CA IDMS - 19.0

Reporting

Date: 16-Jan-2018
This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2017 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.
# Table of Contents

ASF Row-Level Security Reports -- IREPORTS .............................................. 28
Syntax ................................................................................................................. 28
Parameters .......................................................................................................... 28
Summary Table ................................................................................................... 29
Example .............................................................................................................. 29
ASF Row-Level Security Reports ........................................................................ 29

CA ADS Reports -- AREPORTS ................................................................. 32
Producing CA ADS Reports ................................................................................ 32
Syntax .................................................................................................................. 33
Parameters .......................................................................................................... 33
Examples ............................................................................................................. 33
AREPORT 001 and 002 - ADS Dialogs and Their Components ...................... 34
Contents ............................................................................................................. 34
Field Descriptions .............................................................................................. 35
AREPORT 003 - ADS Dialogs by Process Key .................................................. 37
Contents ............................................................................................................. 37
Field Descriptions .............................................................................................. 38
AREPORT 004 - ADS Dialogs by Record Key .................................................... 39
Contents ............................................................................................................. 39
Field Descriptions .............................................................................................. 40
AREPORT 005 - ADS Dialogs by Subschema Key ........................................... 41
Contents ............................................................................................................. 41
Field Descriptions .............................................................................................. 41
AREPORT 006 - ADS Dialogs by Map Key ....................................................... 42
Contents ............................................................................................................. 42
Field Descriptions .............................................................................................. 42

CA ICMS Catalog Reports -- DREPORTS .................................................. 44
Summary of Catalog Reports .............................................................................. 44
Producing Catalog Reports ................................................................................ 45
Syntax .................................................................................................................. 45
Parameters .......................................................................................................... 46
Examples ............................................................................................................................................. 46
DREPORT 090 - Catalog Summary Report .......................................................................................... 47
  Field Descriptions .......................................................................................................................... 47
DREPORT 091 - Catalog Detail Report .............................................................................................. 48
  Field Descriptions .......................................................................................................................... 49
DREPORT 092 - Group Detail Report ................................................................................................. 50
  Field Descriptions .......................................................................................................................... 51
DREPORT 093 - User Detail Report .................................................................................................... 52
  Field Descriptions .......................................................................................................................... 52
DREPORT 094 - Folder Detail Report ................................................................................................. 54
  Field Descriptions .......................................................................................................................... 54
DREPORT 095 - Object Detail Report ............................................................................................... 55
  Field Descriptions .......................................................................................................................... 56
DREPORT 096 - Catalog Summary Key Report ............................................................................... 57
  Field Descriptions .......................................................................................................................... 57
DREPORT 097 - Catalog Detail Key Report ........................................................................................ 58
  Field Descriptions .......................................................................................................................... 58

CA IDMS/DB Journal Reports -- JREPORTS ....................................................................................... 61
Uses for Journal Reports .................................................................................................................. 62
Types of Journal Records ................................................................................................................ 62
Summary of Records Required for Journal Reports .......................................................................... 71
Producing Journal Reports ................................................................................................................ 72
  Uses Archive Journal File as Input ................................................................................................ 72
  Required CA Culprit for CA IDMS Modules ............................................................................... 72
Syntax ................................................................................................................................................ 72
Parameters ........................................................................................................................................ 73
Examples ........................................................................................................................................... 74
Operating System Considerations .................................................................................................... 75
JREPORT 001 - Transaction Summary ............................................................................................ 75
  Field Descriptions .......................................................................................................................... 76
JREPORT 002 - Program Termination Statistics .............................................................................. 77
  Field Descriptions .......................................................................................................................... 77
JREPORT 003 - Program I/O Statistics ............................................................................................. 79
  Field Descriptions .......................................................................................................................... 80
JREPORT 004 - Program Summary .................................................................................................. 81
  Field Descriptions .......................................................................................................................... 82
JREPORT 005 - Detail Area/Transaction .......................................................................................... 83
  Field Descriptions .......................................................................................................................... 84
JREPORT 006 - Detail Program/Area ............................................................... 85
  Field Descriptions ........................................................................... 86
JREPORT 007 - Area Summary ............................................................... 86
  Field Descriptions ........................................................................... 87
JREPORT 008 - Formatted Record Dump .............................................. 88
  Field Descriptions ........................................................................... 89
JREPORT 009 - User ID ......................................................................... 93
  Field Descriptions ........................................................................... 94
JREPORT 010 - External User Identity .................................................... 95
  Field Descriptions ........................................................................... 95
JREPORT 011 - Count By Journal Record Type Report ......................... 96
  Field Descriptions ........................................................................... 97
JREPORTS z/OS JCL ............................................................................. 97
  Local Mode ....................................................................................... 97
  Local Mode Statements and Data Sets .............................................. 99
  Central Version ............................................................................... 100
  Central Version Statements and Data Sets ...................................... 100

CA IDMS/DB SQL Dictionary Reports -- QREPORTS ................... 101
QREPORT 001 - SQL Column Name Report ......................................... 101
  Symbolic Parameter Overrides ......................................................... 102
  Job Submission .............................................................................. 102
  Field Descriptions ........................................................................ 102
QREPORT 002 - Table and Column Report ......................................... 103
  Symbolic Parameter Overrides ......................................................... 103
  Job Submission .............................................................................. 104
  Field Descriptions ........................................................................ 105
QREPORT 003 - Schema and Table Report ......................................... 109
  Symbolic Parameter Overrides ......................................................... 109
  Job Submission .............................................................................. 109
  Field Descriptions ........................................................................ 110
QREPORT 004 - SQL Access Module Information .......................... 112
  Symbolic Parameter Overrides ......................................................... 112
  Job Submission .............................................................................. 112
  Field Descriptions ........................................................................ 113
QREPORT 005 - SQL Table Access Report ......................................... 114
  Symbolic Parameter Overrides ......................................................... 114
  Job Submission .............................................................................. 114
  Field Descriptions ........................................................................ 115
QREPORT 006 - SQL Table Syntax Report ......................................... 116
Central Version Modifications ................................................................. 138
IDMSLBLS Procedure Listing ................................................................. 138

Compliance Reporting ........................................................................... 143

DB Analyzer System Output Reports ..................................................... 144
Individual Reports ................................................................................... 144
Comparative Reports ............................................................................... 144
Area Report ............................................................................................... 145
Area Report Field Descriptions ............................................................... 145
Distribution Section Fields ..................................................................... 151
Histogram Section Fields ......................................................................... 154
Analysis by Record Section Fields ......................................................... 156
Comparative Area Report .......................................................................... 157
Comparative Area Report Field Descriptions ........................................ 158
Distribution Section Fields ..................................................................... 159
Histogram Section Fields ......................................................................... 161
Analysis by Record Section Fields ......................................................... 163
Record Report ........................................................................................... 164
Record Report Field Descriptions ......................................................... 164
Distribution Section Fields ..................................................................... 168
Histogram Section Fields ......................................................................... 170
Comparative Record Report ...................................................................... 171
Comparative Record Report Field Descriptions ..................................... 171
Distribution Section Fields ..................................................................... 173
Histogram Section Fields ......................................................................... 174
Set Report .................................................................................................. 176
Set Report Field Descriptions ................................................................. 176
Distribution Section Fields ..................................................................... 181
Histogram Section Fields ......................................................................... 183
Multi-Member Set Analysis Section Fields ............................................ 184
Comparative Set Report ............................................................................ 185
Set Statistic Section Fields ..................................................................... 186
Distribution Section Fields ..................................................................... 188
Histogram Section Fields ......................................................................... 190
Multi-Member Set Analysis Section Fields ............................................ 191
SR8 Index Report ..................................................................................... 192
SR8 Index Report Fields ........................................................................... 192
Table Entry Distribution Section Fields ................................................ 195
DC/UCF Statistics Reports -- SREPORTS .................................................. 202
Uses for Statistics Reports ................................................................. 202
Other Tools Available ..................................................................... 203
Types of Statistics Records ............................................................. 203
  Layout of Statistics Log Records .................................................... 204
  Task or Transaction Statistics ....................................................... 204
    Order of DSECTs for Task and Transaction Statistics ................... 205
System Statistics ........................................................................... 205
    Order of DSECTs for System Statistics ....................................... 205
Histograms ..................................................................................... 206
    Order of DSECTs for Histograms .............................................. 206
Number of Statistics Log Records .................................................. 206
SREPORTs for Statistics Records .................................................... 207
  SREPORT 099 ............................................................................. 207
  SREPORT 000 ............................................................................. 207
Producing Statistics Reports ........................................................... 207
  Syntax ....................................................................................... 208
Parameters .................................................................................... 208
Examples ....................................................................................... 210
Operating System Considerations .................................................. 211
DC/UCF System Statistics Reports .................................................. 213
  SREPORT 003 - IDMS DC System Statistics .................................. 214
  SREPORT 012 - IDMS DC Task Summary .................................... 224
  SREPORT 013 - IDMS DC Program Summary ............................... 225
  SREPORT 014 - IDMS DC Queue Summary ................................. 227
  SREPORT 015 - IDMS DC Line Summary ..................................... 228
  SREPORT 016 - IDMS DC Physical Terminal Summary .................. 230
Task and External Request Unit Service (ERUS) Statistics Reports ........................................................................ 231
  Uses for Task and ERUS Statistics .............................................. 232
Statistics Collected Only Upon Request ........................................ 232
Summary of Task Statistics Reports .............................................. 232
Field Descriptions ........................................................................ 236
Transaction Statistics Reports ....................................................... 241
  Summary of Transaction Statistics Reports ................................. 241
DC/UCF System Reports -- CREPORTS ............................................. 256

Uses for DC/UCF System Reports ............................................... 256
Summary of CREPORTs .............................................................. 257
Producing DC/UCF System Reports ............................................. 259
  Syntax ................................................................................. 259
  Parameters .......................................................................... 259
  Examples ............................................................................. 260
CA ADS Parameter Reports (CREPORTs 040 and 045) ..................... 261
  Field Descriptions .............................................................. 261
Defined Devices Report (CREPORT 029) ......................................... 264
  Field Descriptions .............................................................. 264
Defined Messages Report (CREPORT 028) ....................................... 265
  Field Descriptions .............................................................. 265
Destination Reports (CREPORTs 007 and 024) ................................ 266
  Field Descriptions .............................................................. 266
Load Area Report (CREPORT 050) ............................................... 267
  Field Descriptions .............................................................. 267
Mapping Reports (CREPORTs 030 through 035) .............................. 268
  Field Descriptions .............................................................. 268
Nodes and Resource Table Reports (CREPORTS 043 and 044) .......... 271
  Field Descriptions .............................................................. 272
Module Text to Card Utility (CREPORT 051) ................................... 272
Module Text to File Utility (CREPORT 052) .................................... 273
Network Description Reports (CREPORTs 001-003, 014-018) .......... 274
  Contents ............................................................................. 274
  Sample Reports .................................................................. 275
  Field Descriptions .............................................................. 277
CA OLQ Reports (CREPORTs 041 and 046) ..................................... 280
  Field Descriptions .............................................................. 280
Program Description Reports (CREPORTs 004 and 019) ................. 282
CA IDMS Log Analyzer Parameters .................................................................................. 317
  Parameters and Their Uses .......................................................................................... 318
  COMMENTS Option .................................................................................................. 318
Billing Reports/Billing Record File Parameters ................................................................. 320
  REPORT Syntax ........................................................................................................ 320
  START Syntax .......................................................................................................... 320
  STOP Syntax ............................................................................................................. 321
  INTERVAL Syntax ..................................................................................................... 321
  LEVEL Syntax and Options ....................................................................................... 321
  FILE Syntax and Options .......................................................................................... 322
  RUTYPE Syntax and Options ..................................................................................... 322
  RUNAME Syntax and Options .................................................................................... 322
  NAME Syntax and Options ....................................................................................... 323
Program Report Parameters ............................................................................................. 324
  REPORT Syntax ........................................................................................................ 324
  START Syntax .......................................................................................................... 324
  STOP Syntax ............................................................................................................. 325
  INTERVAL Syntax ..................................................................................................... 325
  LEVEL Syntax and Options ....................................................................................... 325
  NAME Syntax and Options ....................................................................................... 326
Management Highlights Summary Report Parameters .................................................. 326
  REPORT Syntax ........................................................................................................ 326
  START Syntax .......................................................................................................... 327
  STOP Syntax ............................................................................................................. 327
  INTERVAL Syntax ..................................................................................................... 327
  LEVEL Syntax .......................................................................................................... 328
Buffer Pool Utilization Report Parameters ...................................................................... 328
  REPORT Syntax ........................................................................................................ 328
  START Syntax .......................................................................................................... 328
  STOP Syntax ............................................................................................................. 329
Management Ranking Report Parameters ....................................................................... 329
  REPORT Syntax ........................................................................................................ 330
  START Syntax .......................................................................................................... 330
  STOP Syntax ............................................................................................................. 330
  INTERVAL Syntax ..................................................................................................... 330
  RUTYPE Syntax and Options ..................................................................................... 331
  RANKWHAT Syntax and Options .............................................................................. 331
  RANKVALUE Syntax and Options ............................................................................ 332
  RANKHOW Syntax and Options .............................................................................. 332
  RANK Syntax ........................................................................................................... 333
Customizing the Billing Reports ....................................................................................... 333
Customizing the GSISVCX Module ............................................................................... 334
Customizing the USLBILX Module ................................................................. 335
Customizing the USLRPT5 Module ................................................................. 335
CA IDMS Log Analyzer Operations ................................................................. 335
  z/OS Environment ......................................................................................... 336
    z/OS JCL ....................................................................................................... 336
  Z/VSE Environment ......................................................................................... 337
    Z/VSE JCL ..................................................................................................... 338
  Z/VM Environment ........................................................................................ 339
    Z/VM EXEC .................................................................................................. 339
USLBILX and USLRPT5 Source Code ............................................................... 342
  Listing Modules for Z/OS ............................................................................. 342
  Listing Modules for Z/VSE ............................................................................ 342
External Request Element Extension ............................................................... 342
Billing the Record File ...................................................................................... 343
I/O Modules ...................................................................................................... 344

IDMS Performance Monitor System Reports ................................................. 345
Archiving -- z/OS ........................................................................................... 345
  Archiving Statistics From the DDLDCLOG Area .......................................... 346
    Archiving -- z/VM ....................................................................................... 351
Using SMF to Archive Statistics (z/OS only) .................................................. 352
Sample Job Streams For Running Reports .................................................... 356
  Running reports -- z/OS .............................................................................. 356
  Running reports -- z/VSE ........................................................................... 360
  Running reports -- z/VM ............................................................................. 361
Replacing the COPY Parameters (z/VSE only) ............................................... 362
Replacing the COPY Parameters For Tape Input (z/VSE only) ......................... 363
Note For DDR-Only Shops ............................................................................. 363
Interval Monitor Batch Reports ..................................................................... 364
  Selection criteria parameters ....................................................................... 365
    Syntax ......................................................................................................... 365
    Parameters ................................................................................................. 366
    Example ..................................................................................................... 369
  Request Reports (Interval Monitor) ............................................................... 369
    Syntax ......................................................................................................... 370
    Parameters ................................................................................................. 370
    Example ..................................................................................................... 371
  Report Samples ........................................................................................... 372
    PMIRPT01 Management summary report ................................................. 373
    PMIRPT02 Trend analysis report ............................................................... 374

---

Reporting 13
Application Monitor Batch Reports ................................................................. 407
Request Reports (Application Monitor) ............................................................. 408

Syntax ........................................................................................................... 408
Report selection parameters ........................................................................ 412
List of Report Samples ................................................................................... 415

PMARPT01 Task detail report ....................................................................... 416
PMARPT02 Task summary report ................................................................... 418
PMARPT03 CA ADS dialog detail report ....................................................... 420
PMARPT04 CA ADS dialog summary report ................................................. 421
PMARPT05 User detail report ....................................................................... 421
PMARPT06 User summary report .................................................................. 423
PMARPT07 Billing group detail report .......................................................... 425
PMARPT08 Billing group summary report ..................................................... 427
PMARPT09 Abnormal termination detail report ........................................... 428
PMARPT10 Abnormal termination summary report ....................................... 429
PMARPT11 LTERM detail report .................................................................. 429
How to Make a Report from a Table

Step 1 -- Select the type of table
Step 2 -- Select a table
Step 3 -- Choose columns and specify selection criteria
Step 4 -- Specifying additional selection criteria
Step 5 -- Retrieve the data
Step 6 -- Display your report

Creating Your Report From an SQL Table

Creating Your Report From an ASF Table

Online Query (OLQ) Reports

What are CA OLQ features? Menu facility CA OLQ command mode Two processing modes Two access modes Sign on and Sign off Getting Help Using Menu Facility Using PF Keys How to Make a Report from a Table Key Terms Creating Your Report From an SQL Table Creating Your Report From an ASF Table
Step 3 -- Choose columns and specify selection criteria .................................................. 467
Step 4 -- Specifying additional selection criteria ................................................................. 468
Step 5 -- Retrieve the data .................................................................................................. 470
Step 6 -- Display your report ............................................................................................... 471

How to Make a Report From More Than One Table ............................................................ 472
How to Combine Data From More Than One Table ............................................................. 473

Creating your report ............................................................................................................ 476
  Step 1 -- Select the type of table ...................................................................................... 477
  Step 2 -- Select tables ..................................................................................................... 477
  Step 3 -- Select, project, and join .................................................................................... 478
  Step 4 -- Retrieve the data ............................................................................................... 479
  Step 5 -- Display your report ........................................................................................... 479

How to Format Your Report ................................................................................................. 480
Format Report Key Terms ..................................................................................................... 481

Creating a Report ................................................................................................................... 482
  Step 1 -- Select a subschema ......................................................................................... 482
  Step 2 -- Select records ................................................................................................. 483
  Step 3 -- Choose columns .............................................................................................. 483
  Step 4 -- Retrieve the data ............................................................................................... 484
  Step 5 -- Display your report ........................................................................................... 485
  Step 1 - Create a Current Report .................................................................................... 485
  Step 2 - Modify Your Report Headers .......................................................................... 488
  Step 3 - Change Your Report Column Pictures ............................................................. 489
  Step 4 - Sort Your Report Rows and Specify Group Calculations .................................... 490

Sorting Your Report Rows .................................................................................................... 491
  Step 1 - Specify your sort criteria .................................................................................... 491
  Step 2 - Display your report ............................................................................................ 492
  Step 1 - Sort your report ................................................................................................. 492
  Step 2 - Display your report ............................................................................................ 493

Editing Your Report Values .................................................................................................. 493
  Step 1 - Specify your edit criteria .................................................................................... 495
  Step 2 - Display your report ............................................................................................ 496
  Additional editing features ............................................................................................... 496

Changing Your Column Relative Positions ......................................................................... 498

Changing Your Column Pictures .......................................................................................... 500
  Step 1 - Modify a column picture ................................................................................... 502
  Step 2 - Display your report ............................................................................................ 503

Changing Column Headers .................................................................................................. 503
  Step 1 - Change your column headers .......................................................................... 505
  Step 2 - Display your report ............................................................................................ 505

Making Page Headers and Footers ...................................................................................... 506
  Step 1 - Specify page headers and footers ...................................................................... 509
How to Save a Set of Commands as a Qfile                     510
How To Format Reports Containing Calculations.................... 510
Report Totals and Subtotals Key terms ................................ 511
A Quick Example Creating a Subtotal ................................... 512
The Report Format - Sort screen ......................................... 513
The Report Format - Group By screen ................................... 516
A Step-By-Step Example Creating a Report With Subtotals ........ 518
Create Your Report ........................................................... 519
Step 1 - Set your session options ........................................ 519
Step 2 - Select your table .................................................. 520
Step 3 - Select columns .................................................... 520
Step 4 - Retrieve your data ................................................ 521
Step 5 - Display your report ............................................... 521
Sorting Your Report Rows 1 ................................................. 522
Creating Groups and Specifying Subtotals .............................. 522
Step 1 - Specify your group level ........................................ 522
Step 2 - Specify your subtotal ............................................ 523
Step 3 - Display your report ............................................... 523
Creating Report Totals ....................................................... 524
Step 1 - Specify your group level ........................................ 524
Step 2 - Specify your report total ....................................... 525
Step 3 - Display your report ............................................... 525
Creating Levels of Nested Subtotals .................................... 526
Step 1 - Specify your group level ........................................ 526
Step 2 - Specify your subtotal ............................................ 527
Step 3 - Display your report ............................................... 528
How To Format Reports Containing Calculations .................... 528
Create Your Reports ........................................................ 529
Skipping Lines After Groups ............................................... 530
Specifying a SCparator character ........................................ 532
Giving Your Computation a Heading .................................... 533
Displaying Only Summary Computations .............................. 536
Displaying Only Detail Lines .............................................. 537
Skipping To a New Page After Computations .......................... 538
How to Save a Set of Commands as a Qfile ............................ 540
Qfiles and Reports Key Terms ............................................. 542
Creating a Qfile .............................................................. 542
Step 1 - Build the salary report ......................................... 543
Step 2 - Add formatting enhancements ............................... 545
Step 3 - Save the report as a Qfile ..................................... 547
Executing a Qfile ............................................................. 548
Using One Qfile to Create Different Reports .......................... 551
Batch Process to Create a Report .......................................................................................... 593
How to Print Your Report ..................................................................................................... 595
Printing a Report .................................................................................................................. 595
Introducing the OLQ SELECT statement .............................................................................. 597
The OLQ SELECT Statement ................................................................................................. 597
Defining Your Data .............................................................................................................. 599
Some Tips On Using CA OLQ ............................................................................................... 599
  Signing on .......................................................................................................................... 600
  Signing off .......................................................................................................................... 601
  Entering commands ......................................................................................................... 601
  Setting the access to IDMS ............................................................................................. 603
  Tailoring reports .............................................................................................................. 603
Retrieving Information from a Table .................................................................................... 608
  Retrieving All Columns .................................................................................................. 608
  Retrieving Selected Columns ......................................................................................... 609
Eliminating Duplicate Data ..................................................................................................... 609
Displaying Calculations in Columns ..................................................................................... 610
Putting Rows in Order ......................................................................................................... 611
Retrieving Selected Rows ..................................................................................................... 613
  A simple comparison ...................................................................................................... 613
  Complex comparisons .................................................................................................... 615
  Comparisons to a list of values ..................................................................................... 617
  Exclusive comparisons ................................................................................................... 617
  Range comparisons ........................................................................................................ 618
  Character string comparisons ....................................................................................... 619
Using Built-In Functions ....................................................................................................... 620
Testing Your Knowledge ...................................................................................................... 622
Summarizing Information ..................................................................................................... 623
  Summarizing Data In Groups ....................................................................................... 624
  Specifying a Condition For a Group .............................................................................. 625
  Summarizing Information From Another Table .......................................................... 625
Testing Your Knowledge 1 .................................................................................................. 626
Joining Tables or Database Records ................................................................................... 626
Joining Tables ..................................................................................................................... 627
  Joining different tables ................................................................................................. 627
  Joining a table to itself ................................................................................................. 630
  Comparing a column to more than one value ............................................................. 632
Retrieving Information From CA IDMS/DB Records ......................................................... 633
  Retrieving data from a single record .......................................................................... 633
  Retrieving data from two or more records ................................................................. 634
  Retrieving data from a record joined to itself ............................................................. 636
Retrieving Data From Tables and Records ........................................................................ 639
<table>
<thead>
<tr>
<th>Report Type</th>
<th>Reporting Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Reports (DREPORTs 003, 053, 073)</td>
<td>670</td>
</tr>
<tr>
<td>System Summary Report (DREPORT 053)</td>
<td>670</td>
</tr>
<tr>
<td>System Detail and Key Reports (DREPORTs 003 and 073)</td>
<td>671</td>
</tr>
<tr>
<td>User Reports (DREPORTs 004, 054, 074)</td>
<td>673</td>
</tr>
<tr>
<td>User Summary Report (DREPORT 054)</td>
<td>673</td>
</tr>
<tr>
<td>User Detail and Key Reports (DREPORTs 004 and 074)</td>
<td>674</td>
</tr>
<tr>
<td>Program Reports (DREPORTs 005, 055, 075)</td>
<td>678</td>
</tr>
<tr>
<td>Program Summary Report (DREPORT 055)</td>
<td>678</td>
</tr>
<tr>
<td>Program Detail and Key Reports (DREPORTs 005 and 075)</td>
<td>679</td>
</tr>
<tr>
<td>Module Reports (DREPORTs 006, 056, 076)</td>
<td>682</td>
</tr>
<tr>
<td>Module Summary Report (DREPORT 056)</td>
<td>682</td>
</tr>
<tr>
<td>Module Detail and Key Reports (DREPORTs 006 and 076)</td>
<td>683</td>
</tr>
<tr>
<td>File Reports (DREPORTs 007, 057, 077)</td>
<td>686</td>
</tr>
<tr>
<td>File Summary Report (DREPORT 057)</td>
<td>686</td>
</tr>
<tr>
<td>File Detail and Key Reports (DREPORTs 007 and 077)</td>
<td>686</td>
</tr>
<tr>
<td>Record Reports (DREPORTs 008, 038, 058, 078)</td>
<td>689</td>
</tr>
<tr>
<td>Record Summary Report (DREPORT 058)</td>
<td>689</td>
</tr>
<tr>
<td>Record Detail and Key Report (DREPORTs 008 and 078)</td>
<td>690</td>
</tr>
<tr>
<td>Record/Attribute Key Report (DREPORT 038)</td>
<td>693</td>
</tr>
<tr>
<td>Element Reports (DREPORTs 009, 010, 039, 059, 079)</td>
<td>694</td>
</tr>
<tr>
<td>Element Summary Report (DREPORT 059)</td>
<td>694</td>
</tr>
<tr>
<td>Element Detail and Key Reports (DREPORTs 009 and 079)</td>
<td>695</td>
</tr>
<tr>
<td>Inactive Element Detail Report (DREPORT 010)</td>
<td>698</td>
</tr>
<tr>
<td>Attribute/Element Key Report (DREPORT 039)</td>
<td>699</td>
</tr>
<tr>
<td>Site-Specific Entity Reports</td>
<td>700</td>
</tr>
<tr>
<td>Class Reports (DREPORTs 001 and 071)</td>
<td>700</td>
</tr>
<tr>
<td>Attribute Reports (DREPORTs 002 and 072)</td>
<td>702</td>
</tr>
<tr>
<td>User-Defined Entity Reports (DREPORTs 019 and 089)</td>
<td>704</td>
</tr>
<tr>
<td>Teleprocessing Entity Reports</td>
<td>705</td>
</tr>
<tr>
<td>Task Reports (DREPORTs 011, 061, 081)</td>
<td>707</td>
</tr>
<tr>
<td>Task Summary Report (DREPORT 061)</td>
<td>707</td>
</tr>
<tr>
<td>Task Detail and Key Reports (DREPORTs 011 and 081)</td>
<td>708</td>
</tr>
<tr>
<td>Queue Reports (DREPORTs 012, 062, 082)</td>
<td>709</td>
</tr>
<tr>
<td>Queue Summary Report (DREPORT 062)</td>
<td>709</td>
</tr>
<tr>
<td>Queue Detail and Key Reports (DREPORTs 012 and 082)</td>
<td>710</td>
</tr>
<tr>
<td>Destination Reports (DREPORTs 013, 063, 083)</td>
<td>712</td>
</tr>
<tr>
<td>Destination Summary Report (DREPORT 063)</td>
<td>712</td>
</tr>
<tr>
<td>Destination Detail and Key Reports (DREPORTs 013 and 083)</td>
<td>712</td>
</tr>
<tr>
<td>Logical Terminal Reports (DREPORTs 014, 064, 084)</td>
<td>714</td>
</tr>
<tr>
<td>Logical Terminal Summary Report (DREPORT 064)</td>
<td>714</td>
</tr>
<tr>
<td>Logical Terminal Detail and Key Reports (DREPORTs 014 and 084)</td>
<td>715</td>
</tr>
<tr>
<td>Physical Terminal Reports (DREPORTs 015, 065, 085)</td>
<td>716</td>
</tr>
</tbody>
</table>
Physical Terminal Summary Report (DREPORT 065) ............................................................. 716
Physical Terminal Detail and Key Reports (DREPORTs 015 and 085) ................................. 717
Line Reports (DREPORTs 016, 066, 086) ........................................................................... 719
Line Summary Report (DREPORT 066) .............................................................................. 719
Line Detail and Key Reports (DREPORTs 016 and 086) ................................................. 719
Panel Reports (DREPORTs 017, 067, 087) ........................................................................ 721
Panel Summary Report (DREPORT 067) ........................................................................... 721
Panel Detail and Key Reports (DREPORTs 017 and 087) ............................................. 721
Map Reports (DREPORTs 018, 068, 088) ........................................................................... 724
Map Summary Report (DREPORT 068) ......................................................................... 724
Map Detail and Key Reports (DREPORTs 018 and 088) ............................................. 725
Cross-Reference Reports ..................................................................................................... 726
File/Record Cross-Reference Report (DREPORT 020) ................................................... 727
File Synonym Cross-Reference Report (DREPORT 021) ........................................... 729
Record Synonym Cross-Reference Report (DREPORT 022) ....................................... 729
Element Synonym Cross-Reference Report (DREPORT 023) ..................................... 730
Element Description Report (DREPORT 024) ............................................................... 731
Element Designator Report (DREPORT 025) ................................................................. 733
File Activity Report (DREPORT 026) .............................................................................. 734
IDMS Set Activity Report (DREPORT 027) ..................................................................... 735
IDMS Record Activity Report (DREPORT 028) ............................................................ 736
IDMS Area Activity Report (DREPORT 029) ................................................................... 737
Element/Program Cross-Reference Report (DREPORT 030) .................................... 739
Special-Purpose Report Modules .................................................................................... 740
Comment/Nest Resolution Report Module (DREPORT 000) ........................................ 740
Level Number Report (DREPORT 050) ............................................................................ 741
Module Text to Card Utility (DREPORT 051) .................................................................. 741
Module Text to File Utility (DREPORT 052) .................................................................... 742

