 142
  IC 60, 66, 172
  INCLUDE 66, 182
  IX 60, 67, 195
  LIST 81, 82, 83, 200
  LOADCTL 60, 67, 81, 202
  MODIFY 81, 212
  OFILECTL 60, 61, 68, 82, 219
  OUTPUT 60, 68, 75, 81, 82, 225
  OVERRIDE 230
  PERFORM 68, 231, 468
  PLAN_FILECTL 73, 232
  REGISTER 73, 257
  REPORT 60, 70, 71, 78, 82, 260
  RESET 273
  THRESHOLD 60, 71, 80, 330
  USER_RECORD 60, 72, 341
SUFFIX 297
summary of changes 27
support, customer 3
SYNONYM_CHAIN_ANALYSIS 325, 382
SYNONYM_CHAIN_INCREMENT 326
SYNONYM_CHAIN_LENGTH 327
SYNONYM_CHAIN_MAXIMUM 328
SYNONYM_CHAIN_PERCENT 329
syntax diagrams
  repository maintenance commands/subcommands 615
  syntax statement conventions 25
  syntax, format for diagrams 26
T
  technical support 3
  threading, input 188
  threading, output 229
  THRESHOLD 60, 71, 80, 330
  TIMESTAMP 58, 221, 332
  TRACK 332
truncating command names 54
truncating keyword names 55
TYPE_RUN 333
typed literal 361
U

UDA 337
UDP 338
UIA 339
UIM 340
UIP 340
underlining, syntax 54, 55
UNIT 134, 334
UNLOAD 61, 72, 335
UNLOAD/RELOAD
  convention of registers on entry 526
  entry code convention 527
UOW_DETAILED_ANALYSIS 336, 382
UOW_DOVF_PERCENT 337
UOW_IOVF_AVERAGE 338
UOW_IOVF_MAXIMUM 339
UOW_IOVF_PERCENT 340
UR6FCTL DD statement 509
UR6RPT DD statement 508, 509
UR7CTL DD statement 522
UR7DATA DD statement 522
UR7RPT DD statement 522
USER_RECORD 60, 72, 341
utilities
  Area Change Modeling 442, 443
  Control Interval Dump and Modification 443, 468
  DMAC_PRINT 442, 465
  Randomizer Interface Routine 443
  SDEP Space Utilization 443
utility control statements
  FABEUR6 509
  FABEUR7 522

V

validation 246, 290
validation of Root Anchor Points 246
VER function 473
VERIFY 61, 72, 342
VOLCNT 134, 343, 345
VOLSER 134, 344
VOLSER2 307, 345
VSO 237, 346

W

WARNING 58, 221, 348
warning effects 58
WHERE 143, 349, 356
WORK_DATASET 58, 221, 349
WTO message descriptor code 57
WTO message routing codes 58