Task Analyzer Reports ........................................................................................................ 744
Task Analyzer System Output ............................................................................................ 745
CA IDMS Task Analyzer Reports ...................................................................................... 746
About CA IDMS Task Analyzer Billing Reports ......................................................... 746
Tying Task Activity to an ID and a Time ........................................................................ 746
Three Reports One Set of CA IDMS Log or SMF File Statistics .................................... 747
Hierarchical Nature of Reports ..................................................................................... 747
Overview of Billing Reports .......................................................................................... 747
CA IDMS Task Analyzer Billing Details Report ............................................................. 747
Report Fields .................................................................................................................. 748
CA IDMS Task Analyzer Billing Summary Report ............................................................ 750
Report Fields .................................................................................................................. 750
CA IDMS Task Analyzer Billing System Summary Report ................................................................. 753
  Report Fields ................................................................................................................................. 753
About CA IDMS Task Analyzer Program Reports ............................................................................ 755
  Three Reports One Set of CA IDMS Log or SMF File Statistics .................................................... 756
  Hierarchical Nature of Reports ........................................................................................................ 756
Overview of Program Reports ........................................................................................................... 756
CA IDMS Task Analyzer Program Details Report ............................................................................. 756
  Report Fields ................................................................................................................................. 756
CA IDMS Task Analyzer Program Summary Report ......................................................................... 760
  Report Fields ................................................................................................................................. 760
CA IDMS Task Analyzer Program System Summary Report .............................................................. 763
  Report Fields ................................................................................................................................. 763
About CA IDMS Task Analyzer CA ADS Reports ............................................................................ 765
  Three Reports One Set of CA IDMS Log or SMF File Statistics .................................................... 766
  Hierarchical Nature of Reports ........................................................................................................ 766
Overview of CA ADS Reports ........................................................................................................... 766
CA IDMS Task Analyzer CA ADS Details Report ............................................................................. 766
  Report Fields ................................................................................................................................. 767
CA IDMS Task Analyzer CA ADS Summary Report ......................................................................... 769
  Report Fields ................................................................................................................................. 769
CA IDMS Task Analyzer CA ADS System Summary Report .............................................................. 772
  Report Fields ................................................................................................................................. 772
About CA IDMS Task Analyzer Abend Report .................................................................................. 775
  Tying Abend Activity to an ID and a Time ......................................................................................... 775
  One Report One Set of CA IDMS Log Statistics .............................................................................. 775
Overview of Abend Reports .............................................................................................................. 775
CA IDMS Task Analyzer Abend Report ............................................................................................ 775
  Report Fields ................................................................................................................................. 776
About CA IDMS Task Analyzer Program Loads Report ................................................................. 777
  Tying Task Activity to an ID and a Time ......................................................................................... 778
  One Report One Set of CA IDMS Log Statistics .............................................................................. 778
Overview of Program Loads Reports ................................................................................................ 778
CA IDMS Task Analyzer Program Loads Report .............................................................................. 778
  Report Fields ................................................................................................................................. 778
About CA IDMS Task Analyzer Integrated Index Reports ............................................................... 781
  Tying Task Activity to an ID and a Time ......................................................................................... 781
  Three Reports One Set of CA IDMS Log or SMF File Statistics .................................................... 782
  Hierarchical Nature of Reports ........................................................................................................ 782
Overview of Integrated Index Reports .............................................................................................. 782
CA IDMS Task Analyzer Integrated Index Details Report ............................................................... 782
  Report Fields ................................................................................................................................. 783
CA IDMS Task Analyzer Integrated Index Summary Report ........................................................... 786
  Report Fields ................................................................................................................................. 786
Integrated Index System Summary Report ................................................................. 789
Report Fields ........................................................................................................ 789
CA IDMS Task Analyzer Ranking Report ................................................................. 791
Report Fields ........................................................................................................ 792
CA IDMS Task Analyzer Input Parameter Report ..................................................... 793
Report Fields ........................................................................................................ 794
CA IDMS Task Analyzer Parameters ................................................................. 794
Parameters and Their Uses .................................................................................... 795
Order of Parameter Statements ............................................................................ 795
Maximum Number of Reports Possible Per Execution ......................................... 796
Billing Report Parameters .................................................................................... 796
  How RUTYPE, RUNAME, and NAME Parameters Interrelate .......................... 799
Program Report Parameter .................................................................................. 800
CA ADS Report Parameters .................................................................................. 802
Abend Report Parameters ..................................................................................... 805
Program Loads Report Parameters ...................................................................... 808
Integrated Index Report Parameters .................................................................... 810
Ranking Report Parameters .................................................................................. 813
Task Analyzer Operations ...................................................................................... 818
CA IDMS Task Analyzer Operating Requirements ................................................ 818
CA IDMS Task Analyzer Statistics Plan Options .................................................. 820
  Screen Fields ..................................................................................................... 820
Task Analyzer z/OS Operation .............................................................................. 823
  JCL to Create an Extract File from the CA IDMS Log ...................................... 824
  JCL to Create an Extract File from the SMF File .............................................. 826
  Report Execution JCL ....................................................................................... 827
CA IDMS Task Analyzer z/VSE Operations ......................................................... 831
  z/VSE File Assignments .................................................................................... 831
CA IDMS Task Analyzer z/VM Operations .......................................................... 835
  The Extract EXEC USFXTRCT ........................................................................ 836
  The Report Execution EXEC USFRPRT ............................................................ 838
External Request Element Extension 1 ................................................................. 840
CA Culprit Procedure JCL .................................................................................... 841

z/VM and z/VM Commands .................................................................................. 844
Central Version Modifications .............................................................................. 845
Creating the SYSIPT or SYSIDMS file ................................................................. 845

Endevor and DB for IDMS Reporting .................................................................. 847
CCDB Reporting ................................................................................................................... 847
CA Culprit Modules ........................................................................................................... 847
  Common Modules ............................................................................................................. 848
  Input Modules .................................................................................................................. 848
  Output Modules ............................................................................................................... 848
Running Reports ................................................................................................................ 849
  Selecting Input .................................................................................................................. 850
    Legend .............................................................................................................................. 850
  Report Groups ................................................................................................................... 851
  Selection Criteria ............................................................................................................ 851
  Selection Criteria Coding Rules ..................................................................................... 852
  Creating an Archive of Change Log Entries ................................................................. 853
  Using an Archive of Change Log Entries ....................................................................... 853
  Coding Examples ............................................................................................................ 853
  Sample JCL ....................................................................................................................... 854
    Legend .............................................................................................................................. 854
Report Formats ................................................................................................................... 856
  Report No. 00 Reporting Commands and Messages ...................................................... 857
  Report No. 01 Change Log Detail ..................................................................................... 857
  Report No. 02 User Change Log Detail ........................................................................... 858
  Report No. 03 CCID Change Log Detail ......................................................................... 859
  Report No. 04 Entity Change Log Detail ......................................................................... 860
  Report No. 05 CCID Change Log Summary ................................................................. 860
  Report No. 06 CCID List .................................................................................................. 861
  Report No. 07 User Change Log Summary .................................................................... 861
  Report No. 08 User List ................................................................................................... 862
  Report No. 09 Status List ................................................................................................. 863
  Report No. 10 Management Group List .......................................................................... 863
  Report No. 11 Entity / CCID Change Log Summary .................................................. 864
  Report No. 12 Entity Change Log Summary .................................................................. 865
  Report No. 13 CCID / Entity Status .............................................................................. 866
  Report No. 14 Entity Status History .............................................................................. 867
  Report No. 15 Post Migration Activity .......................................................................... 867
  Report No. 16 Source Migration Summary .................................................................... 868
  Report No. 17 Target Migration Summary ..................................................................... 869
  Report No. 18 Entity Signout Detail ............................................................................... 869
  Report No. 19 Entity Signout by User .......................................................................... 870
  Report No. 20 Entity Signout by CCID ........................................................................ 870
  Report No. 21 Security Class/User ............................................................................... 871
  Report No. 22 Security Class/CCID .............................................................................. 871
  Report No. 23 Management Group/CCID ..................................................................... 871
  Report No. 24 Entity/User Change Log Summary ...................................................... 872
How to Save Your Report ................................................................. 875
OLQ Report Key Terms ................................................................. 877
Creating an OLQ Report ................................................................. 877
Saving the Report ........................................................................ 877
Using a Saved Report ................................................................... 879
Modifying a Saved Report ............................................................. 880
Deleting Saved Reports ................................................................. 882
   Deleting All Saved Reports ......................................................... 883

Modifying CA IDMS Reports ......................................................... 884
Obtaining Report Module Listings .................................................. 884
   Types of Listings ....................................................................... 884
Coding in CA Culprit for CA IDMS ............................................... 885
   CA Culprit for CA IDMS Coding Guidelines .............................. 886
   Types of CA Culprit for CA IDMS Parameters ......................... 886
      Parameters Within Each Category ......................................... 886
   Report-Specific and Global Parameters .................................... 887
Edit Parameters ............................................................................ 887
      Functions of Each Column ..................................................... 888
   Summary of Parameter Types .................................................... 888
Temporary Modifications .............................................................. 891
   Copying the Report Module ...................................................... 891
   Deleting Columns .................................................................... 892
   Changing Headings .................................................................. 893
   Selecting Specific Data ............................................................. 893
   Specifying a New Sort Sequence .............................................. 894
   Moving Columns ..................................................................... 894
Permanent Modifications .............................................................. 895
Reporting

This section helps you with information on various IDMS reports:

- ASF Row-Level Security Reports -- IREPORTS (see page 28)
- CA ADS Reports -- AREPORTS (see page 32)
- CA ICMS Catalog Reports -- DREPORTS (see page 44)
- CA IDMS/DB Journal Reports -- JREPORTS (see page 61)
- CA IDMS/DB SQL Dictionary Reports -- QREPORTS (see page 101)
- CA IDMS Extractor Reports (see page 122)
- CA IDMS Module Listing (see page 130)
- CA IDMS reports z/VSE JCL (see page 137)
- Compliance Reporting (see page 143)
- DB Analyzer System Output Reports (see page 144)
- DC/UCF Statistics Reports -- SREPORTS (see page 202)
- DC/UCF System Reports -- CREPORTS (see page 256)
- IDMS Dictionary Migrator Reports (see page 302)
- IDMS Log Analyzer Reports (see page 313)
- IDMS Performance Monitor System Reports (see page 345)
- Online Query (OLQ) Reports (see page 449)
- Other CA IDMS Reporting Facilities (see page 659)
- Standard Dictionary Reports -- DREPORTS (see page 664)
- Task Analyzer Reports (see page 744)
- z/VM and z/VM Commands (see page 844)
- Endevor and DB for IDMS Reporting (see page 847)
- How to Save Your Report (see page 875)
- Modifying CA IDMS Reports (see page 884)
ASF Row-Level Security Reports -- IREPORTS

To obtain summary and detail reports about tables that enforce row-level security, follow these steps:

1. By using ASF, obtain the required passkey and authority:
   
   a. Obtain a READ passkey to the $SRT-OST-CROSS-REFERENCE$ table.
   
   b. Obtain authority to access the $SECURITY-RUNTIME-TABLE$ table.

   **Note:** For information on assigning passkeys and authority, see the *CA IDMS ASF Using section*.

2. Run an IREPORT report, under the IDMS DC/UCF central version, that contains the following input parameters. The source code for the IREPORTs must be stored in the ASF dictionary as modules named IREPORT 001, IREPORT 002, and so on.

**Contents:**

- Syntax (see page 28)
- Parameters (see page 28)
- Summary Table (see page 29)
- Example (see page 29)

**Syntax**

The following syntax and syntax rules for the CA Culprit for CA IDMS parameters required to run row-level security reports appear, followed by examples.

```
Col 2
   DATABASE DICTNAME=asf-dictionary-name
   PROFILE USER=user-id PW=password
   =COPY 'IREPORT 00n'
```

**Parameters**

- **DICTNAME=asf-dictionary-name**
  
  Specifies the name of the dictionary that contains the CA Culprit for CA IDMS source modules and to which the $SRT-OST-CROSS-REFERENCE$ table is defined.

- **PROFILE USER=user-id**
  
  Specifies the ID of the user running the IREPORT. The user must have a READ passkey to the $SRT-OST-CROSS-REFERENCE$ table and must be authorized to access the $SECURITY-RUNTIME-TABLE$ table.
- **PW=password**
  Specifies the password of the user running the CA Culprit for CA IDMS reports. The user must have a READ passkey to the $SRT-OST-CROSS-REFERENCE$ table and must be authorized to access the $SECURITY-RUNTIME-TABLE$ table.

- **IREPORT 00n**
  Specifies the report number as described in the following summary table.

### Summary Table

<table>
<thead>
<tr>
<th>IREPORT</th>
<th>Title</th>
<th>Sorted by</th>
</tr>
</thead>
<tbody>
<tr>
<td>IREPORT 001</td>
<td>Row Level Security Summary Report</td>
<td>Table name</td>
</tr>
<tr>
<td>IREPORT 002</td>
<td>Row Level Security Detail Report</td>
<td>Table name</td>
</tr>
<tr>
<td>IREPORT 003</td>
<td>Row Level Security Summary Report by User</td>
<td>User id</td>
</tr>
<tr>
<td>IREPORT 004</td>
<td>Row Level Security Detail Report by User</td>
<td>User id</td>
</tr>
<tr>
<td>IREPORT 005</td>
<td>Row Level Security Summary Report by Owner/Security Name</td>
<td>Security name within user id</td>
</tr>
</tbody>
</table>

### Example

In this example, user BEAR has a READ passkey to the $SRT-OST-CROSS-REFERENCE$ table and authority to access the $SECURITY-RUNTIME-TABLE$ table. The user requests a report that lists detailed information on each table in ASFDICT that enforces row-level security.

```
DATABASE DICTNAME=ASFDICT
PROFILE USER=BEAR PW=CUB
=COPY 'IREPORT 002'
```

### ASF Row-Level Security Reports

#### Sample IREPORT 001:

```
REPORT NO.  01
IREPORT 001
INFORMATION CENTER MANAGEMENT SYSTEM
ROW LEVEL SECURITY SUMMARY REPORT
mm/dd/yy PAGE 1
TABLE.: BUDGET2
OWNER.: QAL
TDN....: 247
USER
QAD
JAN GT 10000
```

#### Sample IREPORT 002:
Sample IREPORT 003:

<table>
<thead>
<tr>
<th>TABLE</th>
<th>USER</th>
<th>SECURITY NAME</th>
<th>TDN</th>
<th>WHERE CLAUSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEMO</td>
<td>JSS</td>
<td>DEMO123</td>
<td>137</td>
<td>LNAME = 'ABC'</td>
</tr>
</tbody>
</table>

Sample IREPORT 004:

<table>
<thead>
<tr>
<th>TABLE</th>
<th>USER</th>
<th>SECURITY NAME</th>
<th>TDN</th>
<th>WHERE CLAUSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUDGET2</td>
<td>QAL</td>
<td>QALPASS</td>
<td>247</td>
<td>JAN GT 10000</td>
</tr>
<tr>
<td>SMALLTAB</td>
<td>QAL</td>
<td>QALPASS2</td>
<td>260</td>
<td>ACCOUNT EQ '1'</td>
</tr>
</tbody>
</table>

Sample IREPORT 005:

<table>
<thead>
<tr>
<th>TABLE</th>
<th>USER</th>
<th>SECURITY NAME</th>
<th>TDN</th>
<th>WHERE CLAUSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEMO</td>
<td>QAC</td>
<td>SECURROW</td>
<td>223</td>
<td>COL1 = 69</td>
</tr>
</tbody>
</table>

Field Descriptions

- **TABLE**
  Name of a table for which row-level security is established.

- **OWNER**
  Owner of the table.

- **TDN**
  Table definition number of the table.

- **SECURITY NAME (SECNAME)**
  Security name assigned to the table.

- **USER**
  Id of a user authorized to access certain rows of the table.

- **WHERE**
  Specifies criteria that define the data that the user is permitted to see. If a user has unlimited access to the table (that is, no WHERE clause exists), the following literal appears:
  
  *** NO RESTRICTIONS ***

- **STATUS**
  Specifies whether the $SECURITY-RUNTIME-TABLE$ entry has been validated; that is, whether user, owner, security name, and WHERE criteria are entered correctly in the $SECURITY-RUNTIME-TABLE$ table. The report generates the following messages:

  - (V) VALIDATED
  - (I) REQUIRES VALIDATION
  - (E) ERROR IN VALIDATION

- **GROUP**
  Name of the group that has authority to access the table.
CA ADS reports document dialogs defined to the data dictionary and identify associated components. CA ADS reports can be used to:

- Assist in test to production migration (for example, AREPORT 002 identifies all the components associated with a particular dialog that must be migrated along with the dialog)
- Determine what dialogs have to be regenerated when a particular component changes (for example, AREPORT 005 lists all dialogs associated with a particular subschema)

The following table lists the CA ADS reports in order by report module number:

<table>
<thead>
<tr>
<th>AREPORT Module</th>
<th>Report Title</th>
<th>KEY Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>ADS Dialogs and Their Components (Detail)</td>
<td></td>
</tr>
<tr>
<td>002</td>
<td>ADS Dialogs and Their Components (Key)</td>
<td>KEY PROG-NAME-051 'dialog-name'</td>
</tr>
<tr>
<td>003</td>
<td>ADS Dialogs by Process Key</td>
<td>KEY MOD-NAME-067 'process-name'</td>
</tr>
<tr>
<td>004</td>
<td>ADS Dialogs by Record Key</td>
<td>KEY RSYN-NAME-079 'record-name'</td>
</tr>
<tr>
<td>005</td>
<td>ADS Dialogs by Subschema Key</td>
<td>KEY SS-NAM-026 'subschema-name'</td>
</tr>
<tr>
<td>006</td>
<td>ADS Dialogs by Map Key</td>
<td>KEY MAP-NAME-098 'map-name'</td>
</tr>
</tbody>
</table>

For more information, see the following topics:
- Producing CA ADS Reports (see page 32)
- AREPORT 001 and 002 - ADS Dialogs and Their Components (see page 34)
- AREPORT 003 - ADS Dialogs by Process Key (see page 37)
- AREPORT 004 - ADS Dialogs by Record Key (see page 39)
- AREPORT 005 - ADS Dialogs by Subschema Key (see page 41)
- AREPORT 006 - ADS Dialogs by Map Key (see page 42)

Producing CA ADS Reports

Contents

- Syntax (see page 33)
- Parameters (see page 33)
- Examples (see page 33)

CA ADS reports are produced by submitting a job that includes the standard JCL shown in Appendices A through D and user-supplied control parameters. Syntax for the control parameters is shown below, followed by examples. Coding is freeform except for the starting column of each parameter.
Parameters

Syntax rules appear in Section 1; except as follows:

- **AREPORT=areport-number**
  Identifies the report module name, where `areport-number` is a value in the range 001 through 006; leading zeros can be omitted. The parameter must be coded starting in column 1. Only one report module can be specified per run.

- **KEY**
  Specifies the parameter type, starting in column 2. A KEY parameter is required for AREPORTs 002 through 006. More than one KEY parameter can be specified per report.

- **key-field-name**
  Identifies the key field. The Summary Table lists possible values for each report.

- **'key-field-value'**
  Specifies a value or a list of values for the key field. Each value must be enclosed in single quotation marks. A list of items must be enclosed in parentheses and each item must be separated from another by a space or a comma.

Examples

Example 1
Data dictionary CULPDICT contains report module AREPORT 001, which is requested to report on dialogs defined to data dictionary DOCUDICT. The SELECT parameter selects only those dialogs that were created on a specified date. By request, the input parameters will appear on the Sequential and Input Parameter Listings for the report.

```
DATABASE DICTNAME=CULPDICT DBNAME=DOCUDICT
PARAM=LIST
AREPORT=001
SELECT BUFFER WHEN DATE-CREATED-051 EQ 'mm/dd/yy'
```

**Example 2**

AREPORT reports on all dialogs associated with the EMPLOYEE and DEPARTMENT records. The SELECT parameter selects only those dialogs that were created on a specified date. By default, the input parameters will not appear on the report listings.

```
DATABASE DICTNAME=CULPDICT DBNAME=DOCUDICT
AREPORT=004
KEY RSYN-NAME-079 ('EMPLOYEE' 'DEPARTMENT')
SELECT PROG-051 WHEN DATE-CREATED-051 EQ 'mm/dd/yy'
```

**AREPORT 001 and 002 - ADS Dialogs and Their Components**

**Contents**

- Contents (see page 34)
- Field Descriptions (see page 35)

**Contents**

The ADS Dialogs and Their Components reports list information on dialogs that have been generated within the DC/UCF system. AREPORT 001 is a detail report that lists every dialog; AREPORT 002 is a key report that lists selected dialogs. For each dialog, the reports provide information on the following components, if applicable:

- Subschema
- Map
- Processes
- Database records
- Logical records

The fields and format of the two reports are the same.

**Sample AREPORT 002:**
Field Descriptions

A description of the fields in AREPORTs 001 and 002 follows:

- **DIALOG, VERS**
  Name and version number of the dialog.

- **DATE CREATED**
  Date the dialog was added to the data dictionary.

- **DATE LAST UPDATED**
  Date the dialog was last modified.

- **MAINLINE DIALOG**
  A literal that appears if the dialog is a mainline dialog.

- **SUBSCHEMA NAME**
  Name of the subschema associated with the dialog. If no subschema is associated with the dialog, the following message appears:
  
  * * NO SUBSCHEMA FOR THIS WK-DIALOG * *

- **SCHEMA NAME, SCHEMA VERSION**
  Name and version number of the schema associated with the subschema.

- **MAP**
  Name and version number of the map associated with the dialog. If no map is associated with the dialog, the following message appears:
  
  * * MAPLESS DIALOG * *
- **TYPE**
  Indicates the type of map associated with the dialog:
  - ONL is the dialog's online map.
  - IN is the dialog's input file map.
  - OUT is the dialog's output file map.

- **DATE**
  Date stamp for the map.

- **TIME**
  Time stamp for the map.

- **DDNAME-IN**
  (CA ADS Batch only) Indicates the ddname of an input file map.

- **OUT**
  (CA ADS Batch only) Indicates the ddname of an output file map.

- **SUSP**
  (CA ADS Batch only) Indicates the ddname of the suspense file associated with a dialog.

- **MAP ENTRY PT**
  Indicates that the map is the entry point for the dialog. This is true even if the dialog has a premap process.

- **PROCESS NAME, VERSION**
  Name and version number of a process associated with the dialog. If no processes are associated with the dialog, the following message appears:
  *
  * * NO PROCESSES FOR THIS WK-DIALOG * *

- **EXECUTE ON EDIT ERRORS**
  A literal that appears beside the response processes designated to execute even when there are map input errors.

- **DATE CREATED, BY**
  Date the process was added to the data dictionary and the ID of the user who created it.

- **DATE LAST UPDATED, BY**
  Date the process was last modified and the ID of the user who modified it.
• **CONTROL KEY/EVENT**
  Either of the following:

  - (CA ADS) The control key, if any, that causes the process to be executed if the process is a response process for the dialog.
  
  - (CA ADS Batch) A batch control event (EOF or IOERR), associated with a batch dialog, that causes a process to be executed when its associated condition (end-of-file, I/O error) is met.

• **RESPONSE FIELD VALUE**
  The response field value, if any, that causes the process to be executed if the process is a response process for the dialog.

• **RECORD NAME, VERSION**
  Name and version number of each record associated with the dialog. The following descriptors appear next to the records, as applicable:

  - *SUBSCHEMA* indicates that the record is associated with the dialog as a subschema record.
  
  - *WORK* indicates that the record is associated with the dialog as a work record.
  
  - *N/C* indicates that the record is associated with the dialog as a new copy record.
  
  - *STATUS* indicates that the record is associated with the dialog as a status definition record.
  
  - *IN LR* indicates that the record is included in a logical record associated with the dialog.
  
  - *INMAP*, *I/OMAP, *ONLMAP, *OUTMAP indicate a record associated with a dialog’s:
    
    - Input map file
    
    - Input and output map file
    
    - Online map
    
    - Output map file

  If no records are associated with the dialog, the following message appears:

  * * NO DATABASE RECORDS FOR THIS WK-DIALOG * *

• **RECORD NAME, SUBSCHEMA NAME**
  Record name and subschema name for each logical record associated with the dialog, along with a * LOGICAL RECORD * flag. If no logical records are associated with the dialog, the following message appears:

  * * NO LOGICAL RECORDS FOR THIS WK-DIALOG * *
AREPORT 003, the ADS Dialogs by Process Key report, lists all dialogs associated with one or more specified processes.

Sample AREPORT 003:

Field Descriptions

A description of the fields in AREPORT 003 follows:

- **PROCESS, VERS**
  Name and version number of the process specified in the KEY parameter.

- **DATE CREATED, BY**
  Date the process was added to the data dictionary and the ID of the user who created it.

- **LAST UPDATED, BY**
  Date the process was last modified and the ID of the user who modified it.

- **DIALOG NAME, DIALOG VERSION**
  Name and version number of all dialogs associated with the process. If no dialogs are associated with the process specified in a KEY parameter, the following message appears:

  * * NO DIALOGS USE THIS MODULE * *

- **DATE CREATED**
  Date the dialog was added to the data dictionary.

- **DATE LAST UPDATED**
  Date the dialog was last modified.

- **MAINLINE DIALOG**
  Indicates (with an X) whether the dialog is a mainline dialog.

- **PROCESS TYPE**
  Indicates whether the process is used as a premap or response process for the dialog.
CONTROL KEY/EVENT FOR PROCESS
Indicates either of the following:

- (CA ADS) The control key, if any, that causes the process to be executed if the process is used as a response process for the dialog.

- (CA ADS Batch) A batch control event (EOF or IOERR), associated with a batch dialog, that causes a process to be executed when its associated condition (end-of-file, I/O error) is met.

RESPONSE FIELD VALUE FOR PROCESS
The response field value, if any, that causes the process to be executed if the process is a response process for the dialog.

EXECUTE ON EDIT ERRORS
Indicates (with an X) whether a response process is designated to execute even when there are map input errors.

AREPORT 004 - ADS Dialogs by Record Key

Contents

- Contents (see page 39)
- Field Descriptions (see page 40)

Contents

AREPORT 004, the ADS Dialogs by Record Key report, lists all dialogs associated with one or more specified records.

Sample AREPORT 004:

<table>
<thead>
<tr>
<th>DIALOG NAME</th>
<th>DIALOG VERSION</th>
<th>DATE CREATED</th>
<th>DATE LAST UPDATED</th>
<th>MAINLINE DIALOG</th>
<th>MAP RECORD</th>
<th>SUBSCHEMA RECORD</th>
<th>WORK RECORD</th>
<th>NEW COPY RECORD</th>
<th>IN LOGICAL RECORD</th>
<th>STATUS RECORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADM01D</td>
<td>1</td>
<td>mm/dd/yy</td>
<td></td>
<td></td>
<td>X</td>
<td>ONL</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>DLR01AD</td>
<td>1</td>
<td>mm/dd/yy</td>
<td></td>
<td></td>
<td>X</td>
<td>ONL</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>DRPP01D</td>
<td>1</td>
<td>mm/dd/yy</td>
<td></td>
<td></td>
<td>X</td>
<td>ONL</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>LDCT01D</td>
<td>1</td>
<td>mm/dd/yy</td>
<td></td>
<td></td>
<td>X</td>
<td>ONL</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>LDTD01D</td>
<td>1</td>
<td>mm/dd/yy</td>
<td></td>
<td></td>
<td>X</td>
<td>ONL</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>LDTD01BD</td>
<td>1</td>
<td>mm/dd/yy</td>
<td></td>
<td></td>
<td>X</td>
<td>ONL</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>LDTD01D</td>
<td>1</td>
<td>mm/dd/yy</td>
<td></td>
<td></td>
<td>X</td>
<td>ONL</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>LNTN01D</td>
<td>1</td>
<td>mm/dd/yy</td>
<td></td>
<td></td>
<td>X</td>
<td>ONL</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>LNTN03D</td>
<td>1</td>
<td>mm/dd/yy</td>
<td></td>
<td></td>
<td>X</td>
<td>ONL</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>LRDA01D</td>
<td>1</td>
<td>mm/dd/yy</td>
<td></td>
<td></td>
<td>X</td>
<td>ONL</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>LRFC01D</td>
<td>1</td>
<td>mm/dd/yy</td>
<td></td>
<td></td>
<td>X</td>
<td>ONL</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>LRFC02D</td>
<td>1</td>
<td>mm/dd/yy</td>
<td></td>
<td></td>
<td>X</td>
<td>ONL</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>LRFD01D</td>
<td>1</td>
<td>mm/dd/yy</td>
<td></td>
<td></td>
<td>X</td>
<td>ONL</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Field Descriptions

A description of the fields in AREPORT 004 follows:

- **RECORD NAME, VERSION**
  Name and version number of the subschema record, map record, or work record specified in the KEY parameter. If multiple versions of the record exist in the data dictionary, they are listed in ascending order.

- **DIALOG NAME, DIALOG VERSION**
  Names and version numbers of all dialogs associated with the record. If no dialogs are associated with a particular version of a record, the following message appears:
  
  * * NO DIALOGS USE THIS RECORD * *

- **DATE CREATED**
  Date the dialog was added to the data dictionary.

- **DATE LAST UPDATED**
  Date the dialog was last modified.

- **MAINLINE DIALOG**
  Indicates (with an X) whether the dialog is a mainline dialog.

- **MAP RECORD**
  Indicates the type of map associated with the dialog:

  - ONL is an online map.
  - IN is an input map file.
  - OUT is an output map file.
  - I/O is an input and output map file.

- **SUBSCHEMA RECORD**
  Indicates (with an X) whether the record is associated with the dialog as a subschema record.

- **WORK RECORD**
  Indicates (with an X) whether the record is associated with the dialog as a work record.

- **NEW COPY RECORD**
  Indicates (with an X) whether the record is associated with the dialog as a new copy record.

- **IN LOGICAL RECORD**
  Indicates (with an X) whether the record is included in a logical record associated with the dialog.

- **STATUS RECORD**
  Indicates (with an X) whether the record is associated with the dialog as a status definition record.
AREPORT 005 - ADS Dialogs by Subschema Key

Contents

- Contents (see page 41)
- Field Descriptions (see page 41)

Contents

AREPORT 005, the ADS Dialogs by Subschema Key report, lists all dialogs associated with one or more specified subschemas.

Sample AREPORT 005:

```
REPORT NO. 05          CA ADS DIALOG REPORTER          Rnn.n          mm/dd/yy PAGE 1
CA ADS DIALOGS BY SUBSCHEMA KEY
SUBSCHEMA NAME...: EMPSSLR  SCHEMA NAME....: EMPSCHM  SCHEMA VERSION....: 1

DIALOG NAME   DIALOG VERSION   DATE CREATED   DATE LAST UPDATED   MAINLINE DIALOG
OTFT03D       1               mm/dd/yy       mm/dd/yy             
RENO01D       1               mm/dd/yy       mm/dd/yy             X  
RENO02D       1               mm/dd/yy       mm/dd/yy             X  
RENO03D       1               mm/dd/yy       mm/dd/yy             X  
```

Field Descriptions

A description of the fields in AREPORT 005 follows:

- **SUBSCHEMA NAME**
  Name of the subschema in the KEY parameter.

- **SCHEMA NAME, SCHEMA VERSION**
  Name and version number of the schema to which the subschema belongs.

- **DIALOG NAME, DIALOG VERSION**
  Names and version numbers of all dialogs associated with the subschema. If no dialogs are associated with the subschema named in the KEY parameter, the following message appears:

  * * NO DIALOGS USE THIS SUBSCHEMA * *

- **DATE CREATED**
  Date the dialog was added to the data dictionary.

- **DATE LAST UPDATED**
  Date the dialog was last modified.

- **MAINLINE DIALOG**
  Indicates (with an X) whether the dialog is a mainline dialog.
AREPORT 006 - ADS Dialogs by Map Key

Contents

- Contents (see page 42)
- Field Descriptions (see page 42)

AREPORT 006, the ADS Dialogs by Map Key report, lists all dialogs associated with one or more specified maps.

Sample AREPORT 006:

<table>
<thead>
<tr>
<th>REPORT NO.</th>
<th>CA ADS DIALOG REPORTER</th>
<th>Rnn.n</th>
<th>mm/dd/yy PAGE</th>
<th>1</th>
</tr>
</thead>
</table>

CA ADS DIALOGS BY MAP KEY

**MAP NAME,,,: LRTD01M VERS: 1 MAP-DATE,,,: mm/dd/yy MAP-TIME,,,: hh:mm:ss **

**DIALOG,,,: LRTD01D VERS: 1 MAP USE: ONL DATE CREATED: mm/dd/yy LAST UPDATED: mm/dd/yy MAINLINE DIALOG **

Field Descriptions

A description of the fields in AREPORT 006 follows:

- **MAP NAME, VERS**
  Name and version number of the map specified in the KEY parameter.

- **DIALOG, VERS**
  Name and version number of a dialog associated with the map. If no dialogs are associated with the map named in the KEY parameter, the following message appears:
  
  * * NO DIALOGS USE THIS MAP * *

- **MAP-DATE**
  Date stamp for the map.

- **MAP-TIME**
  Time stamp for the map.

- **MAP USE**
  Indicates the type of map associated with the dialog:
  
  - ONL is an online map.
  - IN is an input map file.
  - OUT is an output map file.
• DATE CREATED
  Date the dialog was added to the data dictionary.

• LAST UPDATED
  Date the dialog was last modified.

• MAINLINE DIALOG
  A literal that appears if the dialog is a mainline dialog.
CA ICMS Catalog Reports -- DREPORTS

The catalog is a directory of information used by a number of CA IDMS products, including the Automatic System Facility (ASF) and the Information Center Management System. Eight standard catalog reports provide information on the contents of the catalog. These catalog reports are a subset of the standard DREPORTs.

Catalog-related records are stored in the DDLDML area of the dictionary. Information displayed in the fields of the catalog reports is taken from the corresponding fields of the dictionary records.

Catalog reports can:

- Help the information center administrator monitor the contents of the dictionary and the catalog structure
- Help users monitor their own private data

For more information about the structure of the catalog records and how catalog records are defined to the data dictionary, see the CA IDMS Dictionary Structure Reference section and the following topics:

- Summary of Catalog Reports (see page 44)
- Producing Catalog Reports (see page 45)
- DREPORT 090 - Catalog Summary Report (see page 47)
- DREPORT 091 - Catalog Detail Report (see page 48)
- DREPORT 092 - Group Detail Report (see page 50)
- DREPORT 093 - User Detail Report (see page 52)
- DREPORT 094 - Folder Detail Report (see page 54)
- DREPORT 095 - Object Detail Report (see page 55)
- DREPORT 096 - Catalog Summary Key Report (see page 57)
- DREPORT 097 - Catalog Detail Key Report (see page 58)

Summary of Catalog Reports

The following table lists the catalog reports in order of presentation in this section:

<table>
<thead>
<tr>
<th>DREPORT Module</th>
<th>DREPORT Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>090</td>
<td>Catalog Summary Report</td>
</tr>
<tr>
<td>091</td>
<td>Catalog Detail Report</td>
</tr>
<tr>
<td>092</td>
<td>Group Detail Report</td>
</tr>
<tr>
<td>093</td>
<td>User Detail Report</td>
</tr>
<tr>
<td>094</td>
<td>Folder Detail Report</td>
</tr>
<tr>
<td>095</td>
<td>Object Detail Report</td>
</tr>
</tbody>
</table>
Producing Catalog Reports

Contents

- Syntax (see page 45)
- Parameters (see page 46)
- Examples (see page 46)

A catalog report is produced by submitting a job that includes the standard JCL for CA Culprit for CA IDMS report writers and report-specific control statements. The JCL for z/OS, z/VSE and z/VM operating systems is shown in Appendixes A through D, respectively.

Syntax

```
Col 2 ▼ DATABASE ▼ DICTNAME=dictionary-name
    DBNAME=database-name
    DICTNODE=dictionary-node-name
    DBNODE=database-node-name

Col 2 ▼ PARAM= ▼ LIST ▼ NOLIST ▼ EJECT

Col 1 ▼ dREPORT= ▼ dreport-number

Col 2 ▼ KEY CAT-ENTRY-CALCKEY-160

Col 1 ▼ 'owner-name' ▼ Col 35 ▼ 'member-name'
    Col 67

Col 2 ▼ Select ▼ BYPass ▼ BUFFER ▼ record-name ▼ level-number ▼ IN PATH ▼ path-id

WHEN boolean-expression
```
Parameters

General syntax rules for DREPORTs are presented in Section1. Syntax rules specific to the catalog reports are described below:

- **dREPORT=**\textit{dreport-number}
  
  Identifies the catalog report that you want to run. The D of DREPORT is optional. If D is specified, DREPORT must begin in column 1; if D is not specified, REPORT must begin in column 2. 
  
  \textit{Dreport-number} is the 3-digit report number. Catalog reports are assigned numbers 090 through 097. Leading zeros can be omitted. 
  
  Multiple reports can be requested in the same job run, except for the key reports (DREPORTs 096 and 097), which must be run alone.

- **KEY CAT-ENTRY-CALCKEY-160**
  
  (For DREPORTs 096 and 097 only) Identifies the report as a key report; code the literal starting in column 2.

- **'owner-name member-name'**
  
  Specifies the owner and entity name of the selected entity. This field must be exactly 64 characters long and must be enclosed in quotation marks. 
  
  \textit{Owner-name} identifies either a user or the catalog (CORP); it can be up to 32 characters long. If it is less than 32 characters, pad the name with blanks so that the member name starts at character 33 of the literal (that is, within the quotation marks). 
  
  \textit{Member-name} specifies the selected entity. \textit{Member-name} can be up to 32 characters long; it must begin in column 33 of the literal. If \textit{member-name} is less than 32 characters, pad the entry with blanks so that the entry is exactly 32 characters long. The closing quotation mark immediately follows \textit{member-name}. (See the following Example 2.)

Examples

**Example 1**

These control statements can be used to run a summary catalog report. The report modules used to run the report are in the default dictionary; the data for the report is taken from the ASFDICT dictionary.

```
DATABASE DBNAME=ASFDICT
DREPORT=90
```

**Example 2**

These control statements can be used to report on all entities whose path includes CORP.FINANCIAL ANALYSIS DATA. The asterisk is the CULPRIT continuation character. Note that CORP begins in column 3 and FINANCIAL ANALYSIS DATA begins in column 35, the 33rd column of the literal. The closing quotation mark is in column 67.

The report modules used to run this report are in the CULPDICT dictionary; data for the report is taken from the ASFDICT dictionary.
DREPORT 090 - Catalog Summary Report

The Catalog Summary report provides information on all entities defined to the catalog, including their entity types and descriptions. The list is presented in hierarchical order to illustrate the relationships between entities. All catalog entities except passkeys are represented in the dictionary as occurrences of the CATENTRY-160 record. Passkeys are represented as occurrences of the CATPASSKEY-162 record.

Sample DREPORT 090:

```
ENTRY NAME       ENTRY TYPE  TDN        DESCRIPTION

CORP             CATALOG    DATA DICTIONARY CATALOG
                 GROUP      DATABASE ADMINISTRATION GROUP
                 USER
.. CARNE01       USER
.. CULPRITS      DATATABLE  111 CULPRIT Tests out in ProTesT
.. CULL DBA      USER
.. IAQ           USER
.. M3H           USER
.. THRUSH        DATATABLE  120
.. PAGTO01       USER
.. FOLD          FOLDER
.. SP            DATATABLE  184
.. NEWVIEW       DATATABLE  129
.. P             DATATABLE  182
.. S             DATATABLE  183
.. SORTFORM      DATATABLE  186
.. SORTLRGE      DATATABLE  185
.. SORTSMAL      DATATABLE  187
.. SYSTABLE      DATATABLE  113
.. VGA           USER
.. ENK           USER
.. DATASF02      DATATABLE  122
```

Field Descriptions

A description of the fields in the sample report follows:

- **ENTRY NAME**
  Identifies the names of all catalog entities.

- **ENTRY TYPE**
  Identifies the entity as CORP, GROUP, USER, FOLDER, or an object type. Object types can be GRAPH, DATATABLE, MESSAGE, PROCEDURE, MODEL, WORKSHEET, PICTURE, DOCUMENT, GRAPH FORMAT, SYSTEM, DOS, SYNONYM, or OLQ REPORT.
- **TDN**
  Identifies the table definition number of the object.

- **DESCRIPTION**
  Identifies the entity description defined to the catalog.

---

**DREPORT 091 - Catalog Detail Report**

The Catalog Detail report provides information on all entities defined to the catalog, including their entity types, descriptions, and passkey allocations.

**Sample DREPORT 091:**

<table>
<thead>
<tr>
<th>ENTRY NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORP</td>
<td>CATALOG</td>
<td>DATA DICTIONARY CATALOG</td>
</tr>
<tr>
<td></td>
<td>OWNER:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CREATED</td>
<td>mm/dd/yy hh:mm ss.s IDMSDDL</td>
</tr>
<tr>
<td></td>
<td>ACCESSED</td>
<td>mm/dd/yy hh:mm ss.s</td>
</tr>
<tr>
<td></td>
<td>MODIFIED</td>
<td>mm/dd/yy hh:mm ss.s</td>
</tr>
<tr>
<td></td>
<td>REDEFINED</td>
<td>mm/dd/yy hh:mm ss.s</td>
</tr>
<tr>
<td></td>
<td>ALTERED</td>
<td>mm/dd/yy hh:mm ss.s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PASSKEYS GIVEN: BROWSE COPY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TO USER SQA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PASSKEYS GIVEN: ADD BROWSE COPY CREATE ERASE LIST MANAGEMENT MODIFY REDEFINE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TO USER IQA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PASSKEYS GIVEN: ADD BROWSE COPY CREATE ERASE LIST MANAGEMENT MODIFY REDEFINE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TO USER PAGTO01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PASSKEYS GIVEN: BROWSE CREATE MANAGEMENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TO GROUP DBA GROUP</td>
</tr>
<tr>
<td>. DBA GROUP</td>
<td>GROUP</td>
<td>DATABASE ADMINISTRATION GROUP</td>
</tr>
<tr>
<td></td>
<td>OWNER:</td>
<td>CORP</td>
</tr>
<tr>
<td></td>
<td>ACCESS:</td>
<td>FOUNDATION</td>
</tr>
<tr>
<td></td>
<td>CREATED</td>
<td>mm/dd/yy hh:mm ss.s IDMSDDL</td>
</tr>
<tr>
<td></td>
<td>ACCESSED</td>
<td>mm/dd/yy hh:mm ss.s</td>
</tr>
<tr>
<td></td>
<td>MODIFIED</td>
<td>mm/dd/yy hh:mm ss.s</td>
</tr>
<tr>
<td></td>
<td>REDEFINED</td>
<td>mm/dd/yy hh:mm ss.s</td>
</tr>
<tr>
<td></td>
<td>ALTERED</td>
<td>mm/dd/yy hh:mm ss.s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PASSKEYS HELD: MANAGEMENT FOR OBJECT $UNSTRUCTURED-IDB-OBJECT$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PASSKEYS HELD: ADD BROWSE COPY CREATE ERASE LIST MANAGEMENT MODIFY REDEFINE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FOR FOLDER DBA FOLDER</td>
</tr>
<tr>
<td>. . CARNE01</td>
<td>USER</td>
<td>CORP</td>
</tr>
<tr>
<td></td>
<td>AFFIL SIZE: 160 STACK SIZE: 64 DIRECTORY: 2048</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACCESS:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CREATED</td>
<td>mm/dd/yy hh:mm ss.s CULL DBA</td>
</tr>
<tr>
<td></td>
<td>ACCESSED</td>
<td>mm/dd/yy hh:mm ss.s</td>
</tr>
<tr>
<td></td>
<td>MODIFIED</td>
<td>mm/dd/yy hh:mm ss.s</td>
</tr>
<tr>
<td></td>
<td>REDEFINED</td>
<td>mm/dd/yy hh:mm ss.s</td>
</tr>
<tr>
<td></td>
<td>ALTERED</td>
<td>mm/dd/yy hh:mm ss.s</td>
</tr>
<tr>
<td>. . . CULPRITS</td>
<td></td>
<td>DATATABLE CULPRIT Tests out in ProTesT</td>
</tr>
</tbody>
</table>
Field Descriptions

A description of the fields in the sample report follows:

- **ENTRY NAME**
  Identifies the entity names in the catalog.

- **TYPE**
  Identifies the entity as CORP, GROUP, USER, FOLDER, or an object type. Object types can be GRAPH, DATATABLE, MESSAGE, PROCEDURE, MODEL, WORKSHEET, PICTURE, DOCUMENT, GRAPH FORMAT, SYSTEM, DOS, SYNONYM, or OLQ REPORT.

- **DESCRIPTION**
  Identifies the entity description defined to the catalog.

- **OWNER**
  Identifies the name of the owner of each entity.

- **Date/time stamp**
  Information, including the user responsible for the time stamp (IDMSDDDL is displayed for catalog foundation entities):
  - **CREATED**
    Specifies the date the entity was defined to the catalog.
  - **ACCESSED**
    Specifies the date the object entity was last accessed or, for a user entity, the date the user last signed on.
  - **MODIFIED**
    Is currently an unused field.
  - **REDEFINED**
    Is currently an unused field.
  - **ALTERED**
    Specifies the date the catalog definition for the entity was last modified.

- **Passkey information**
  Lists the following:
  - **PASSKEYS GIVEN**
    Identifies the passkeys given to groups or users for access to the listed entity.
  - **PASSKEYS HELD**
    Identifies the passkeys held by the listed entity for access to specific entities.
OBJECT TYPE/ID
Identifies the object type and the definition number of each object entity. For a list of object types, see TYPE above.

ACCESS
Identifies the type of access control:

- FOUNDATION
  Identifies the entity as a member of the catalog foundation; catalog foundation members cannot be deleted or renamed.

- NO DUPLICATES
  (Users and CORP only) Specifies that duplicate names are not allowed for objects and folders owned by the listed entity.

- NO PROPAGATION
  (Users and CORP only) Specifies that ambiguous associations that involve duplicate names are not allowed.

- ACCESS LOCK
  (Users only) Specifies that the user is restricted from signing on.

- ENCRYPT
  Specifies that the user's password is encrypted.

Memory requirements
Identifies memory requirements for each user entity defined to the catalog:

- AFFIL SIZE
  Identifies the number of bytes required to hold the list of groups with which the user is affiliated.

- STACK SIZE
  Identifies the number of bytes required to hold the bill-of-material structure explosion /implosion levels.

- DIRECTORY
  Is currently an unused field.

**DREPORT 092 - Group Detail Report**

The Group Detail report provides information on all groups defined to the catalog.

Sample DREPORT 092:

<table>
<thead>
<tr>
<th>GROUP NAME</th>
<th>OWNER NAME</th>
<th>TIME STAMP</th>
<th>DATE</th>
<th>TIME</th>
<th>USER</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBA GROUP</td>
<td>CORP</td>
<td>CREATED: mm/dd/yy hh:mm ss.s</td>
<td>IDMS00DL</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MODIFIED: mm/dd/yy hh:mm ss.s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>REDEFINED: mm/dd/yy hh:mm ss.s</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Field Descriptions

A description of the fields in the sample report follows:

- **GROUP NAME**
  Identifies the group defined to the catalog.

- **OWNER NAME**
  Identifies CORP as the owner of each group.

- **TIME STAMP: DATE/TIME**
  Identifies the date and time stamps for the group:
  
  - **CREATED**
    Specifies the date the group definition was defined to the catalog.
  
  - **ACCESSED**
    Is used for object and user entities only.
  
  - **MODIFIED**
    Is currently an unused field.
  
  - **REDEFINED**
    Is currently an unused field.
  
  - **ALTERED**
    Specifies the date the catalog definition for the group was last modified.

- **USER**
  Identifies the user responsible for the time stamp. (IDMSDDDL is displayed for DBA GROUP.)

- **DESCRIPTION**
  Displays the group description defined to the catalog.

- **ACCESS**
  Indicates whether the group is a member of the catalog foundation (FOUNDATION). Catalog foundation members cannot be renamed or deleted.
DREPORT 093 - User Detail Report

The User Detail report provides information on all user occurrences defined to the catalog, including date/time stamps for the user and the amount of memory required by the user.

Sample DREPORT 093:

<table>
<thead>
<tr>
<th>USER NAME</th>
<th>OWNER NAME</th>
<th>TIME STAMP:</th>
<th>DATE</th>
<th>TIME</th>
<th>USER</th>
</tr>
</thead>
<tbody>
<tr>
<td>CARNE01</td>
<td>CORP</td>
<td>CREATED</td>
<td>2048</td>
<td></td>
<td>CULL DBA</td>
</tr>
<tr>
<td>AFFIL SIZE: 160 STACK SIZE: 64 DIRECTORY: 2048</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION:</td>
<td>ACCESS:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CULL DBA</td>
<td>CORP</td>
<td>CREATED</td>
<td>2048</td>
<td></td>
<td>IDMSDODL</td>
</tr>
<tr>
<td>AFFIL SIZE: 32 STACK SIZE: 128 DIRECTORY: 2048</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION: INITI HALE GUB GROUP USER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACCESS:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENK</td>
<td>CORP</td>
<td>CREATED</td>
<td>2048</td>
<td></td>
<td>IDBCAT</td>
</tr>
<tr>
<td>AFFIL SIZE: 160 STACK SIZE: 64 DIRECTORY: 2048</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION:</td>
<td>ACCESS:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HANELO1</td>
<td>CORP</td>
<td>CREATED</td>
<td>2048</td>
<td></td>
<td>IDBCAT</td>
</tr>
<tr>
<td>AFFIL SIZE: 160 STACK SIZE: 64 DIRECTORY: 2048</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION:</td>
<td>ACCESS:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDBSYSTEM</td>
<td>CORP</td>
<td>CREATED</td>
<td>2048</td>
<td></td>
<td>IDBCAT</td>
</tr>
<tr>
<td>AFFIL SIZE: 160 STACK SIZE: 64 DIRECTORY: 2048</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION:</td>
<td>ACCESS:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Descriptions

A description of the fields in the sample report follows:

- **USER NAME**
  Identifies the users defined to the catalog.

- **OWNER NAME**
  Identifies CORP as the owner of each user.
• **TIME STAMP: DATE/TIME**
  Identifies the date and time stamps for the user:

  - **CREATED**
    Specifies the date the user was defined to the catalog.

  - **SIGNED ON**
    Specifies the date the user last signed on to ASF or IDB.

  - **MODIFIED**
    Is currently an unused field.

  - **REDEFINED**
    Is currently an unused field.

  - **ALTERED**
    Specifies the date the catalog definition for the user was last modified.

• **USER**
  Identifies the user responsible for the time stamp (IDMSDDDL is displayed for CULL DBA).

• **Memory requirements**
  Identifies memory requirements for the user:

  - **AFFIL SIZE**
    Indicates the number of bytes required to hold the list of groups with which the user is affiliated.

  - **STACK SIZE**
    Indicates the number of bytes required to hold the bill-of-material structure explosion /implosion levels.

  - **DIRECTORY**
    Is currently an unused field.

• **DESCRIPTION**
  Displays the user description defined to the catalog.

• **ACCESS**
  Identifies the access control specified for the user:

  - **FOUNDATION**
    Identifies the user as a member of the catalog foundation; catalog foundation members cannot be deleted or renamed.

  - **NO DUPLICATES**
    Indicates that duplicate names are not allowed for objects and folders owned by the user.

  - **NO PROPAGATION**
    Indicates that ambiguous associations involving duplicate names are not allowed.
- **ACCESS LOCK**
  Indicates that the user is restricted from signing on.

- **ENCRIPT**
  Specifies that the user's password is encrypted.

---

**DREPORT 094 - Folder Detail Report**

The Folder Detail report provides information on all folders defined to the catalog.

**Sample DREPORT 094:**

<table>
<thead>
<tr>
<th>REPORT NO. 94</th>
<th>DATA DICTIONARY REPORTER</th>
<th>REL mm.n</th>
<th>FOLDER DETAIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DREPORT 094</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FOLDER NAME</th>
<th>OWNER NAME</th>
<th>TIME STAMP: DATE/TIME</th>
<th>DESCRIPTION</th>
<th>ACCESS:</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBA FOLDER</td>
<td>IDBSYSTEM</td>
<td>CREATED: mm/dd/yy hh:mm ss.s</td>
<td>IDBCAT</td>
<td>FOUNDATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACCESSED: mm/dd/yy hh:mm ss.s</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MODIFIED: mm/dd/yy hh:mm ss.s</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>REDEFINED: mm/dd/yy hh:mm ss.s</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALTERED: mm/dd/yy hh:mm ss.s</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field Descriptions**

A description of the fields in the sample report follows:

- **FOLDER NAME**
  Identifies the folders defined to the catalog.

- **OWNER NAME**
  Identifies the owner of each folder.

- **TIME STAMP: DATE/TIME**
  Identifies date and time stamps for the folder:

- **CREATED**
  Specifies the date the folder definition was created in the catalog.
- **ACCESSED**
  Is used for object and user entities only.

- **MODIFIED**
  Is currently an unused field.

- **REDEFINED**
  Is currently an unused field.

- **ALTERED**
  Specifies the date the catalog definition for the folder was last modified.

- **USER**
  Identifies the user responsible for the time stamp.

- **DESCRIPTION**
  Displays the folder description defined to the catalog.

- **ACCESS**
  Indicates whether the folder is a member of the catalog foundation (FOUNDATION). Catalog foundation members cannot be deleted or renamed.

---

**DREPORT 095 - Object Detail Report**

The Object Detail report provides information on all objects defined to the catalog.

Sample DREPORT 095:

<table>
<thead>
<tr>
<th>OBJECT NAME</th>
<th>OWNER NAME</th>
<th>TIME STAMP:</th>
<th>DATE</th>
<th>TIME</th>
<th>USER</th>
</tr>
</thead>
<tbody>
<tr>
<td>$OBJECT-SECURITY-NAME-TABLE$</td>
<td>IDBSYSTEM</td>
<td>CREATED: mm/dd/yy hh:mm:ss.s</td>
<td>CULL DBA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OBJECT TYPE: DATATABLE</td>
<td>ID: 21</td>
<td>ACCESS:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODIFIED: mm/dd/yy hh:mm:ss.s</td>
<td>CULL DBA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REDefined: mm/dd/yy hh:mm:ss.s</td>
<td>CULL DBA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALTERED: mm/dd/yy hh:mm:ss.s</td>
<td>CULL DBA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION:</td>
<td>ACCESS:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| $SECURITY-RUNTIME-TABLE$ | IDBSYSTEM | CREATED: mm/dd/yy hh:mm:ss.s | CULL DBA |
| OBJECT TYPE: DATATABLE | ID: 20 | ACCESS: | | | |
| MODIFIED: mm/dd/yy hh:mm:ss.s | CULL DBA |
| REDefined: mm/dd/yy hh:mm:ss.s | CULL DBA |
| ALTERED: mm/dd/yy hh:mm:ss.s | CULL DBA |
| DESCRIPTION: | ACCESS: |

| $SRT-OST-CROSS-REFERENCE$ | IDBSYSTEM | CREATED: mm/dd/yy hh:mm:ss.s | CULL DBA |
| OBJECT TYPE: DATATABLE | ID: 22 | ACCESS: | | | |
| MODIFIED: mm/dd/yy hh:mm:ss.s | CULL DBA |
| REDefined: mm/dd/yy hh:mm:ss.s | CULL DBA |
| ALTERED: mm/dd/yy hh:mm:ss.s | CULL DBA |
| DESCRIPTION: | ACCESS: |

| $UNSTRUCTURED-IDB-OBJECT$ | IDBSYSTEM | CREATED: mm/dd/yy hh:mm:ss.s | IDBCAT |
| OBJECT TYPE: DATATABLE | ID: 101 | ACCESS: | | | |
| MODIFIED: mm/dd/yy hh:mm:ss.s | MJH |
| REDefined: mm/dd/yy hh:mm:ss.s | PAGTO01 |
| ALTERED: mm/dd/yy hh:mm:ss.s | |
DESCRIPTION:
ACCESS:

CULPRITS CARNE01

OBJECT TYPE: DATATABLE ID: 111

DESCRIPTION: CULPRIT Tests out in ProTesT
ACCESS:

Field Descriptions

A description of the fields in the sample report follows:

- **OBJECT NAME**
  Identifies the objects in the catalog.

- **OWNER NAME**
  Identifies the owner of each object.

- **TIME STAMP: DATE/TIME**
  Identifies date and time stamps for the object:

  - **CREATED**
    Specifies the date the object definition was defined to the catalog.

  - **ACCESSED**
    Specifies the date the object was last accessed.

  - **MODIFIED**
    Is currently an unused field.

  - **REDEFINED**
    Is currently an unused field.

  - **ALTERED**
    Specifies the date the catalog definition for the object was last modified.

- **USER**
  Identifies the user responsible for the time stamp.

- **OBJECT TYPE**
  Identifies the type of object being described as GRAPH, DATATABLE, MESSAGE, PROCEDURE, MODEL, WORKSHEET, PICTURE, DOCUMENT, GRAPH FORMAT, SYSTEM, DOS, SYNONYM, or OLQ REPORT.

- **ID**
  Identifies the table definition number.

- **DESCRIPTION**
  Displays the object description defined to the catalog.
ACCESS
Indicates whether the object is a member of the catalog foundation (FOUNDATION). Catalog foundation members cannot be renamed or deleted.

DREPORT 096 - Catalog Summary Key Report

The Catalog Summary Key report provides information on selected entities defined to the catalog. The selection is based on a key, which consists of an owner name and entity name. Any catalog entity except the passkey entity can be used as the key.

The following figure shows sample output for DREPORT 096. The DREPORT and KEY parameters used to create the sample report are:

DREPORT=096
KEY CAT-ENTRY-CALCKEY-160
* MJH BIRDS

Sample DREPORT 096:

<table>
<thead>
<tr>
<th>ENTRY NAME</th>
<th>ENTRY TYPE</th>
<th>TDN</th>
<th>DESCRIPTION</th>
<th>OWNER</th>
</tr>
</thead>
<tbody>
<tr>
<td>MJH</td>
<td>USER</td>
<td></td>
<td></td>
<td>CORP</td>
</tr>
<tr>
<td>THRUSH</td>
<td>DATATABLE</td>
<td>120</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Descriptions

A description of the fields in the sample report follows:

- **ENTRY NAME**
  Identifies the entities being described.

- **ENTRY TYPE**
  Identifies the entity as CORP, GROUP, USER, FOLDER, or an object type. Object types can be GRAPH, DATATABLE, MESSAGE, PROCEDURE, MODEL, WORKSHEET, PICTURE, DOCUMENT, GRAPH FORMAT, SYSTEM, DOS, or SYNONYM.

- **TDN**
  Identifies the table definition number of the object.

- **DESCRIPTION**
  Identifies the entity description defined to the catalog.

- **OWNER**
  Identifies the owner of the catalog entity.
DREPORT 097 - Catalog Detail Key Report

The Catalog Detail Key report provides detailed information on selected entities, including their entity types, descriptions, and passkey allocations. The selection is made based on a key, which includes an owner name and member name. Any catalog entity except the passkey entity can be used as the key.

The following figure shows sample output for DREPORT 097. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=097
KEY CAT-ENTRY-CALCKEY-160
*C'CORP CARNE01'

Sample DREPORT 097:

```
REPORT NO. 97                      DATA DICTIONARY REPORTER  REL nn.n                      mm/dd/yy          PAGE   1
DREPORT 097                      CATALOG DETAIL
ENTRY NAME                        TYPE    DESCRIPTION
CARNE01                           USER
  OWNER:  CORP
  AFFIL SIZE:  160  STACK SIZE:  64  DIRECTORY:  2048
  ACCESS:  FOUNDATION NO DUPLICATES NO PROPAGATION ACCESS LOCK ENCRYPT
            CREATED  mm/dd/yy  hh:mm ss.s  CULL DBA
            ACCESSED  mm/dd/yy  hh:mm ss.s
            MODIFIED  mm/dd/yy  hh:mm ss.s
            REDefined  mm/dd/yy  hh:mm ss.s
            ALTERED   mm/dd/yy  hh:mm ss.s
  . CULPRITS
    DATATABLE  CULPRIT Tests out in ProTesT
    OWNER:  CARNE01
    OBJECT TYPE: DATATABLE   ID:    111
    ACCESS:
            CREATED  mm/dd/yy  hh:mm ss.s  NBC
            ACCESSED  mm/dd/yy  hh:mm ss.s  CARNE01
            MODIFIED  mm/dd/yy  hh:mm ss.s
            REDefined  mm/dd/yy  hh:mm ss.s  NBC
            ALTERED   mm/dd/yy  hh:mm ss.s  NBC
```

Field Descriptions

A description of the fields in the sample report follows:

- **ENTRY NAME**
  Identifies the entity being described.

- **TYPE**
  Identifies the entity as CORP, GROUP, USER, FOLDER, or an object type. Object types can be GRAPH, DATATABLE, MESSAGE, PROCEDURE, MODEL, WORKSHEET, PICTURE, DOCUMENT, GRAPH FORMAT, SYSTEM, DOS, SYNONYM, or OLQ REPORT.

- **DESCRIPTION**
  Displays the entity description defined to the catalog.

- **OWNER**
  Identifies the owner of each entity.
- **Memory requirements**
  Identifies memory requirements for the user:
  
  - **AFFIL SIZE**
    Indicates the number of bytes required to hold the list of groups with which the user is affiliated.
  
  - **STACK SIZE**
    Indicates the number of bytes required to hold the bill-of-material structure explosion /implosion levels.
  
  - **DIRECTORY**
    Is currently an unused field.

- **ACCESS**
  Indicates the access control specified for the entity:
  
  - **FOUNDATION**
    Identifies the entity as a member of the catalog foundation; members of the catalog foundation cannot be renamed or deleted.
  
  - **NO DUPLICATES**
    (Users and CORP only) Indicates that duplicate names are not allowed for objects and folders owned by the listed entity.
  
  - **NO PROPAGATION**
    (Users and CORP only) Indicates that ambiguous associations involving duplicate names are not allowed.
  
  - **ACCESS LOCK**
    (Users only) Indicates that the user is restricted from signing on.
  
  - **ENCRYPT**
    Specifies that the user's password is encrypted.

- **Date/time stamp**
  Information is listed, including the user responsible for the time stamp (IDMSDDDL is displayed for catalog entities):
  
  - **CREATED**
    Specifies the date the object was defined to the catalog.
  
  - **ACCESSED**
    Specifies the date the object was last accessed.
  
  - **MODIFIED**
    Is currently an unused field.
  
  - **REDEFINED**
    Is currently an unused field.
  
  - **ALTERED**
    Specifies the date the catalog definition for the entity was last modified.
Passkey information
Lists the following:

- **PASSKEYS GIVEN**
  Identifies the passkeys given to groups or users for access to the listed entity.

- **PASSKEYS HELD**
  Identifies the passkeys held by the listed entity for access to specific entities.
CA IDMS/DB Journal Reports -- JREPORTS

A journal report describes the contents of the journal file. The following table lists the journal reports in order by report module number.

<table>
<thead>
<tr>
<th>JREPORT Module</th>
<th>Report Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>No listing (contains journal record field descriptions)</td>
</tr>
<tr>
<td>001</td>
<td>Transaction Summary</td>
</tr>
<tr>
<td>002</td>
<td>Program Termination Statistics</td>
</tr>
<tr>
<td>003</td>
<td>Program I/O Statistics</td>
</tr>
<tr>
<td>004</td>
<td>Program Summary</td>
</tr>
<tr>
<td>005</td>
<td>Detail Area/Transaction</td>
</tr>
<tr>
<td>006</td>
<td>Detail Program/Area</td>
</tr>
<tr>
<td>007</td>
<td>Area Summary</td>
</tr>
<tr>
<td>008</td>
<td>Formatted Record Dump</td>
</tr>
<tr>
<td>009</td>
<td>User ID</td>
</tr>
<tr>
<td>010</td>
<td>External User Identity</td>
</tr>
<tr>
<td>011</td>
<td>Count By Journal Record Type</td>
</tr>
</tbody>
</table>

This section discusses types of journal records, report-specific input parameters required to process a journal report, and sample output. For more information on journaling, see the CA IDMS Database Administering section.

See the following topics for more information:
- Uses for Journal Reports (see page 62)
- Types of Journal Records (see page 62)
- Summary of Records Required for Journal Reports (see page 71)
- Producing Journal Reports (see page 72)
- JREPORT 001 - Transaction Summary (see page 75)
- JREPORT 002 - Program Termination Statistics (see page 77)
- JREPORT 003 - Program I/O Statistics (see page 79)
- JREPORT 004 - Program Summary (see page 81)
- JREPORT 005 - Detail Area/Transaction (see page 83)
- JREPORT 006 - Detail Program/Area (see page 85)
- JREPORT 007 - Area Summary (see page 86)
- JREPORT 008 - Formatted Record Dump (see page 88)
- JREPORT 009 - User ID (see page 93)
- JREPORT 010 - External User Identity (see page 95)
- JREPORT 011 - Count By Journal Record Type Report (see page 96)
- JREPORTS z/OS JCL (see page 97)
Uses for Journal Reports

Journal reports produce statistics that can be used to monitor CA IDMS/DB database and DC/UCF system performance. As a DBA, journal statistics allow you to monitor the following performance features for each program that accesses the database:

- **Database page access statistics**, such as the number of pages written, read, and requested by each program and the number of records requested and made current of the transaction.
- **CALC and VIA overflow statistics**, such as the number of CALC records stored on pages other than the target page.
- **Variable-length record statistics**, such as the number of variable-length record fragments stored on pages other than the target page.
- **Area usage statistics**, such as the name and usage modes of areas accessed by each application program.

Reports Help You Monitor Performance

By running journal reports frequently, a DBA can monitor trends in database and system performance.

Other Reporting Facilities

Other tools available to the DBA are the Database Analysis Utility (IDMSDBAN) and the BACKUP utility.

**Note:** For more information on these utilities, see Other CA IDMS Reporting Facilities (see page 659).

Types of Journal Records

Journal reports use the following journal record types:

<table>
<thead>
<tr>
<th>Record type</th>
<th>Description</th>
</tr>
</thead>
</table>
| TIME        | Time record
  - TIME records the date and time the contents of the journal buffer are written to the journal file. A TIME record is created each time a journal buffer is initialized; however, the date and time fields contain binary zeros until the contents of the journal buffer are written to the journal file. |
| BGIN        | Checkpoint
  - BGIN checkpoints mark the beginning of local work done by a transaction branch. They are written to the journal file when a database transaction is initiated if JOURNAL RETRIEVAL is
<table>
<thead>
<tr>
<th>Record type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>specified in the system definition or when the first update occurs otherwise. Starting with r16 SP4, for compliance and audit reporting, the BGIN checkpoint record contains the user ID of the user signed on who executed the application causing the BGIN to be written. Starting with IDMS Server r16.1 (or IDMS r16 SP6), the BGIN checkpoint record contains the external identity from a multitiered application such as a web application.</td>
<td></td>
</tr>
<tr>
<td>AREA</td>
<td>Checkpoint</td>
</tr>
<tr>
<td>AREA checkpoints record transaction access to a database area. One AREA checkpoint is written to the journal file for each area readied by an explicit DML READY statement or readied automatically by the DBMS. AREA checkpoints are written to the journal file as follows:</td>
<td></td>
</tr>
<tr>
<td>Under the central version, AREA checkpoints are written to the journal file at the time of the first functional call issued by the application program.</td>
<td></td>
</tr>
<tr>
<td>In local mode, AREA checkpoints for areas readied in update mode are written to the journal file as each READY statement is processed. AREA checkpoints for areas readied in retrieval mode are written to the journal file at the time of the first functional call.</td>
<td></td>
</tr>
<tr>
<td>COMT</td>
<td>Checkpoint</td>
</tr>
<tr>
<td>COMT checkpoints are written during a commit operation to mark the successful completion of a transaction branch. A COMT checkpoint is similar to an ENDJ checkpoint except that it enables work done after the commit to be recorded on the journal file under the same local identifier (LID). It is only written in response to a COMMIT or COMMIT WORK CONTINUE statement and then only if ON COMMIT WRITE COMT is specified in the system definition.</td>
<td></td>
</tr>
<tr>
<td>ENDJ</td>
<td>Checkpoint</td>
</tr>
<tr>
<td>ENDJ checkpoints are written during a commit operation to mark the successful completion of a transaction branch.</td>
<td></td>
</tr>
<tr>
<td>RTSV</td>
<td>Checkpoint</td>
</tr>
<tr>
<td>RTSV checkpoints mark an SQL statement whose updates must be rolled out. An RTSV checkpoint is written to the journal file whenever an error is encountered while processing an SQL statement and that statement has updated the database.</td>
<td></td>
</tr>
<tr>
<td>ABRT</td>
<td>Checkpoint</td>
</tr>
<tr>
<td>ABRT checkpoints are written during a backout operation to mark the abnormal completion of a transaction branch. If running under the central version, ABRT checkpoints are written only after CA IDMS/DB automatically recovers a failing transaction.</td>
<td></td>
</tr>
<tr>
<td>BFOR</td>
<td>Journal record entry</td>
</tr>
<tr>
<td>BFOR entries record the before image of a database record. A BFOR entry is created when a CA IDMS/DB application program issues a request to update information in the database. When a new record is stored, the BFOR entry contains a null before image, indicating the previous absence of the record in the database. When a BFOR entry contains a null before image, the record-image-length field in the entry is set to zero.</td>
<td></td>
</tr>
<tr>
<td>AFTR</td>
<td>Journal record entry</td>
</tr>
<tr>
<td>AFTR entries record the after image of a database record. An AFTR entry is created when a CA IDMS/DB application program issues a request to update information in the database. When an existing record is removed, the AFTR entry contains a null after image, indicating the deletion of the record from the database. When an AFTR entry contains a null after image, the record-image-length field in the entry is set to zero.</td>
<td></td>
</tr>
<tr>
<td>DIND</td>
<td>Distributed checkpoint</td>
</tr>
</tbody>
</table>
| (In doubt) DIND entries are written by a two-phase commit participant after it has successfully prepared its resources for commit and prior to returning an OK response to its
The DIND entry contains the Local Transaction Identifiers (LIDs) identifying the work done by the local transaction branches that participated in the distributed transaction. It also contains information on a participant's coordinator and about a coordinator's participants. The specific information that is recorded varies depending on the type of the coordinator or participant.

DCOM Distributed checkpoint
(Commit) DCOM entries are written by a two-phase commit coordinator to signify that a distributed transaction's changes will be committed. Its existence demarcates the first and second phases of the commit process. A participant also writes a DCOM entry immediately upon receiving a Commit request from its coordinator. The DCOM entry contains the Local Transaction Identifiers (LIDs) identifying the work done by the local transaction branches that participated in the distributed transaction. It also contains information on a participant's coordinator and about a coordinator's participants. The specific information that is recorded varies depending on the type of the coordinator or participant.

DBAK Distributed checkpoint
(Backout) DBAK entries are written by a two-phase commit coordinator to signify that a transaction's changes will be backed out. Its existence demarcates the first and second phases of the commit process. A participant also writes a DBAK entry immediately upon receiving a Backout request from its coordinator but only if a DIND had previously been written. The DBAK entry contains the Local Transaction Identifiers (LIDs) identifying the work done by the local transaction branches that participated in the distributed transaction. It also contains information on a participant's coordinator and about a coordinator's participants. The specific information that is recorded varies depending on the type of the coordinator or participant.

DPND Distributed checkpoint
(Pending) DPND entries are written by a two-phase commit coordinator during the second phase of a commit operation if a participant is unable to complete its commit processing due to a failure. By writing this entry, the coordinator is able to forget some participants while remembering others. It is written by a participant if it is forced to heuristically complete its portion of a distributed transaction. The DPND entry contains information on a participant's coordinator and about a coordinator's participants. The specific information that is recorded varies depending on the type of the coordinator or participant.

DFGT Distributed checkpoint
(Forget) DFGT entries are written by two-phase commit coordinators and participants when they have completed their two-phase commit processing for a distributed transaction. A DFGT entry is written only if some other distributed checkpoint entry was previously written for the transaction.

Relating Local and Distributed Journal Entries

BGIN, COMT, ENDJ, and ABRT checkpoints and BFOR and AFTR journal entries log work done by a transaction branch within the local system. They contain a 4-byte local identifier (LID) that uniquely identifies this work. In order to associate work done locally with a distributed transaction, DIND, DCOM, and DBAK checkpoints contain a list of LID values representing the local work units that are part of the distributed work unit.

The following illustrates the sequence in which local and distributed journal records can be written to a journal file for a distributed transaction:
- **BGIN** - indicates the start of work done locally
- **BFOR/AFTR** - one or more pairs
- **DIND** - on a participant only
- **DCOM or DBAK** - on a participant and a coordinator
- **COMT or ENDJ** - if a DCOM was written
- **ABRT** - if a DBAK was written
- **DPND** - on a coordinator if the commit operation was interrupted; on a participant if the transaction was heuristically completed
- **DFGT** - on a participant and a coordinator if any other distributed checkpoint was written

**Special Considerations for BFOR and AFTR Entries**

- BFOR and AFTR entries are also written when CA IDMS/DB brings relocated records back to the home page or physically deletes logically deleted records. These actions can occur during any operation; therefore, BFOR and AFTR entries can appear for retrieval-only transactions that have readied the applicable area in update mode.

- Corresponding BFOR and AFTR entries are not necessarily contiguous in the journal file. For example, the BFOR record for one transaction may be immediately followed by an AFTR record for a different transaction. The BFOR and AFTR entries for a database record contain either the whole record or a portion of the record, as follows:
  - If the data portion of the record has been changed, the journal record entries contain the entire database record (that is, the dbkey, prefix, and data portions).
  - If only pointers in the record prefix have been changed, only the dbkey and prefix portions of the record are written to the journal file.

- BFOR and AFTR entries can also span journal blocks. A record that spans a journal block appears as two records with the second portion offset by the length of the first portion. For example, if 200 bytes remain in a journal block, a 500-byte BFOR record is journaled as one 200-byte and one 300-byte BFOR journal entry; the 300-byte entry has an offset of 200.

**Special Considerations for Distributed Checkpoint Entries**

- Distributed Checkpoint entries (types DIND, DCOM, DBAK, DPND, and DFGT) can be larger than a single disk journal block. If this is the case, they are split into as many journal blocks as necessary to hold the entire record.

- Distributed Checkpoint entries can also be split across disk journal files and, hence, across archive files.

The following figures show the record layouts for the journal record types:
<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Length (in bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Journal record length</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Reserved (initialized to zeroes)</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Journal record type (TIME)</td>
<td>alphanumeric</td>
<td>4</td>
</tr>
<tr>
<td>UTC datetime stamp</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>Journal record sequence number</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>Journal record qualifier</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>GV number (central version only)</td>
<td>binary</td>
<td>1</td>
</tr>
<tr>
<td>Journaling version</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Reserved</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>datetime stamp</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Journal record length</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Reserved (initialized to zeroes)</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
</tbody>
</table>

**IDMSDB--Types of Journal Records**

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Length (in bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Journal record length</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Reserved (initialized to zeroes)</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Journal record type (BGIN)</td>
<td>alphanumeric</td>
<td>4</td>
</tr>
<tr>
<td>UTC datetime stamp</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>Journal record sequence number</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>Journal record qualifier</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>GV number (central version only)</td>
<td>binary</td>
<td>1</td>
</tr>
<tr>
<td>Journaling version</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Local transaction ID (LID)</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Address of VSB</td>
<td>address</td>
<td>4</td>
</tr>
<tr>
<td>Program name</td>
<td>alphanumeric</td>
<td>8</td>
</tr>
<tr>
<td>datetime stamp</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>Reserved (initialized to zeroes)</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>Transaction update quiesce level</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Transaction quiesce level</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Local task ID for transaction or savepoint sequence number</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>Journal record length</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Reserved (initialized to zeroes)</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
</tbody>
</table>

**IDMSDB--Types of Journal Records (2)**
### AREA JOURNAL RECORD TYPE

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Length (in bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Journal record length</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Reserved (initialized to zeros)</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Journal record type (AREA)</td>
<td>alphanumeric</td>
<td>4</td>
</tr>
<tr>
<td>UTC date/time stamp</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>Journal record sequence number</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>Journal record qualifier</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>CV number (central version only)</td>
<td>binary</td>
<td>1</td>
</tr>
<tr>
<td>Journaled version</td>
<td>binary</td>
<td>1</td>
</tr>
<tr>
<td>Reserved</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Local transaction ID (LID)</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Area type</td>
<td>binary</td>
<td>1</td>
</tr>
<tr>
<td>DBMS verb number</td>
<td>binary</td>
<td>1</td>
</tr>
<tr>
<td>Schema page group identifier</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Area name</td>
<td>alphanumeric</td>
<td>18</td>
</tr>
<tr>
<td>Area flag byte</td>
<td>binary</td>
<td>1</td>
</tr>
<tr>
<td>Reserved</td>
<td>binary</td>
<td>1</td>
</tr>
<tr>
<td>Low page of area</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>High page of area</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>*OPEN mode</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>**OPEN access</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Journal record length</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Reserved (initialized to zeros)</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
</tbody>
</table>

### BFOR and AFTR JOURNAL RECORD TYPE

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Length (in bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Journal record length</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Reserved (initialized to zeros)</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Journal record type (BFOR or AFTR)</td>
<td>alphanumeric</td>
<td>4</td>
</tr>
<tr>
<td>UTC date/time stamp</td>
<td>binary</td>
<td>8 (2 fullwords)</td>
</tr>
<tr>
<td>Journal record sequence number</td>
<td>binary</td>
<td>8 (2 fullwords)</td>
</tr>
<tr>
<td>Journal record qualifier</td>
<td>binary</td>
<td>8 (2 fullwords)</td>
</tr>
<tr>
<td>CV number (central version only)</td>
<td>binary</td>
<td>1</td>
</tr>
<tr>
<td>Journaled version</td>
<td>binary</td>
<td>1</td>
</tr>
<tr>
<td>Reserved</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Local transaction ID (LID)</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Area type</td>
<td>binary</td>
<td>1</td>
</tr>
<tr>
<td>DBMS verb number</td>
<td>binary</td>
<td>1</td>
</tr>
<tr>
<td>Schema page group identifier</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Schema DBKey format word</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Database key</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Record ID</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Page displacement</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Record image length</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Prefix length</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Offset into DBMS of call</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Spanned record offset</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>User record (prefix and data)</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Journal record length</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Reserved (initialized to zeros)</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
</tbody>
</table>
### COMT, ENDJ, ABRT, RTSV, Journal Record Type

<table>
<thead>
<tr>
<th>Field</th>
<th>Data Type</th>
<th>Length (in bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Journal record length</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Reserved (initialized to zeroes)</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Journal record type (BGIN)</td>
<td>alphanumeric</td>
<td>4</td>
</tr>
<tr>
<td>UTC datetime stamp</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>Journal record sequence number</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>Journal record qualifier</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>CV number (central version only)</td>
<td>binary</td>
<td>1</td>
</tr>
<tr>
<td>Journaling version</td>
<td>binary</td>
<td>1</td>
</tr>
<tr>
<td>Reserved</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Local transaction ID (LID)</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Address of VB</td>
<td>address</td>
<td>4</td>
</tr>
<tr>
<td>Program name</td>
<td>alphanumeric</td>
<td>8</td>
</tr>
<tr>
<td>Datetime stamp</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>Reserved (initialized to zeroes)</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>Transaction update quiesce level</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Transaction quiesce level</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Local task ID for transaction or savepoint sequence number</td>
<td>alphanumeric</td>
<td>8</td>
</tr>
<tr>
<td>Number of user records updated</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of pages read from database</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of pages written to database</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Total pages requested by DBMS</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of times calc record stored on target page</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of times calc record not stored on target page</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of times via record stored on target page</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of times via record not stored on target page</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of records requested from DBMS</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of current-of-transaction records</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of calls to DBMS</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of record fragments stored</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of record fragments returned to home page</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>+Total locks acquired</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of shared DB-key locks held</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>*Number of non-shared DB-key locks held</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Transaction ID</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Task ID</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Local task ID for transaction</td>
<td>binary</td>
<td>8 (2 Fullwords)</td>
</tr>
<tr>
<td>Number of SRS splits</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of SRS spawms</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of SRS records stored</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of SRS records erased</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of SR7 records stored</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of SR7 records erased</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of B-Tree searches</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of B-Tree levels searched</td>
<td>binary</td>
<td>4 (Fullword)</td>
</tr>
<tr>
<td>Number of orphaned adopted</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>#index levels searched - best case</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>#index levels searched - worst case</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Journal record length</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
<tr>
<td>Reserved (initialized to zeroes)</td>
<td>binary</td>
<td>2 (Halfword)</td>
</tr>
</tbody>
</table>
IDMSDB--Types of Journal Records (6)

Field Definitions for JREPORT 000

The journal record field definitions as they appear in the JREPORT 000 parameter module appear as follows. The parameters show the actual names (and synonyms) of the journal record fields.

```
REC$0**** 'J' REPORTS IDMS JOURNAL FILE volser REC PARAMETERS
REC REC-LEN 1 2 1
REC TYPE 5 4
REC GMT 9 8
REC SEQ 17 8 1
REC SEQ-A 17 8 $IDMS/2882 HEX COMPARE
REC QUAL 25 8 1 $qualifier:
REC QUAL-CV 25 8 $CV: node
REC QUAL-LOCAL 25 8 1 $Local: GMT
REC CVNO 33 1
REC VERS 34 1
REC RSV1 35 2 $Reserved
REC $ $BGIN,COMT,ENDJ,ABRT,RTSV, Plus AREA,BFOR,AFTRECORDS ********
REC TRANACT-ID 37 4 1
REC TRANACT-IDC 37 4
REC $ $BGIN,COMT,ENDJ,ABRT,RTSV, Plus AREA RECORDS **************
REC VIB 41 4 1
REC $ $BGIN,COMT,ENDJ,ABRT,RTSV RECORDS**************************
REC PROGRAM-NAME 45 8
REC DTESTAMP 53 8
REC DTESTAMPX 53 8 1
REC OLDTIM 61 8 2 $not used
REC UPDATE-QUIESCE 69 2 1
REC QUIESCE 71 2 1
REC SAVEPOINT-SEQ 73 8 1 $RTSV only.
REC TASK-ID 73 8 $Local Task id.
REC LT-P1 73 4
REC LT-P2 77 4 1
REC $ $COMT,ENDJ,ABRT RECORDS ONLY ******************************
REC UPDATED 81 4 1
REC DB-STATS 85 6 1
REC READ 85 4 1
REC WRITTEN 89 4 1
REC PAGE-REQUESTS 93 4 1
REC CALC-ON 97 4 1
```
REC CALC-NOT-ON 101 4 1
REC VIA-ON 105 4 1
REC VIA-NOT-ON 109 4 1
REC REC-REQUESTED 113 4 1
REC CURR-OF-TR 117 4 1
REC CALLS 121 4 1
REC FGMT-STORED 125 4 1
REC FGMT-RETURNED 129 4 1
REC LOCKS-REQUESTED 133 4 1
REC SHARED-LOCKS-HELD 137 4 1
REC EXCLUSIVE-LOCKS-HELD 141 4 1
REC TR-ID-CV 145 4 1 $TRANSACTION ID.
REC TASK-ID-CV 149 4 1 $TCE address.
REC LOCAL-TASK-ID-CV 153 8
REC LT-PART1 153 4
REC LT-PART2 157 4 1
REC LT-PART2A 157 3 1 $IDMS/2549
REC LT-PART2B 160 1 1 $IDMS/2549
REC IX-STATS 161 40
REC IX-SR8-SPLITS 161 4 1
REC IX-SR8-SPAWNS 165 4 1
REC IX-SR8-STORED 169 4 1
REC IX-SR8-ERASED 173 4 1
REC IX-SR7-STORED 177 4 1
REC IX-SR7-ERASED 181 4 1
REC IX-BTREE-SEARCHES 185 4 1
REC IX-BTREE-LEVELS-SEARCHED 189 4 1
REC IX-ORPHANS-ADOPTED 193 4 1
REC IX-LEVELS-SEARCHED-BEST 197 2 1
REC IX-LEVELS-SEARCHED-WORST 199 2 1
REC $ TIME RECORD ONLY *********************************************
REC T-DATESTMP 37 8 1
REC $ BEGIN RECORD ONLY *********************************************
REC USER-ID 81 32
REC EXTERNAL-ID 113 32
REC $ AREA RECORD ONLY *********************************************
REC AREA 45 18
REC RESERVED2 63 2
REC LOW-PAGE 65 4 1
REC HIGH-PAGE 69 4 1
REC MODE 73 2 1
REC ACCESS 75 2 1
REC $ BFOR & AFTR RECORDS ONLY *************************************
REC IDMS-VSAM-FLAG 41 1 1
REC AREA-TYPE 41 1 1
REC VERB-NUM 42 1 1
REC VERB 42 1 1
REC PAGE-GROUP 43 2 1
REC DBK-FORMAT 45 4 1
REC DBK-LINES 48 1 1
REC DB-KEY 49 4 1
REC DB-KEY1 49 3 1 $IDMS/1209
REC DB-KEY2 52 1 1 $IDMS/1209
REC DB-KEY-A 49 4 $Hex compares IDMS/2721
REC USER-REC-ID 53 2 1
REC USER-REC-ID-A 53 2 $Hex compares IDMS/1093
REC PG-DISPL 55 2 1
REC IMAGE-LEN 57 2 1
REC PREFIX-LEN 59 2 1
REC DISPLACEMENT 61 2 1
REC SPAN-OFFSET 63 2 1
REC USER-RECORD 65 2 1
REC $ Rnn.n DXXX RECORDS: DIND,DCOM,DBAK,DPND,DFGT ******************
REC DTRIDQ 37 24
REC DTRID 37 16
REC DNODE 37 8
REC DID 45 8
REC DBRANCH-ID 53 8
REC DFLAG1 61 1
REC DFLAG2 62 1
REC STATE 63 1
Summary of Records Required for Journal Reports

The following table lists the journal reports and journal records from which each report extracts data. Columns 3 through 10 are the journal record types, as follows:

- **T** - TIME
- **B** - BGIN
- **A** - AREA
- **C** - COMT
- **E** - ENDJ
- **AB** - ABRT
- **BF** - BFOR
- **AF** - AFTR
- **D** - DIND, DCOM, DBAK, DPND, DFGT

<table>
<thead>
<tr>
<th>Report no. and title</th>
<th>T</th>
<th>B</th>
<th>A</th>
<th>C</th>
<th>E</th>
<th>AB</th>
<th>BF</th>
<th>AF</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 Transaction Summary</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02 Program Termination Statistics</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03 Program I/O Statistics</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04 Program Summary</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05 Detail Area/Transaction</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>06 Detail Program/Area</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07 Area Summary</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>08 Formatted Record Dump</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>09 User ID</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 External User Identity</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 Count By Journal</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Producing Journal Reports

- Uses Archive Journal File as Input (see page 72)
- Required CA Culprit for CA IDMS Modules (see page 72)
- Syntax (see page 72)
- Parameters (see page 73)
- Examples (see page 74)
- Operating System Considerations (see page 75)

Uses Archive Journal File as Input

A journal report uses an archive journal file as input. The archive journal file is created by off-loading the disk journal file(s) to a tape or disk backup file using the ARCHIVE JOURNAL utility. If the journal file is written directly to tape, the tape journal file can be used as input in place of the archive journal file. *Disk journal files that have not been offloaded cannot be used as the input file.*

Required CA Culprit for CA IDMS Modules

The following CA Culprit for CA IDMS modules, which are stored in the start-up dictionary at installation, are required to report on journal files:

- One input module (CULLJRNL) consisting of an Assembler routine that reads and deblocks the archive journal file and passes individual records to the CA Culprit for CA IDMS buffer
- One module (JREPORT 000) that provides REC parameters defining all fields contained in the journal records
- One of eight report modules (JREPORT 001 to JREPORT 008) that process and format information extracted from the journal file

Syntax

Syntax for the CA Culprit for CA IDMS input parameters is shown as follows, followed by examples. Coding is freeform, except for the starting column of the initial keyword in each parameter. JREPORT= begins in column 1; all other parameters begin in column 2.
Parameters

Syntax rules appear in Section 1, "Introduction," except as follows:

- **INPUT**
  Specifies the record size and block size of the input archive journal file and also names the CA Culprit for CA IDMS input module required to execute journal reports.

- **archive=block-size**
  Specifies the record size and block size of the archive journal file. *Archive-block-size* must equal the archive journal block size specified in the physical DDL ARCHIVE JOURNAL statement.

  **Note:** *Archive-block-size* must be specified twice.

- **UM(CULLJRN)**
  Specifies the name of the CA Culprit for CA IDMS input module.

- **JREPORT=jreport-number**
  Identifies the reports to be printed. Multiple JREPORT statements can be submitted in a single run; each statement can specify one or more report numbers. The JREPORT statement must be coded starting in column 1.
  *jreport-number* specifies a journal report number in the range 001 through 008; leading zeros can be omitted. Multiple journal report numbers must be separated by blanks or commas.

- **SELECT/BYPASS WHEN boolean-expression**
  Specifies selection criteria to be applied to every journal record during the extract phase of processing.
  The CA Culprit for CA IDMS buffer contains one journal record at a time. Selection criteria are applied to each record. Therefore, if the selection criteria pertain to a field in one journal record type, additional selection criteria must be specified to select (rather than bypass) other journal record types required for a particular report.

  **Note:** To determine which fields are contained in a journal record, refer to JREPORT-000. If a field is not in all journal record types, you must select the type in conjunction with the field to ensure correct record addressing. Field TYPE will be equal to one of the four-character journal record types documented under *Types of Journal Records* (see page 62).
Examples

Example 1

CA Culprit for CA IDMS retrieves JREPORT 002 and JREPORT 004 from the CULPDICT dictionary. The archived journal file contains 19068-byte, fixed-length records. By default, the CA Culprit for CA IDMS is not printed in the Sequential and Input Parameter Listings generated for the report.

```
DATABASE DICTNAME=CULPDICT
INPUT 19068 19068 UM(CULLJRNL)
JREPORT=2,4
```

Example 2

In this example, selected records are printed in JREPORT 008 (Formatted Record Dump). The SELECT parameter selects only those BFOR and AFTR records that have a dbkey of 'X'000CEB08'.

```
DATABASE DICTNAME=CULPDICT
INPUT 4276 4276 UM(CULLJRNL)
JREPORT=008
SELECT TYPE EQ ('BFOR' 'AFTR') AND
  DBKEY-A EQ (X'000CEB08')
```

Example 3

In this example, JREPORTs 005 and 006 are requested. By default, CA Culprit for CA IDMS retrieves the journal report modules from the system dictionary. For both reports, the CA Culprit for CA IDMS code is printed on the Sequential Parameter Listing and on the Input Parameter Listing.

```
INPUT 19068 19068 UM(CULLJRNL)
PARAM=LIST
JREPORT=05
JREPORT=06
```

Example 4

This example prints JREPORT 008 (Formatted Record Dump). The report includes only the BFOR and AFTR records within the specified sequence for run unit 44248. By default, the CA Culprit for CA IDMS code for the report is not printed in the parameter listings.

```
INPUT 19068 19068 UM(CULLJRNL)
JREPORT=008
SEL WHEN TYPE EQ ('BFOR' 'AFTR')
  AND TRANSACT-ID EQ 44248
  AND SEQ EQ (1755732 TO 1755749)
```

Example 5

This example prints selected information for JREPORT 007, the Area Summary report. The SELECT parameter selects all AREA records that specify DDLDML or DDLDCLOD. The SELECT parameter also selects record types BGIN, ENDJ, and ABRT, which are required for JREPORT 007.

```
INPUT 4276 4276 UM(CULLJRNL)
JREPORT=007
SELECT WHEN (TYPE EQ 'AREA' AND AREA EQ ('DDLDML' 'DDLDCLOD'))
  OR TYPE EQ ('ABRT' 'BGIN' 'ENDJ')
```

Example 6
This example prints JREPORT 008 (Formatted Record Dump). The report includes the distributed checkpoint records for the distributed transaction ID, SYSTEM74::01650C9509CE38A3.

**INPUT** 19068 19068 UM(CULLJRNL)
JREPORT=008
SELECT WHEN TYPE EQ ('DIND' 'DCOM' 'DBAK' 'DPND' 'DFGT')
*   AND DNODE = 'SYSTEM74'
*   AND DID = x'01650C9509CE38A3'

---

**Operating System Considerations**

Journal reports can run either in local mode or under central version. The JCL to run journal reports appears in Appendices A through D for z/OS, z/VSE, z/VM and z/VM operating systems, respectively. The archived journal file must be defined with ddname/filename/linkname SYS010, as follows:

- **For z/OS operating systems**, modify ddname SYS010 in Appendix A, as follows:

  //SYS010   DD  DSN=idms.archive,DISP=OLD,UNIT=tape,VOL=SER=nnnnnn

<table>
<thead>
<tr>
<th>idms.archive</th>
<th>data set name of the archive journal file</th>
</tr>
</thead>
<tbody>
<tr>
<td>tape</td>
<td>symbolic device name of the archive journal file</td>
</tr>
<tr>
<td>nnnnnn</td>
<td>volume serial number of the archive journal file</td>
</tr>
</tbody>
</table>

- **For z/VSE operating systems**, modify filename SYS010 in Appendix B, as follows:

  //TLBL   SYS010,'idms.archive'
  //ASSGN SYS010,TAPE,VOL=nnnnnn

<table>
<thead>
<tr>
<th>idms.archive</th>
<th>file-id of the archive journal file</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnnn</td>
<td>volume serial number</td>
</tr>
</tbody>
</table>

- **For z/VM and z/VM operating systems**, modify ddname SYS010 in Appendix C, as follows:

  For tape file: FILEDEF SYS010 TAPI SL VOLID nnnnnn
                  (RECFM FB LRECL lll BLKSIZE bbbb)

  For disk file: FILEDEF SYS010 DISK input file a

| nnnnnn       | volume serial number of the archive journal file |
| lll          | record length of the archive journal file |
| bbbb         | block size of the archive journal file |
| input file a | filename, filetype, and filemode of the archive journal file |

---

**JREPORT 001 - Transaction Summary**

JREPORT 001, the Transaction Summary report:

- Lists every program for which activity is recorded in the journal file
Provides information on the duration and disposition of each transaction associated with the programs.

Extracts statistics from the BGIN, COMT, ENDJ, and ABRT journal records for each transaction.

The output is sorted by transaction identifier within program name.

The following report shows one page of a sample report:

**Sample JREPORT 001:**

<table>
<thead>
<tr>
<th>PROGRAM NAME</th>
<th>NODE</th>
<th>TRANSACTION ID</th>
<th>TASK-ID</th>
<th>ORIGIN ID</th>
<th>DATE</th>
<th>TIME</th>
<th>DATE</th>
<th>TIME</th>
<th>TRM.</th>
<th>COMMITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0043</td>
<td>67</td>
<td>DBDC</td>
<td>67</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>ENDJ</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0046</td>
<td>111</td>
<td>DBDC</td>
<td>111</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>ENDJ</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0048</td>
<td>159</td>
<td>DBDC</td>
<td>159</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>ENDJ</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0051</td>
<td>186</td>
<td>DBDC</td>
<td>186</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>ENDJ</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0052</td>
<td>217</td>
<td>DBDC</td>
<td>217</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>ENDJ</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0055</td>
<td>233</td>
<td>DBDC</td>
<td>233</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>ENDJ</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0059</td>
<td>229</td>
<td>DBDC</td>
<td>229</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>ENDJ</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0058</td>
<td>230</td>
<td>DBDC</td>
<td>230</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>ENDJ</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0062</td>
<td>227</td>
<td>DBDC</td>
<td>227</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>ENDJ</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0064</td>
<td>225</td>
<td>DBDC</td>
<td>225</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>ENDJ</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0060</td>
<td>241</td>
<td>DBDC</td>
<td>241</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>ENDJ</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0057</td>
<td>231</td>
<td>DBDC</td>
<td>231</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>mm/dd/yy hh:mm.ss.ht</td>
<td>ENDJ</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**Field Descriptions**

A description of the fields in the Transaction Summary report follows:

- **PROGRAM NAME**
  Indicates the name of the program to which the information applies.

  **Note:** For non-SQL transactions, the subschema control supplies the name of the program. If the program name is blank, then the program issued a BIND RUN UNIT before moving its name to the name field in the subschema control.

  For SQL transactions, the RCM for the program initiating the transaction supplies the name of the program.

- **NODE**
  Specifies the node name of the central version on which the transaction executed. If the journal being analyzed was created in local mode, this field will contain *local*. 

TRANSACT ID
Specifies the unique identifier (LID) assigned to each transaction associated with the program.

TASK
Specifies the identifier of the task with which the transaction is associated. The task identifier is assigned by the DC/UCF system at run time.

ORIGINATOR ID
 Specifies the identifier consisting of a 4-character name that designates the originating interface (for example, BATC, DBDC, or CICS) and a numeric identifier assigned to the transaction by the interface.

START DATE/START TIME
Specifies the date and time the transaction started.

END DATE/END TIME
Specifies the date and time the transaction ended.

TERMINATION RECORDS
Specifies the journal record type (ENDJ or ABRT) marking the termination of the transaction.

COMMITS
Identifies the number of COMT records written for the transaction. (That is, the number of COMMIT or COMMIT WORK CONTINUE statements issued by the transaction).

JREPORT 002 - Program Termination Statistics

JREPORT 002, the Program Termination Statistics report:

- Lists every program for which activity is recorded in the journal file.
- Contains detailed information on the database processing activities of each transaction associated with the programs.
- Extracts statistics from the ENDJ and ABRT journal records for each transaction.

The output is sorted by transaction identifier within program name.

The following report shows one page of a sample report.

Sample JREPORT 002:
Field Descriptions

A description of the fields in the Program Termination Statistics report follows:

- **PROGRAM/NODE**
  Identifies the name of the program to which the information applies and the node name on which it executed. Transactions that terminate abnormally are flagged as ABORT.

  **Note:** For non-SQL transactions, the subschema control supplies the name of the program. If the program name is blank, then the program issued a BIND RUN UNIT before moving its name to the name field in the subschema control.

For SQL transactions, the RCM for the program initiating the transaction supplies the name of the program.

- **TRANSACT ID**
  Specifies the unique identifier (LID) assigned to each transaction.

- **LOCKS REQUESTED**
  Indicates the number of locks acquired by the transaction.

- **SHARED LOCKS HELD**
  Indicates the number of shared locks held by the transaction at the time the transaction ended.

- **EXCLUSIVE LOCKS HELD**
  Indicates the number of exclusive locks held by the transaction at the time the transaction ended.
- **CALC RECORDS STORED ON TARGET/NOT STORED ON TARGET**
  Indicates the number of CALC records stored/not stored on their target page.
  Interpretation: The ratio of CALC records stored on target to the total number stored (that is, hits plus overflows) is the **CALC cluster ratio**. The ratio reflects the efficiency of the CALC algorithm. Ideally, the value should be 1, which indicates no overflow. Values less than 1 or less than the norm indicate space utilization is getting high and database tuning should be performed.

- **VIA RECORDS STORED ON TARGET/NOT STORED ON TARGET**
  Identifies the number of VIA and/or DIRECT records stored/not stored on their target page.
  Interpretation: The ratio of VIA records stored on target to the total number stored (that is, hits plus overflows) is the **VIA cluster ratio**. The ratio reflects how well VIA records cluster around their owner. Ideally, the value should be 1, which indicates no overflow. Values less than 1 or less than the norm indicate very large data clusters, high utilization of space, or small page size.

- **RECORDS REQUESTED FROM DB**
  Indicates the number of database records requested by IDMSDBMS.

- **RECORDS CURRENT OF TRANS**
  Indicates the number of times the current-of-transaction field in the subschema control block for the transaction was updated.
  Interpretation: The ratio of records requested to records current-of-transaction is the **effectiveness ratio**. The ratio measures the amount of work CA IDMS/DB is doing for the programmer (that is, how many records the DBMS has to examine to find the one requested). The information in JREPORT 002 indicates whether the ratio is constant for all executions of the program or only for certain transactions. The value should be as low as possible. If the value is high, examine set options (for example, sorted order) for appropriateness. If the options are correct, examine the program logic for accurate use of currency.

- **CALLS TO DB**
  Indicates the number of calls to IDMSDBMS issued by the transaction. Execution of each navigational DML request involves one call; execution of each LRF and SQL request typically involves multiple calls.

- **FRAGMENTS STORED**
  Indicates the number of noncontiguous segments (fragments) stored for variable-length records.
  Interpretation: If the number of stored fragments is large or increasing, increase the page size to accommodate larger records. The condition of the databases and areas involved with the program should be analyzed further with the Database Analysis Utility (IDMSDBAN), described in Other CA IDMS Reporting Facilities (see page 659).

- **FRAGMENTS RETND**
  Indicates the number of records relocated from their home page.

---

**JREPORT 003 - Program I/O Statistics**

JREPORT 003, the Program I/O Statistics report:

- Lists every program for which activity is recorded in the journal file
Provides information on the I/O services requested by and performed for each transaction associated with the programs.

Extracts statistics from the ENDJ and ABRT journal records for each transaction.

The output is sorted by transaction identifier within program name.

The following report shows one page of a sample report:

Sample JREPORT 003:

<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>NODE</th>
<th>TRANSACTION ID</th>
<th>READ</th>
<th>WRITTEN</th>
<th>REQUESTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0043</td>
<td>2</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0044</td>
<td>2</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0045</td>
<td>2</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0046</td>
<td>2</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0047</td>
<td>2</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0048</td>
<td>2</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0050</td>
<td>2</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0051</td>
<td>2</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0052</td>
<td>2</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0053 ** ABORT **</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0054</td>
<td>2</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>DBCRUPD</td>
<td>SYSTEM72</td>
<td>0055</td>
<td>2</td>
<td>2</td>
<td>7</td>
</tr>
</tbody>
</table>

Field Descriptions

A description of the fields in the Program I/O Statistics report follows:

- **PROGRAM**
  Identifies the name of the program to which the information applies.

  **Note:** For non-SQL transactions, the subschema control supplies the name of the program. If the program name is blank, then the program issued a BIND RUN UNIT before moving its name to the name field in the subschema control.

  For SQL transactions, the RCM for the program initiating the transaction supplies the name of the program.

- **NODE**
  Specifies the node name of the central version on which the transaction executed. If the journal being analyzed was created in local mode, this field will contain *local*.

- **TRANSACTION ID**
  Is the unique identifier (LID) assigned to each transaction associated with the program. Transactions that terminate abnormally are flagged as ABORT.

- **PAGES READ**
  Indicates the number of pages physically read from disk for the transaction.
- **PAGES WRITTEN**
  Indicates the number of pages physically written to disk for the transaction. A page can be updated several times before it is actually written back to the database.

- **PAGES REQUESTED**
  Indicates the number of pages requested by IDMSDBMS (including pages found in a buffer). A page request does not result in a page read if the page is in the buffer pool.
  Interpretation: The ratio of pages requested to pages read is the *buffer utilization ratio*. It measures the effectiveness of the buffer-pool size and design of the database (for example, CALC and VIA clustering). The higher the ratio the better. Ratios consistently below 2.0 indicate that processing is random or that the buffer-pool size is too small. The information in JREPORT 003 indicates whether the ratio is constant for all executions of the program or only for certain transactions.

> **Note:** The buffer utilization ratio may be artificially high for transactions that keep locks. IDMSDBMS cannot hold a buffer while waiting for a lock; therefore, when locks are kept, IDMSDBMS must free and request a page each time a lock is requested for which a wait must occur.

### JREPORT 004 - Program Summary

JREPORT 004, the Program Summary report:

- Lists every program for which activity is recorded in the journal file
- Provides summary information on the database processing activities of the program
- Compiles statistics from information extracted from the ENDJ and ABRT journal records for transactions associated with each program
- Indicates the averages and ratios for all programs combined

The output is sorted by program name.

The following report shows one page of a sample report:

**Sample JREPORT 004:**

<table>
<thead>
<tr>
<th>PROGRAM NAME</th>
<th>TIMES RUN</th>
<th>PAGES READ</th>
<th>PAGES WRITTEN</th>
<th>CALLS</th>
<th>RECORDS CALLED</th>
<th>BUFFER UTILIZATION EFFECTIVENESS</th>
<th>CALC CLUSTER</th>
<th>SPACE MANAGEMENT</th>
<th>VIA CLUSTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCRUPD</td>
<td>881</td>
<td>2</td>
<td>2</td>
<td>15</td>
<td>12</td>
<td>4.03</td>
<td>2.99</td>
<td>7.04</td>
<td>1.80</td>
</tr>
<tr>
<td>TOTAL</td>
<td>881</td>
<td>2</td>
<td>2</td>
<td>15</td>
<td>12</td>
<td>4.03</td>
<td>2.99</td>
<td>7.04</td>
<td>1.80</td>
</tr>
</tbody>
</table>
Field Descriptions

A description of the fields in the Program Summary report follows:

- **PROGRAM NAME**
  Identifies the name of the program to which the information applies.

  **Note:** For non-SQL transactions, the subschema control supplies the name of the program. If the program name is blank, then the program issued a BIND RUN UNIT before moving its name to the name field in the subschema control.

  For SQL transactions, the RCM for the program initiating the transaction supplies the name of the program.

- **TIMES RUN**
  Indicates the number of times the program was run.

- **AVERAGES/ PAGES READ**
  Indicates the average number of pages physically read from disk for each run of the program.

- **AVERAGES/ PAGES WRITTEN**
  Indicates the average number of pages physically written to disk for each run of the program.

- **AVERAGES/ CALLS**
  Indicates the average number of calls to IDMSDBMS issued for each run of the program.

- **AVERAGES/ RECORDS REQUESTED**
  Indicates the average number of database records requested by IDMSDBMS for each run of the program.

- **RATIOS/ BUFFER UTILIZATION**
  Indicates the ratio of number of pages requested to pages read.
  Interpretation: This ratio indicates the effectiveness of the buffer-pool size and design of the database (for example, CALC and VIA clustering). The higher the ratio, the better. Ratios consistently below 2.0 indicate that processing may be random or that the buffer-pool size may be too small.
  Particular attention should be given to frequently used programs. If the ratio for a program is below 2.0, users can look at the statistics generated by JREPORT 003 to determine whether the number of pages requested and pages read were constant for all executions of the program or only for certain transactions.

  **Note:** The buffer utilization ratio may be artificially high for transactions that keep locks. IDMSDBMS cannot hold a buffer while waiting for a lock; therefore, when locks are kept, IDMSDBMS must free and request a page each time a lock is requested for which a wait must occur.
**RATIOS/ EFFECTIVENESS**
Indicates the ratio of records requested to records made current-of-run-unit.
Interpretation: The effectiveness ratio indicates the amount of work CA IDMS/DB is doing for the programmer (that is, how many records IDMSDBMS has to examine to find the one requested). The lower the ratio the better. If the ratio is very high, examine set options (for example, sort order or next pointers only) for appropriateness. If the options are correct, examine the program logic for accurate use of currency.

**RATIOS/ CALC CLUSTER**
Indicates the ratio of the number of CALC records stored on their target page to the total stored (that is, hits plus overflows). The ratio reflects the efficiency of the CALC algorithm.
Interpretation: The CALC cluster ratio is especially important when the database is loaded or restructured. Ideally, the ratio should be 1, which indicates no overflow. Ratios less than 1 or less than the norm indicate that space utilization is getting high and database tuning should be performed.

**RATIOS/ SPACE MANAGEMENT**
Indicates the ratio of records requested by IDMSDBMS to pages read from the database.
Interpretation: The space management ratio measures how well space is allocated (for example, VIA options, CALC distribution, and buffering). The higher the ratio the better. Ratios less than 4 or less than the norm indicate that the size of the buffer should be increased and database tuning should be performed.

**Note:** The space management ratio may be artificially high for transactions that keep locks. IDMSDBMS cannot hold a buffer while waiting for a lock; therefore, when locks are kept, IDMSDBMS must free and request a page each time a lock is requested for which a wait must occur.

**RATIOS/ VIA CLUSTER**
Specifies the ratio of VIA records stored on the same page as their owner to the total number of VIA records stored (that is, hits plus overflows). The ratio reflects how well VIA records cluster around their owner.
Interpretation: The VIA cluster ratio gives an indication of VIA storage requirements and is an indirect measure of the amount of space still available in the area. Ideally, the ratio should be 1, which indicates no overflow. Ratios less than 1 or less than the norm indicate very large data clusters, high utilization of space, or small page size. The cause may be a database design in which there are too many VIA records (of a particular record type) or in which the VIA records are too large.

---

**JREPORT 005 - Detail Area/Transaction**

JREPORT 005, the Detail Area/Transaction report:

- Lists every database area identified in the journal file
- Provides detailed information on transaction access to the area
- Extracts information from the BGIN, AREA, ENDJ, and ABRT journal records for each transaction
The output is sorted by transaction identifier within program name within area name.

The following report shows one page of a sample Detail Area/Transaction report:

**Sample JREPORT 005:**

<table>
<thead>
<tr>
<th>AREA NAME</th>
<th>PROGRAM NAME</th>
<th>NODE</th>
<th>TRANSACT ID</th>
<th>OPEN ACCESS</th>
<th>OPEN MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCTHIST</td>
<td>DBCRP0</td>
<td>SYSTEM72</td>
<td>0043</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0044</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0045</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0046</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0047</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0048</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0050</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0051</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0052</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0053</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0054</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0055</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td>BRNCHTEL</td>
<td>DBCRP0</td>
<td>SYSTEM72</td>
<td>0043</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0044</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0045</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0046</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0047</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0048</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0050</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0051</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0052</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0053</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0054</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYSTEM72</td>
<td>0055</td>
<td>SHARED</td>
<td>UPDATE</td>
</tr>
</tbody>
</table>

**Field Descriptions**

A description of the fields in the Detail Area/Transaction report follows:

- **AREA NAME**
  Specifies the subschema name of the area to which the access information applies.

- **PROGRAM NAME**
  Specifies the name of each program that accesses the area.

  **Note:** The subschema name field supplies the name of the program. If the program name is blank, then the program issued a BIND RUN UNIT before moving its name to the subschema name field.

- **NODE**
  Specifies the node name of the central version on which the transaction executed. If the journal being analyzed was created in local mode, this field will contain *local*.

- **TRANSACT ID**
  Identifies the unique identifier (LID) assigned to each transaction associated with the program.
OPEN ACCESS
Identifies area usage mode options that prevent update or retrieval of an area by other transaction executing concurrently under the central version. Possible options are SHARED (default), PROTECTED, and EXCLUSIVE.

OPEN MODE
Identifies the mode in which the transaction opened the area. Possible modes are:
- UPDATE - The transaction can issue all DML functions for records in the area
- RETRIEVAL - The transaction cannot issue DML or DDL requests that result in updates to data in the area

JREPORT 006 - Detail Program/Area

JREPORT 006, the Detail Program/Area report:
- Lists every program for which activity is recorded in the journal file
- Provides detailed information on the areas accessed by each transaction associated with the programs
- Extracts information from the BGIN, AREA, ENDJ, and ABRT journal records for each transaction

With this report, you can identify what areas are used exclusively by certain programs and are therefore not available to concurrently running programs. The output is sorted by area name within transaction identifier within program name.

The following report shows one page of a sample Detail Program/Area report:

Sample JREPORT 006:
Field Descriptions

A description of the fields in the Detail Program/Area report follows:

- **PROGRAM NAME**
  Identifies the name of the program to which the information applies.

  **Note:** The subschema name field supplies the name of the program. If the program name is blank, then the program issued a BIND RUN UNIT before moving its name to the subschema name field.

- **NODE**
  Specifies the node name of the central version on which the transaction executed. If the journal being analyzed was created in local mode, this field will contain *local*.

- **TRANSACT ID**
  Identifies the unique identifier (LID) assigned to each transaction associated with the program.

- **AREA NAME**
  Identifies the subschema name of each area accessed by the transaction.

- **OPEN ACCESS**
  Identifies the area usage mode options that prevent update or retrieval of an area by other transactions executing concurrently under the central version. Possible options are SHARED (default), PROTECTED, and EXCLUSIVE.

- **OPEN MODE**
  Identifies the mode in which the transaction opened the area. Possible modes are:

  - **UPDATE** - The transaction can issue all DML functions for records in the area
  - **RETRIEVAL** - The transaction cannot issue DML or DDL requests that result in updates to data in the area

**JREPORT 007 - Area Summary**

JREPORT 007, the Area Summary report:

- Lists every area identified in the journal file
- Provides summary information on program access to the area
- Extracts statistics from the BGIN, AREA, ENDJ, and ABRT journal records for each transaction
Indicates the total number of times all areas were readied in each mode.

The output is sorted by program name within area name.

The following report shows one page of a sample Area Summary report:

**Sample JREPORT 007:**

<table>
<thead>
<tr>
<th>REPORT NO.</th>
<th>JREPORT 007</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>IDMS JOURNAL REPORT</th>
<th>AREA SUMMARY</th>
<th>Rnn.n</th>
<th>mm/dd/yy</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AREA NAME</td>
<td>PROGRAM NAME</td>
<td>SHARED</td>
<td>EXCLUSIVE</td>
</tr>
<tr>
<td></td>
<td>ACCTHIST</td>
<td>DBCRUPD</td>
<td>881</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BRNCHTEL</td>
<td>DBCRUPD</td>
<td>881</td>
<td></td>
</tr>
<tr>
<td><strong>TOTAL FOR ALL AREAS</strong></td>
<td></td>
<td></td>
<td><strong>1762</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Field Descriptions**

A description of the fields in the Area Summary report follows:

- **AREA NAME**
  Identifies the subschema name of the area to which the information applies.

- **PROGRAM NAME**
  Identifies the name of each program that accesses the area.

  **Note:** For non-SQL transactions, the subschema control supplies the name of the program. If the program name is blank, then the program issued a BIND RUN UNIT before moving its name to the name field in the subschema control.

For SQL transactions, the RCM for the program initiating the transaction supplies the name of the program.

- **SHARED RETRIEVAL**
  Indicates the number of times transactions associated with the program readied the area in shared retrieval mode.

- **EXCLUSIVE RETRIEVAL**
  Indicates the number of times transactions associated with the program readied the area in exclusive retrieval mode.

- **PROTECTED RETRIEVAL**
  Indicates the number of times transactions associated with the program readied the area in protected retrieval mode.

- **SHARED UPDATE**
  Indicates the number of times transactions associated with the program readied the area in shared update mode.
• **EXCLUSIVE UPDATE**
  Indicates the number of times transactions associated with the program readied the area in exclusive update mode.

• **PROTECTED UPDATE**
  Indicates the number of times transactions associated with the program readied the area in protected update mode.

## JREPORT 008 - Formatted Record Dump

JREPORT 008, the Formatted Record Dump:

• Produces a formatted dump of TIME, BGIN, AREA, COMT, ENDI, ABRT, BFOR, and AFTR journal records, allowing the user to inspect the entire contents of the journal file.

• Provides a character and hexadecimal dump of the user record portion of BFOR and AFTR journal entries.

• Provides a character and hexadecimal dump of the distributed checkpoint journal records, as well as formats the distributed transaction ID.

• Includes JSEG (journal segment) dumps in character and hexadecimal format. For the layout of the JSEG journal header record, see DSECT #JTRDS in the **CA IDMS DSECT Reference section**.

The following report shows one page from a sample Formatted Record Dump:
A description of the fields in the Formatted Record Dump follows:

- **JOURNAL RECORD**
  Identifies the type of journal record (TIME, BGIN, AREA, COMT, ENDJ, ABRT, BFOR, or AFTR).

- **NODE NAME**
  Specifies the node name of the central version on which the transaction executed. If the journal being analyzed was created in local mode, this field will contain *local*.

- **UTC DATE**
  Identifies the UTC date on which the journal record was written to the journal file (TIME record only) or placed into the journal buffer (all other records). The date is given in the same form as LOCAL DATE.

- **UTC TIME**
  Identifies the UTC time at which the journal record was written to the journal file (TIME record only) or placed into the journal buffer (all other records). The time is given in the same form as LOCAL TIME.

- **JOURNAL SEQUENCE**
  Identifies the sequence number assigned to the journal record.
• **TRANSACT ID**
  Identifies the unique identifier (LID) assigned to the transaction for which the journal record was written.

• **VERB NUM.**
  (BFOR and AFTR records only) Identifies the function code of the navigational DML verb issued by an application program or the SQL or LRF runtime processor.

• **PG GRP**
  (AREA, BFOR and AFTR records only) Identifies the page group associated with the area in which the user record is stored.

• **PROGRAM NAME**
  (BGIN, COMT, ENDJ, and ABRT records only) Identifies the name of the program with which the transaction is associated.

• **LOCAL DATE**
  Identifies the date on which the journal record was either written to the journal file (TIME record only) or placed into the journal buffer (BGIN, COMT, ENDJ, and ABRT records only). The date is given in the form *mm/dd/yy*, where *mm* is the month, *dd* is the day, and *yy* is the last two digits of the year.

• **LOCAL TIME**
  Identifies the time at which the journal record was either written to the journal file (TIME record only) or placed into the journal buffer (BGIN, COMT, ENDJ, and ABRT records only). The time is given in the form *hhmmssht*, where *hh* is hours on a 24-hour clock, *mm* is minutes, *ss* is seconds, and *ht* is hundredths of a second.

• **TRANSACT QUIESCE LEVEL**
  (BGIN, COMT, ENDJ, and ABRT records only) Identifies the number of open transactions after the journal record was written to the journal file.

• **LOCAL TASK ID**
  (BGIN, COMT, ENDJ, and ABRT records only) Is the identifier consisting of a 4-character name that designates the originating interface (for example, BATC, DBDC, or CICS) and a numeric identifier assigned to the transaction by the interface.

• **USER ID**
  The user ID of the user signed on who executed the application causing the BGIN to be written.

• **EXTERNAL ID**
  The external identity of the user signed on to web based application. This field and title will not be displayed if the external identity is equal to spaces.

• **CALLS TO DB**
  (COMT, ENDJ, and ABRT records only) Indicates the number of calls to IDMSDBMS issued by the transaction. Execution of each navigational DML request involves one call; execution of each LRF or SQL request typically involves multiple calls. On COMT checkpoints, this value is a running total for the transaction.
- **PAGES REQUESTED**
  (COMT, ENDJ, and ABRT records only) Indicates the number of pages requested by the transaction (including pages found in a buffer). A page request does not result in a page read if the page is in the buffer pool. On COMT checkpoints, this value is a running total for the transaction.

- **PAGES READ**
  (COMT, ENDJ, and ABRT records only) Indicates the number of pages physically read from disk. On COMT checkpoints, this value is a running total for the transaction.

- **PAGES WRITTEN**
  (COMT, ENDJ, and ABRT records only) Indicates the number of pages physically written to disk. On COMT checkpoints, this value is a running total for the transaction.

- **RECDS REQUESTED**
  (COMT, ENDJ, and ABRT records only) Indicates the number of database records requested by IDMSDBMS for the transaction. On COMT checkpoints, this value is a running total for the transaction.

- **RECORDS UPDATED**
  (COMT, ENDJ, and ABRT records only) Indicates the number of records updated by the transaction. On COMT checkpoints, this value is a running total for the transaction.

- **CURRENT OF TR**
  (COMT, ENDJ, and ABRT records only) Indicates the number of times the current-of-transaction field in the subschema control block for the transaction was updated. On COMT checkpoints, this value is a running total for the transaction.

- **CALC ON TARGET**
  (COMT, ENDJ, and ABRT records only) Identifies the number of CALC records stored on their target page. On COMT checkpoints, this value is a running total for the transaction.

- **CALC NOT TARGET**
  (COMT, ENDJ, and ABRT records only) Identifies the number of CALC records not stored on their target page. On COMT checkpoints, this value is a running total for the transaction.

- **VIA ON TARGET**
  (COMT, ENDJ, and ABRT records only) Identifies the number of VIA and/or DIRECT records stored on their target page. On COMT checkpoints, this value is a running total for the transaction.

- **VIA NOT TARGET**
  (COMT, ENDJ, and ABRT records only) Identifies the number of VIA and/or DIRECT records not stored on their target page. On COMT checkpoints, this value is a running total for the transaction.

- **FRAGMNTS STORED**
  (COMT, ENDJ, and ABRT records only) Identifies the number of noncontiguous segments stored for variable-length records. On COMT checkpoints, this value is a running total for the transaction.

- **RECDS RELOCATED**
  (COMT, ENDJ, and ABRT records only) Identifies the number of records relocated from their home page. On COMT checkpoints, this value is a running total for the transaction.
- **LOCKS REQUESTED**  
  (COMT, ENDJ, and ABRT records only) Indicates the number of locks acquired by the transaction. On COMT checkpoints, this value is a running total for the transaction.

- **SHARED LOCKS**  
  (COMT, ENDJ, and ABRT records only) Indicates the number of shared locks held by the transaction at the time the checkpoint record was written.

- **EXCLUSIVE LOCKS**  
  (COMT, ENDJ, and ABRT records only) Indicates the number of exclusive locks held by the transaction at the time the checkpoint record was written.

- **Area name**  
  (AREA records only) Identifies the name of the area for which the checkpoint record was written.

- **LOW PAGE**  
  (AREA records only) Identifies the page number of the first page in the area.

- **HIGH PAGE**  
  (AREA records only) Identifies the page number of the last page in the area.

- **Open access**  
  (AREA records only) Identifies the mode in which the transaction accessed the area (SHARED, PROTECTED, or EXCLUSIVE).

- **Open mode**  
  (AREA records only) Identifies the mode in which the transaction opened the area (RETRIEVAL or UPDATE).

- **THIS IS A NATIVE IDMS RECORD**  
  (BFOR and AFTR records only) Indicates that the user record is a CA IDMS/DB database record. For native VSAM records, this field reads THIS IS A NATIVE VSAM RECORD.

- **PG DISPLACEMENT**  
  (BFOR and AFTR records only) Specifies the location of the user record occurrence relative to the beginning of the database page (given as a decimal offset).

- **PREFIX LENGTH**  
  (BFOR and AFTR records only) Specifies the length in bytes of the prefix portion of the user record.

- **USER RECORD ID**  
  (BFOR and AFTR records only) Indicates the record id of the user record. An asterisk following the record id indicates that this is a logically deleted record.

- **USER RECD LNGTH**  
  (BFOR and AFTR records only) Specifies the length in bytes of the entire user record (prefix and data portion) as found in the database record line index.

- **VERB NUMBER**  
  (BFOR and AFTR records only) Specifies the function code of the navigational DML verb issued by an application program or the SQL or LRF runtime processor.
- **USER RECORD DB KEY**
  (BFOR and AFTR records only) Specifies the database key of the user record occurrence in hexadecimal format.

- **PAGE/LINE**
  (BFOR and AFTR records only) Specifies the page and line number, in decimal format, of the database key of the user record occurrence.

- **DISPLACEMENT OF CALL**
  (BFOR and AFTR records only) Specifies a trace entry indicating the IDMSDBMS routine that issued the database call (for CA internal use only).

- **CHAR**
  Identifies the contents of the user record in decimal (display) format.

- **ZONE NUMR**
  Identifies the contents of the user record in hexadecimal format.

- **SPANNED OFFSET**
  (Not on sample report) Indicates that the journal entry is a continuation of the previous BFOR or AFTR image for the transaction. The number is the displacement of this portion of the record image relative to the entire image.

- **DTRID-BID**
  (DIND, DCOM, DBAK, DPND, and DFGT records only) Identifies the distributed transaction identifier (DTRID) and the branch identifier (BID) of the top-level branch of the distributed transaction for which the journal record was written.

### JREPORT 009 - User ID

JREPORT 009, the User ID report enables compliance and audit reporting. JREPORT 009:

- Lists the user ID from every transaction with the date, time, and program associated with that transaction.

  **Considerations:**

  - If you want to capture information on retrieval only transactions, you must specify JOURNAL RETRIEVAL.
  
  - If you want to capture information from local batch jobs, you must capture the journal records (many sites take a backup of their database for recovery purposes when running a local mode update job rather than writing the local mode journal file.)
  
  - Many CICS applications or web-based applications capture the user ID on the client side of the application and use a generic user ID to access the data from the database. In these cases, the generic user ID is captured in the BBEGIN checkpoint record.

The following report shows one page from a sample User ID report:

**Sample JREPORT 009:**
Field Descriptions

A description of the fields in the User ID report follows:

- **USER ID**
  Identifies the user ID of the user who executed the application that created the BGIN checkpoint record. The user ID is reported on as follows:
  - If JREPORT 009 is run against journal files created prior to r16 SP4 (the user ID is not present in the BGIN), JREPORT 009 displays USER ID NOT CAPTURED.
  - If JREPORT 009 is run against journal files created after r16 SP4, but the user does not signon, the user ID field in the BGIN is filled with spaces, and JREPORT 009 displays NO USER SIGNON.

- **UTC DATE**
  Identifies the UTC date on which the journal record was written to the journal file. The date is given in the same form as LOCAL DATE.

- **UTC TIME**
  Identifies the UTC time at which the journal record was written to the journal file. The date is given in the same form as LOCAL TIME.

- **TRANSACT ID**
  Identifies the unique identifier (LID) assigned to the transaction for which the journal record was written.

- **PROGRAM NAME**
  Identifies the name of the program with which the transaction is associated.

- **LOCAL DATE**
  Identifies the date on which the journal record was placed into the journal buffer. The date is given in the form mm/dd/yy, where mm is the month, dd is the day, and yy is the last two digits of the year.

- **LOCAL TIME**
  Identifies the time at which the journal record was placed into the journal buffer. The time is given in the form hh.mm.ss.ht, where hh is hours on a 24-hour clock, mm is minutes, ss is seconds, and ht is hundredths of a second.
JREPORT 010 - External User Identity

JREPORT 010, the External User Identity report enables compliance and audit reporting. JREPORT 010:

- Lists the user ID from every transaction with the external user ID, program name, date and time associated with that transaction.

Considerations:

- If you want to capture information on retrieval only transactions, you must specify JOURNAL RETRIEVAL.

- If this transaction was the result of a web-based application and if CA SiteMinder (or similar product) is being used and it passes the identity of the user signed on the web-based application, the EXT ID field on the report will contain the external user identity. If no external user identity was passed, or the BGIN record was the result of other than a web-based application, the EXT ID field will contain the text 'EXT ID NOT CAPTURED'.

The following report shows one page from a sample External User Identity report:

Sample JREPORT 010:

<table>
<thead>
<tr>
<th>REPORT NO. 10</th>
<th>IDMS JOURNAL REPORTS</th>
<th>R16.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>JREPORT 010</td>
<td>EXTERNAL USER IDENTITY JOURNAL REPORT</td>
<td></td>
</tr>
<tr>
<td>USER</td>
<td>EXT</td>
<td>TRANSACT</td>
</tr>
<tr>
<td>ID</td>
<td>ID</td>
<td>IDX</td>
</tr>
<tr>
<td>USERA01</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>USER ID NOT CAP.</td>
<td>6</td>
<td>IDMSDDL</td>
</tr>
<tr>
<td>USERB01</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>NO USER SIGNON</td>
<td>8</td>
<td>RHDCRUAL</td>
</tr>
<tr>
<td>USERC01</td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>C750009 RECORDS WRITTEN FOR REPORT 10 --</td>
<td>8</td>
<td>Figure 8-10. Sample JREPORT 010</td>
</tr>
</tbody>
</table>

Field Descriptions

A description of the fields in the External User Identity report follows:

- **USER ID**
  Identifies the user ID of the user who executed the application that created the BGIN checkpoint record. The user ID is reported on as follows:

  - If JREPORT 010 is run against journal files created prior to r16 SP4 (the user ID is not present in the BGIN), JREPORT 010 displays 'USER ID NOT CAPTURED'.

  - If JREPORT 010 is run against journal files created from r16 SP4, or later but prior to the IDMS Server r16.1 or prior to r16 SP6 (the user ID is present in the BGIN but the External user ID is not), and the user does not signon, the user ID field in the BGIN is filled with spaces, and JREPORT 010 displays NO USER SIGNON.
EXT ID
Identifies the external user ID of the user signed on to the web-based application that has caused the application that created the BGIN checkpoint record on the backend system - if that external user ID was passed to the backend IDMS system. The external user ID is reported on as follows:

- If JREPORT 010 is run against journal files created prior to IDMS Server r16.1 or IDMS SP6 (the external user ID is not present in the BGIN), JREPORT 010 displays 'EXT ID NOT CAPTURED'.
- If JREPORT 010 is run against journal files created after IDMS Server r16.1 or after IDMS r16 SP6. JREPORT 010 will display the contents of the external user ID field in the BGIN checkpoint which will be the external user identity passed from the web-based application or spaces.

TRANSACT IDX
Identifies the unique identifier (LID) assigned to the transaction for which the journal record was written.

PROGRAM NAME
Identifies the name of the backend program with which the transaction is associated.

LOCAL DATE
Identifies the date on which the journal record was placed into the journal buffer. The date is given in the format mm/dd/yy, where mm is the month, dd is the day, and yy is the last two digits of the year.

LOCAL TIME
Identifies the time at which the journal record was placed into the journal buffer. The time is given in the form hh.mm.ss.ht, where hh is hours on a 24-hour clock, mm is minutes, ss is seconds, and ht is hundredths of a second.

JREPORT 011 - Count By Journal Record Type Report

JREPORT 011, the count by journal record type report:

- Lists every journal record type found in an archive journal file
- Provides a count of each record type and a total count of all records

The report is sorted by journal record type.

The following report shows a sample Count By Journal Record Type report:

<table>
<thead>
<tr>
<th>ABRT</th>
<th>AFTTR</th>
<th>AREA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>617</td>
<td>42</td>
</tr>
</tbody>
</table>
Field Descriptions

A description of the fields in the Count By Journal Record Type report follows:

- **TYPE**
  Specifies the journal record type.

- **DESCRIPTION**
  Provides a text description of the journal record type.

- **COUNT**
  The number of records of the specific journal record type.

JREPORTS z/OS JCL

This section describes the JCL for journal reports in local mode and under central version.

- **Local Mode (see page 97)**
- **Central Version (see page 100)**

Local Mode

```
//CULPRIT EXEC PGM=CULPRIT,REGION=1024K
//STEPLIB DD DSN=idms.dba.loadlib,DISP=SHR
// DD DSN=idms.custom.loadlib,DISP=SHR
// DD DSN=idms.cagjload,DISP=SHR
//SORTLIB DD DSN=sys1.sortlib,DISP=SHR
//SORTLIB DD DSN=&&SPRMWORK.,DISP=(NEW,DELETE),
// DD DSN=&&SEXTWORK.,DISP=(NEW,DELETE),
// DD DSN=&&SRTPWORK.,DISP=(NEW,DELETE),
//SORTMSG DD SYSPRINT=A
```
Note: If the user's subschema is not in the CA IDMS/DB load library, then the load library that contains the module must also be included in the STEPLIB concatenation when running in local mode.
Local Mode Statements and Data Sets

| **appldict** | Name of the application dictionary. |
| **asfdata** | Name of the ASF data (IDMSR-AREA2) area; required to execute IREPORTs. |
| **ASFDEFN** | Name of the ASF data definition (IDMSR-AREA) area; required to execute IREPORTs. |
| **asfdml** | Name of the ASF dictionary definition (DDLDML) area; required to execute IREPORTs. |
| **asflod** | Name of the ASF dictionary definition load (DDLDCLOD) area; required to execute IREPORTs. |
| **bbbbb** | Blocksize of the file. |
| **dcmsg** | Name of the system message (DDLDCMSG) area. |
| **dictdb** | Name of the application dictionary definition area. |
| **dirlldb** | Name of the IDMSDIRL definition (DDLDML) area; to execute A, C, and DREPORTs, use a dictionary built by IDMSDIRL. |
| **disk** | Symbolic device name for work files. |
| **dloddb** | Name of the application dictionary definition load area; required for CREPORT 053. |
| **dccat** | Name of the SQL application dictionary catalog (DDLCAT) area. |
| **ddcatx** | Name of the SQL application dictionary catalog index (DDLCATX) area. |
| **ddcatl** | Name of the SQL application dictionary catalog load (DDLCATLOD) area. |
| **dmcl-name** | Name of the DMCL used by the central version. |
| **idms.appldict.ddlclod** | Data set name of the application dictionary definition load (DDLDCLOD) area; required for CREPORT 053. |
| **idms.appldict.ddldml** | Data set name of the application dictionary definition (DDLDML) area. |
| **idms.asfdata.asfdata** | Data set name of the ASF data (IDMSR-AREA2) area; required for IREPORTs. |
| **idms.asfdata.asfdefn** | Data set name of the ASF data definition (IDMSR-AREA) area; required for IREPORTs. |
| **idms.asfdata.ddlclod** | Data set name of the ASF dictionary definition load (DDLDCLOD) area; required for IREPORTs. |
| **idms.asfdata.ddldml** | Data set name of the ASF dictionary definition (DDLDML) area; required for IREPORTs. |
| **idms.custom.loadlib** | Data set name of the load library containing customized CA IDMS executable modules. |
| **idms.dba.loadlib** | Data set name of the load library containing the DMCL and database name table load modules. |
| **idms.cagjload** | Data set name of the load library containing the CA IDMS executable modules that do not require customization. |
| **yourHLQ.CAGISRC** | Data set name of PDS containing parameters to be copied (necessary only if USE, =COPY, or =MACRO features are being used). |
**Central Version**

**To run CA IDMS reports under central version, follow these steps:**

1. Remove the following DD statements:

   ```
   //dictdb    DD   DSN=idsms.appldict.ddldml,DISP=SHR
   //dloddb    DD   DSN=idsms.appldict.ddldclod,DISP=SHR
   //dirldb    DD   DSN=idsms.sysdirl.ddldml,DISP=SHR
   //asfdml    DD   DSN=idsms.asfdict.ddldml,DISP=SHR
   //asflod    DD   DSN=idsms.asfdict.ddldclod,DISP=SHR
   //ASFDEFN   DD   DSN=idsms.asfdict.asfdefn,DISP=SHR
   //asfdmdata DD   DSN=idsms.asfdict.asfdata,DISP=SHR
   //sysjrnl   DD   DUMMY
   ```

2. Add the following DD statement for the system control file anywhere after STEPLIB

   ```
   //sysctl     DD   DSN=idsms.sysctl,DISP=SHR
   ```

**Central Version Statements and Data Sets**

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysctl</td>
<td>Name of the SYSCTL file</td>
</tr>
<tr>
<td>idsms.sysctl</td>
<td>Data set name of SYSCTL file</td>
</tr>
</tbody>
</table>
CA IDMS/DB SQL Dictionary Reports -- QREPORTS

The following table lists the QREPORTS in order by report module number:

<table>
<thead>
<tr>
<th>QREPORT Module</th>
<th>Report Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>SQL Column Name Report</td>
</tr>
<tr>
<td>002</td>
<td>SQL Table Information Report</td>
</tr>
<tr>
<td>003</td>
<td>SQL Schema Information Report</td>
</tr>
<tr>
<td>004</td>
<td>SQL Access Module Information Report</td>
</tr>
<tr>
<td>005</td>
<td>SQL Table Access Report</td>
</tr>
<tr>
<td>006</td>
<td>SQL Table Syntax Report</td>
</tr>
<tr>
<td>007</td>
<td>SQL Table Index Report</td>
</tr>
<tr>
<td>008</td>
<td>SQL Table Constraint Report</td>
</tr>
</tbody>
</table>

For more information, see
- QREPORT 001 - SQL Column Name Report (see page 101)
- QREPORT 002 - Table and Column Report (see page 103)
- QREPORT 003 - Schema and Table Report (see page 109)
- QREPORT 004 - SQL Access Module Information (see page 112)
- QREPORT 005 - SQL Table Access Report (see page 114)
- QREPORT 006 - SQL Table Syntax Report (see page 116)
- QREPORT 007 - SQL Table Index Report (see page 117)
- QREPORT 008 - SQL Table Constraint Report (see page 119)

QREPORT 001 - SQL Column Name Report

Contents
- Symbolic Parameter Overrides (see page 102)
- Job Submission (see page 102)
- Field Descriptions (see page 102)

QREPORT 001, the Column Name Report, provides detailed information on each SQL column selected. The tabular report format lists the column names in alphabetical order.

QREPORT 001 is useful when you need to find out which tables contain a specific SQL column. It also lists the important attributes of each column.
Symbolic Parameter Overrides

- **REQUESTED_DICTIONARY (&&1)**
  Specify the 1 to 8 character name of a specific dictionary.

- **REQUESTED_SCHEMA (&&2)**
  Specify the 1 to 18 character name of a specific SQL schema.

- **REQUESTED_COLUMN (&&3)**
  Specify the 1 to 32 character name of a specific column or use a *like-predicate pattern* to request more than one column.

Job Submission

This syntax will list all of the columns from the SYSTEM schema of the TSTDICT SQL dictionary.

```
DATABASE DICTNAME=dictionary-name
USE "QRPT001" WITH VALUES (REQUESTED_DICTIONARY='TSTDICT'
  REQUESTED_SCHEMA='SYSTEM', REQUESTED_COLUMN='%')
```

Sample QREPORT 001:

<table>
<thead>
<tr>
<th>QREPORT 001</th>
<th>SQL COLUMN NAME</th>
<th>REPORT</th>
<th>mm/dd/yy</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLUMN NAME</td>
<td>SCHEMA NAME / TABLE NAME</td>
<td>DATATYPE / NULL ATTR.</td>
<td>PREC. SCALE</td>
<td>COLUMN OFFSET</td>
</tr>
<tr>
<td>NUMBER</td>
<td>SYSTEM RESOURCE</td>
<td>SMALLINT NOT NULL</td>
<td>15</td>
<td>130</td>
</tr>
<tr>
<td>NUMBLOCKS</td>
<td>SYSTEM JOURNAL</td>
<td>INTEGER NOT NULL</td>
<td>31</td>
<td>92</td>
</tr>
<tr>
<td>NUMBUFFERS</td>
<td>SYSTEM DMCL</td>
<td>SMALLINT NOT NULL</td>
<td>15</td>
<td>94</td>
</tr>
<tr>
<td>NUMCOLS</td>
<td>SYSTEM TABLE</td>
<td>SMALLINT NOT NULL</td>
<td>15</td>
<td>162</td>
</tr>
<tr>
<td>NUMCOLUMNS</td>
<td>SYSTEM CONSTRAINT</td>
<td>SMALLINT NOT NULL</td>
<td>15</td>
<td>116</td>
</tr>
<tr>
<td>NUMCOLUMNS</td>
<td>SYSTEM INDEX</td>
<td>SMALLINT NOT NULL</td>
<td>15</td>
<td>92</td>
</tr>
<tr>
<td>NUMDADS</td>
<td>SYSTEM SEGMENT</td>
<td>SMALLINT NOT NULL</td>
<td>15</td>
<td>96</td>
</tr>
<tr>
<td>NUMDBNAMES</td>
<td>SYSTEM DBTABLE</td>
<td>SMALLINT NOT NULL</td>
<td>15</td>
<td>62</td>
</tr>
<tr>
<td>NUMFILEMAPS</td>
<td>SYSTEM AREA</td>
<td>SMALLINT NULLS ALLOWED</td>
<td>15</td>
<td>96</td>
</tr>
</tbody>
</table>

Field Descriptions

A description of the fields in QREPORT 001 follows:
COLUMN NAME The name of the column included in the schema table.

SCHEMA NAME / TABLE NAME The schema and table name associated with the column.

DATATYPE / NULL ATTR. The data type of the named column, and an indicator to specify whether the column can or cannot contain null values.

PREC. The number of digits in a numeric value.

SCALE The number of digits to the right of the decimal point in a numeric value.

COLUMN / OFFSET The offset within a row to the data for the named column and the offset within a row to the null indicator value for the column. Offsets are relative to zero.

NULL INTERNAL LENGTH The internal length of the data for the named column and the internal length of the null indicator value for the column.

DEFAULT VALUE Indicates whether the WITH DEFAULT clause was specified for the column.

COL NUM IN TABLE The relative position of the column in the table.

QREPORT 002 - Table and Column Report

Contents

- Symbolic Parameter Overrides (see page 103)
- Job Submission (see page 104)
- Field Descriptions (see page 105)

QREPORT 002, the Table and Column Report, provides detailed information on each SQL Table selected, and also lists summary information on each column contained in the table.

This information is especially useful to application developers who need to know the names and attributes of each column in a table.

Symbolic Parameter Overrides

- REQUESTED_DICTIONARY (&&1)
  Specify the 1 to 8 character name of a specific dictionary.

- REQUESTED_SCHEMA (&&2)
  Specify the 1 to 18 character name of a specific SQL schema, or use a like-predicate pattern to request more than one schema.

- CREATED_BY (&&3)
  Specify the 1 to 18 character name of a specific user or use a like-predicate pattern to request more than one user.
### REQUESTED_TABLE (&&4)
Specify the 1 to 18 character name of a specific table or use a *like-predicate pattern* to request more than one table.

### Job Submission
The following syntax will produce an SQL Table Report for the `SYSTEM.COLUMN` table that resides in dictionary `TSTDICT`.

```
DATABASE DICTNAME=dictionary-name
USE "QRPT002" WITH VALUES (REQUESTED_DICTIONARY='TSTDICT',
REQUESTED_SCHEMA='SYSTEM',
CREATED_BY='%',
REQUESTED_TABLE='COLUMN')
```

**Sample QREPORT 002:**

```
QREPORT 002          SQL TABLE INFORMATION   mm/dd/yy          PAGE   1

SCHEMA NAME: SYSTEM
TABLE NAME: COLUMN

TIMESTAMP: yyyy-mm-dd hh.mm   TABLE OWNER: MET
DATE CREATED: yyyy-mm-dd hh.mm   BY USER: MET
DATE LAST USED: yyyy-mm-dd hh.mm   BY USER: MET

SEGMENT NAME: SYSCAT   TABLE ID: 1028   TABLE LENGTH: 184
AREA NAME: DOLCAT   TABLE TYPE: TABLE   DATA LENGTH: 164
LOC. MODE: CLUSTERED   PREFIX LENGTH: 20
PUT ROUTINE: COMPRESS: NO   CONTROL LENGTH: 36
GET ROUTINE: FORMAT: FIXED   FIXED LENGTH: 164
UPDATABLE:   SECTION LENGTH: 0
CHECKOPT:   NUMBER OF SYNTAX: 0

NUMBER OF COLUMNS: 19   DISPLACEMENT: 0
NUMBER OF INDEXES: 0   ESTIMATED ROWS: 0
NUMBER OF TIMES REFERENCED: 0   NUMBER PAGES: 41
REFERENCES TO OTHER TABLES: 2   NUMBER ROWS: 516
ROWS PER PAGE: 0

<table>
<thead>
<tr>
<th>COL NUM</th>
<th>COLUMN NAME</th>
<th>DATATYPE</th>
<th>PREC.</th>
<th>SCALE</th>
<th>COLUMN OFFSET</th>
<th>COLUMN LENGTH</th>
<th>DEFAULT</th>
<th>NULL ATTR.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NAME</td>
<td>CHARACTER</td>
<td>0</td>
<td>32</td>
<td>NO</td>
<td>NOT NULL</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>NUMBER</td>
<td>SMALLINT</td>
<td>15</td>
<td>32</td>
<td>2</td>
<td>NO</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>SCHEMA</td>
<td>CHARACTER</td>
<td>34</td>
<td>18</td>
<td>NO</td>
<td>NOT NULL</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>TABLE</td>
<td>CHARACTER</td>
<td>52</td>
<td>18</td>
<td>NO</td>
<td>NOT NULL</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>TYPE</td>
<td>CHARACTER</td>
<td>70</td>
<td>18</td>
<td>NO</td>
<td>NOT NULL</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>TYPECODE</td>
<td>SMALLINT</td>
<td>15</td>
<td>88</td>
<td>2</td>
<td>NO</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>PRECISION</td>
<td>SMALLINT</td>
<td>15</td>
<td>90</td>
<td>2</td>
<td>NO</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>SCALE</td>
<td>SMALLINT</td>
<td>15</td>
<td>92</td>
<td>2</td>
<td>NO</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>NULLS</td>
<td>CHARACTER</td>
<td>94</td>
<td>1</td>
<td>NO</td>
<td>NOT NULL</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>DEFAULT</td>
<td>CHARACTER</td>
<td>95</td>
<td>1</td>
<td>NO</td>
<td>NOT NULL</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>VOFFSET</td>
<td>SMALLINT</td>
<td>15</td>
<td>96</td>
<td>2</td>
<td>NO</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>VLENGTH</td>
<td>SMALLINT</td>
<td>15</td>
<td>98</td>
<td>2</td>
<td>NO</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>NOFFSET</td>
<td>SMALLINT</td>
<td>15</td>
<td>100</td>
<td>2</td>
<td>NO</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>NLENGTH</td>
<td>SMALLINT</td>
<td>15</td>
<td>102</td>
<td>2</td>
<td>NO</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>NUMVALUES</td>
<td>INTEGER</td>
<td>31</td>
<td>164</td>
<td>4</td>
<td>NO</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>SECOLKVAL</td>
<td>BINARY</td>
<td>168</td>
<td>8</td>
<td>0</td>
<td>NOT NULL</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>SECNUMVAL</td>
<td>BINARY</td>
<td>116</td>
<td>8</td>
<td>0</td>
<td>NOT NULL</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>PROCPARMTYPE</td>
<td>CHARACTER</td>
<td>124</td>
<td>1</td>
<td>NO</td>
<td>NOT NULL</td>
<td>NOT NULL</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>FILLER</td>
<td>BINARY</td>
<td>125</td>
<td>39</td>
<td>NO</td>
<td>NOT NULL</td>
<td>NOT NULL</td>
<td></td>
</tr>
</tbody>
</table>
```
Field Descriptions

A description of the fields in QREPORT 002 follows:

- **SCHEMA NAME**
  The name of the schema.

- **TABLE NAME**
  The name of the table.

- **TIMESTAMP**
  Table timestamp, used for synchronization with access module definitions.

- **TABLE OWNER**
  The owner of the schema in which this table resides.

- **DATE CREATED, BY USER**
  The userid of the person who submitted the SQL CREATE TABLE DDL, and the date and time the table was created.

- **DATE LAST UPDATED, BY USER**
  The userid of the person who last altered the table definition, and the date and time it occurred.

- **SEGMENT NAME**
  Name of the segment associated with the area where table rows are stored.

- **TABLE ID**
  Internal table identifier which identifies the rows of the table within an area.

- **TABLE LENGTH**
  The total length of a row of tables.

- **AREA NAME**
  Name of area where table rows are stored.

- **TABLE TYPE**
  The type of table. Valid values are:
  - Base table
  - Function
  - Procedure
  - Record in a non-SQL defined schema
  - Table procedure
  - View

- **DATA LENGTH**
  Internal length of the data portion of a table row (including 4-byte RDW for a compressed table).
- **LOC. MODE**
  The valid location modes are:
  - CALC
  - CLUSTERED
  - DIRECT
  - ROW ID INDEXED
  - UNIQUE CALC

- **KEY LENGTH**
  Internal length of the non-data portion (the prefix length) of a table row.

- **PUT ROUTINE**
  CA IDMS Presspack data characteristic table (DCT) name.

- **COMPRESS**
  Compress indicator:
  - 'Y' - Compressed
  - 'N' - Uncompressed
  - 'P' - Compressed with CA IDMS Presspack

- **CONTROL LENGTH**
  Internal length of the control portion (without the prefix) of a table row.

- **FORMAT**
  Format of the table row:
  - 'F' - Fixed length
  - 'V' - Variable length (compressed tables only)

- **FIXED LENGTH**
  Internal length of the fixed portion (without the prefix) of the table row.

- **UPDATABLE**
  When TYPE is 'V', updatable view indicator:
  - 'Y' - Updatable
  - 'N' - Not updatable

- **SECTION LENGTH**
  Length of the I-tree stored in the associated section table rows.

- **CHECKOPT**
  When TYPE is 'V', WITH CHECK OPTION indicator:
- 'Y' - View defined with WITH CHECK OPTION
- 'N' - View defined without WITH CHECK OPTION

**NUMBER OF SYNTAX**
Number of the "create view" or "create/alter table" DDL syntax records stored in the SYNTAX table. Create and alter table syntax is present only if it contains a CHECK clause.

**NUMBER OF COLUMNS**
Number of columns in the table or view.

**DISPLACEMENT**
Displacement, in pages, from clustering index or referenced row in a clustering constraint.

**NUMBER OF INDEXES**
Number of indexes on the table.

**ESTIMATED ROWS**
Estimated number of rows in the table.

**NUMBER OF TIMES REFERENCED**
Number of constraints in which the table is the referenced table.

**NUMBER PAGES**
Number of pages containing rows of the table when statistics were last updated.

**REFERENCES TO OTHER TABLES**
Number of constraints where this table is the referencing table.

**NUMBER ROWS**
Actual number of rows in the table when statistics were last updated.

**ROWS PER PAGE**
Number of table rows per page when statistics were last updated.

**EXTERNAL NAME**
Specifies the name of the program which will be called to process references to the procedure.

**LOCAL WORK AREA**
Represents the size of a local storage area that CA IDMS allocates at runtime and passes to each invocation of a procedure.

**TRANSACTION SHARING**
Specifies whether to enable transaction sharing for database sessions started by the procedure. Valid values are:

- ON -- enable transaction sharing
- OFF -- disable transaction sharing
- DEFAULT -- retain the transaction sharing option in effect when the procedure is invoked
• **MODE**
The mode in which the procedure executes. Valid values are:

  - **USER**
  - **SYSTEM**

• **GLOBAL WORK AREA**
Represents the size of a global storage area that CA IDMS allocates at runtime and passes to each invocation of a procedure.

• **DEFAULT DATABASE**
Specifies whether to establish a default database for database sessions started by the procedure. Valid values are:

  - **NULL** -- specifies not to establish a default database
  - **CURRENT** -- specifies to establish the database to which the SQL session is connected as the default database

• **PROTOCOL**
Specifies the protocol to use to invoke the procedure. Valid values are:

  - **IDMS** -- for procedures written in COBOL, PL/I, or Assembler
  - **ADS** -- for procedures written in CA ADS

• **SHARED KEY**
Specifies an identifier for the global storage area.

• **COL NUM**
The column number in the table.

• **COLUMN NAME**
The relative position of the column in the table.

• **DATATYPE**
The data type for the named column.

• **PREC.**
The number of digits in a numeric value.

• **SCALE**
The number of digits to the right of the decimal point in a numeric value.

• **COLUMN OFFSET**
The offset within a row into the data value of the column.

• **COLUMN LENGTH**
The internal length of the column data value.
- **DEFAULT VALUE**
  Indicates whether or not the WITH DEFAULT clause was specified for this column.

- **NULL ATTR.**
  Indicates whether or not nulls are allowed.

## QREPORT 003 -Schema and Table Report

### Contents

- Symbolic Parameter Overrides (see page 109)
- Job Submission (see page 109)
- Field Descriptions (see page 110)

QREPORT 003, the *Schema and Table Report*, provides a one page detailed listing of each SQL schema that in a dictionary. It also includes summary information on each table that belongs to the schema.

Not all schemas have associated tables. Therefore, QREPORT 003 uses an SQL "outer join" to provide information on all SQL schemas in the dictionary.

### Symbolic Parameter Overrides

- **REQUESTED_DICTIONARY (&&1)**
  Enter the 1 to 8 character name of the dictionary you wish to report on.

- **REQUESTED_SCHEMA (&&2)**
  Specify a 1 to 18 character schema name, or use a *like-predicate pattern* to request multiple schemas.

### Job Submission

The following example reports on all schemas residing in the TESTDICT dictionary whose schema name begins with EMP:

```
DATABASE DICTNAME=dictionary-name
USE "QRPT003"  WITH VALUES (REQUESTED_DICTIONARY='TSTDICT'
REQUESTED_SCHEMA='EMP%')
```

### Sample QREPORT 003 (Page 1):

```
QREPORT 003
SQL SCHEMA INFORMATION
mm/dd/yy
PAGE 1

SCHEMA NAME: EMPLODE

DATE CREATED: yyyy-mm-dd-hh.mm
SCHEMA OWNER: MET
DATE LAST UPDATED: dd-mm-yy
BY USER: MET
```

16-Jan-2018  109/898
CA IDMS - 19.0

Sample QREPORT 003 (Page 2):

QREPORT 003                     SQL SCHEMA INFORMATION   mm/dd
/yy                                PAGE  2

SCHEMA NAME: EMPSCHM

DATE CREATED: yyyy-mm-dd-hh.mm    SCHEMA OWNER:
DATE LAST UPDATED:                 BY USER:
SEGMENT NAME: EMPDEMO             SCHEMA TYPE: NON-SQL REFERENCING
AREA NAME:
NON-SQL NODE:                     REFERENCED NON-SQL SCHEMA: EMPSCHM
NON-SQL DICTNAME: APPLDICT        SCHEMA VERSION: 100

THIS SCHEMA HAS NO TABLES CURRENTLY DEFINED

Field Descriptions

A description of the fields in QREPORT 003 follows:

- **SCHEMA NAME**
  Identifies the name of the schema.

- **SCHEMA OWNER**
  Identifies the owner of the schema.

- **DATE CREATED, BY USER**
  Identifies the date and time the schema was created, and the user who created it.

- **DATE LAST UPDATED, BY USER**
  Identifies the date and time the schema was last updated, and the user who last updated it.

- **SEGMENT NAME**
  Name of the segment associated with the default area for relational schemas. For non-SQL schemas, it is the name of the database or segment containing the data that the non-SQL schema defines.
- **AREA NAME**  
  Name of the default area for relational schemas.

- **SCHEMA TYPE**  
  The type of schema. Valid values are:
  - RELATIONAL
  - NON-SQL REFERENCING
  - SQL REFERENCING

- **NON-SQL NODE**  
  Reserved for future use.

- **REFERENCED NON-SQL SCHEMA**  
  Name of the non-SQL schema.

- **NON-SQL DICTNAME**  
  Name of the dictionary in which the referenced non-SQL schema is defined.

- **SCHEMA VERSION**  
  Version number of the non-SQL schema.

- **REFERENCED SQL SCHEMA**  
  Name of the SQL schema referenced by this schema.

- **REFERENCED SQL DICTNAME**  
  Name of the dictionary in which the referenced SQL schema is defined.

- **TABLE NAME**  
  Name of the table.

- **TABLE ID**  
  Internal table identifier which identifies the rows of the table within an area.

- **TABLE TYPE**  
  The type of table. The following are the Valid values:
  - Base table
  - Function
  - Procedure
  - Record in a non-SQL defined schema
  - Table procedure
  - View

- **CREATED BY**  
  Identifies the user who created the table.
QREPORT 004 - SQL Access Module Information

Contents

- Symbolic Parameter Overrides (see page 112)
- Job Submission (see page 112)
- Field Descriptions (see page 113)

QREPORT 004, the SQL Access Module Information report, lists detail information on each access module defined in a dictionary, along with the names of all tables referenced by each access module. This report also uses an outer join to list access modules that do not reference any tables.

Symbolic Parameter Overrides

- REQUESTED_DICTIONARY (&&1)
  Enter the 1 to 8 character name of the dictionary you wish to report on.

- REQUESTED_SCHEMA (&&2)
  Enter the 1 to 18 character name of an SQL schema or use a like-predicate pattern to specify multiple schemas.

- REQUESTED_AM (&&3)
  Enter the 1 to 18 character name of an access module, or use a like-predicate pattern to specify multiple access modules.

Job Submission

The following example reports on all Access Modules defined for the JMA schema in TSTDICT.

```
DATABASE DICTNAME=dictionary-name
USE "QRPT004" WITH VALUES (REQUESTED_DICTIONARY='TSTDICT'
  REQUESTED_SCHEMA='JMA'
  REQUESTED_AM='%')
```
### Sample QREPORT 004:

<table>
<thead>
<tr>
<th>AM NAME</th>
<th>VERSION</th>
<th>SCHEMA</th>
<th>TIME OF CREATION</th>
<th>LENGTH</th>
<th>SCHEMA NAME</th>
<th>TABLE NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>DYNDIAL1</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>0</td>
<td>DEMO</td>
<td>TAB1</td>
</tr>
<tr>
<td>EXTCURS</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>0</td>
<td>DEMO</td>
<td>TAB1</td>
</tr>
<tr>
<td>EXTCURSA</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>0</td>
<td>DEMO</td>
<td>TAB1</td>
</tr>
<tr>
<td>JMAUG01</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>4,900</td>
<td>EMPLOYEE</td>
<td>EMPLOYEE</td>
</tr>
<tr>
<td>JNABAT02</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>4,900</td>
<td>EMPLOYEE</td>
<td>EMPLOYEE</td>
</tr>
<tr>
<td>JNABAT03</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>4,900</td>
<td>EMPLOYEE</td>
<td>EMPLOYEE</td>
</tr>
<tr>
<td>JNADYN1</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>0</td>
<td>EMPLOYEE</td>
<td>TAB1</td>
</tr>
<tr>
<td>JNAMEST</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>0</td>
<td>EMPLOYEE</td>
<td>EMPLOYEE</td>
</tr>
<tr>
<td>JNAPREP</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>4,800</td>
<td>EMPLOYEE</td>
<td>EMPLOYEE</td>
</tr>
<tr>
<td>JNAPREP4</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>0</td>
<td>EMPLOYEE</td>
<td>EMPLOYEE</td>
</tr>
<tr>
<td>JNASQL</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>0</td>
<td>EMPLOYEE</td>
<td>EMPLOYEE</td>
</tr>
<tr>
<td>JNASQLB</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>4,900</td>
<td>EMPLOYEE</td>
<td>EMPLOYEE</td>
</tr>
<tr>
<td>JNASQLD</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>0</td>
<td>EMPLOYEE</td>
<td>EMPLOYEE</td>
</tr>
<tr>
<td>JNASQLIV</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>0</td>
<td>INV</td>
<td>PART</td>
</tr>
<tr>
<td>JNASQL12</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>0</td>
<td>INV</td>
<td>COMPONENT</td>
</tr>
<tr>
<td>JNASQL2</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>4,900</td>
<td>EMPLOYEE</td>
<td>EMPLOYEE</td>
</tr>
<tr>
<td>JNASQL2A</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>4,900</td>
<td>EMPLOYEE</td>
<td>EMPLOYEE</td>
</tr>
<tr>
<td>JNASQL3B</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>4,900</td>
<td>EMPLOYEE</td>
<td>EMPLOYEE</td>
</tr>
<tr>
<td>PREPARE</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>0</td>
<td>EMPLOYEE</td>
<td>EMPLOYEE</td>
</tr>
<tr>
<td>TSTSQL</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>0</td>
<td>EMPLOYEE</td>
<td>EMPLOYEE</td>
</tr>
<tr>
<td>TSTSQLAM</td>
<td>1</td>
<td>JMA</td>
<td>yyyy-mm-dd-hh.mm</td>
<td>0</td>
<td>EMPLOYEE</td>
<td>EMPLOYEE</td>
</tr>
</tbody>
</table>

### Field Descriptions

A description of the fields in QREPORT 004 follows:

- **AM NAME**
  Specifies the name of the access module.

- **VERSION**
  Specifies the version number of the access module.

- **SCHEMA**
  Specifies the schema associated with the access module.

- **TIME OF CREATION**
  Specifies the time and date the access module was created.
CA IDMS - 19.0

- **LENGTH**
  Number of bytes in the access module.

- **TABLES ACCESSED**
  Identifies the tables accessed by this access module.
  - **SCHEMA NAME**
    Specifies the name of the schema to which the table being accessed belongs.
  - **TABLE NAME**
    Specifies the name of the table being accessed.

**QREPORT 005 -SQL Table Access Report**

**Contents**

- Symbolic Parameter Overrides (see page 114)
- Job Submission (see page 114)
- Field Descriptions (see page 115)

QREPORT 005, the SQL Table Access report, lists each SQL table referenced by an access module, along with the names of all access modules that reference it. The report is sorted by schema name, table name, access module name, and access module version.

Tables with no referencing access modules are omitted from this report.

**Symbolic Parameter Overrides**

- **REQUESTED_DICTIONARY (&&1)**
  Enter the 1 to 8 character name of the dictionary you wish to report on.

- **REQUESTED_SCHEMA (&&2)**
  Enter the 1 to 18 character name of an SQL schema, or use a *like-predicate pattern* to request more than one schema.

- **REQUESTED_TABLE (&&3)**
  Enter the 1 to 18 character name of an SQL table, or use a *like-predicate pattern* to request more than one table.

**Job Submission**

The following example lists all tables in all schemas of the TSTDICT dictionary that have referencing access modules.

The use of the DICTNAME clause on the DATABASE parameter allows CA Culprit for CA IDMS to retrieve QRPT005 syntax from the DIRLDICT dictionary.
Sample QREPORT 005:

<table>
<thead>
<tr>
<th>SCHEMA.TABLE NAME</th>
<th>AM NAME</th>
<th>VERSION</th>
<th>AM SCHEMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEMO.TAB1</td>
<td>DYNDIAL1</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td>EXTCURS</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td>EXTCURSA</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td>INSSDIAL</td>
<td>1</td>
<td>DEMO</td>
</tr>
<tr>
<td></td>
<td>JMADYN1</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td>DEMOEMPL.DBCSTAB2</td>
<td>DBCSQLA2</td>
<td>1</td>
<td>DEMOEMPL</td>
</tr>
<tr>
<td>DEMOEMPL.DBCSTAB3</td>
<td>DBCSQLA5</td>
<td>1</td>
<td>DEMOEMPL</td>
</tr>
<tr>
<td>DEMOEMPL.DIVISION</td>
<td>AMDCSQLT</td>
<td>1</td>
<td>DEMOEMPL</td>
</tr>
<tr>
<td>EMPDEND.EMPLOYEE</td>
<td>JMAUG01</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td>JMABAT02</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td>JMABAT93</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td>JMANEST</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td>JMARPREP</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td>JMARPREP4</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td>JMASQL</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td>JMASQLB</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td>JMASQLD</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td>JMASQL2</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td>JMASQL2A</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td>JMASQL30</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td>SQLDEMOD</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td>TSTSQL</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td>TSTSQLAM</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td>INV.COMPONENT</td>
<td>JMASQL12</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td>INV.PART</td>
<td>JMASQLIV</td>
<td>1</td>
<td>JMA</td>
</tr>
<tr>
<td></td>
<td>JMASQL12</td>
<td>1</td>
<td>JMA</td>
</tr>
</tbody>
</table>

Field Descriptions

A description of the fields in QREPORT 005 follows:

- **SCHEMA.TABLE NAME**
The name of the schema and the table.

- **ACCESS MODULES THAT REFERENCE TABLE**
  Specifies the AM Name, Version, and AM Schema used to reference the named table.
  - **AM NAME**
    Specifies the name of the access module.
  - **VERSION**
    Specifies the version number of the access module.
Symbolic Parameter Overrides

- **REQUESTED_DICTIONARY (&&1).**
  Enter the 1 to 8 character name of the SQL dictionary you wish to report on.

- **REQUESTED_SCHEMA (&&2).**
  Enter the 1 to 18 character name of an SQL schema, or use a like-predicate pattern to request more than one schema.

- **REQUESTED_TABLE (&&3).**
  Enter the 1 to 18 character name of an SQL table, or use a like-predicate pattern to request more than one table.

Job Submission

The following example lists syntax for all qualifying tables and views in the TSTDICT dictionary.

DATABASE DICTNAME=dictionary-name
USE "QRPT006" WITH VALUES (REQUESTED_DICTIONARY='TSTDICT'
  REQUESTED_SCHEMA='%'
  REQUESTED_TABLE='%')

Sample QREPORT 006:

<table>
<thead>
<tr>
<th>SCHEMA NAME</th>
<th>TABLE NAME</th>
<th>SYNTAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEMEMPL</td>
<td>EMPLOYEE</td>
<td>{ (EMP_ID &lt;= 8999) AND (STATUS IN ('A', 'S', 'L', 'T')) }</td>
</tr>
<tr>
<td>DEMEMPL</td>
<td>POSITION</td>
<td>{ (HOURLY_RATE IS NOT NULL AND SALARY_AMOUNT IS NULL) OR (HOURLY_RATE IS NULL AND SALARY_AMOUNT IS NOT NULL) }</td>
</tr>
</tbody>
</table>
Field Descriptions

A description of the fields in QREPORT 006 follows:

- **SCHEMA NAME**
The schema name associated with the table.

- **TABLE NAME**
The name of the table.

- **SYNTAX**
  Specifies the syntax used to create each SQL table defined with a check constraint, and each SQL view.

QREPORT 007 - SQL Table Index Report

Contents

- Symbolic Parameter Overrides (see page 118)
- Job Submission (see page 118)
- Field Descriptions (see page 119)

QREPORT 007, the SQL Table Index report, provides information on each index that is associated with the schema and table requested. For requests involving multiple schemas and tables, the report is sorted by schema name, table name, and index name.

Indexes with a name of HASH represent calc keys defined on a base table using the SQL CREATE CALC key syntax. On the final release of QRPT007, the word CALC will appear in column 9 instead of INDEX HASH.

The report layout is generated in the form of syntax which is similar to that used to create the index.
Symbolic Parameter Overrides

- **REQUESTED_DICTIONARY (&&1).**
  Enter the 1 to 8 character name of the SQL dictionary you wish to report on.

- **REQUESTED_SCHEMA (&&2).**
  Enter the 1 to 18 character name of an SQL schema, or use a *like-predicate pattern* to request more than one schema.

- **REQUESTED_TABLE (&&3).**
  Enter the 1 to 18 character name of an SQL table, or use a *like-predicate pattern* to request more than one table.

Job Submission

The following example will retrieve QRPT007 syntax from the DIRLDICT dictionary of IDD and will provide an index report of all tables in the SYSTEM schema of the TSTDICT dictionary.

```plaintext
DATABASE DICTNAME=dictionary-name
USE "QRPT007" WITH VALUES (REQUESTED_DICTIONARY='TSTDICT', REQUESTED_SCHEMA='SYSTEM', REQUESTED_TABLE='%')
```

Sample QREPORT 007:

<table>
<thead>
<tr>
<th>INDEX NAME</th>
<th>SCHEMA TABLE / INDEX COLUMNS</th>
<th>ORDER</th>
<th>OTHER SPECIFICATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIQUE INDEX</td>
<td>ON SYSTEM.AM</td>
<td>ASC, ASC</td>
<td>UNCOMPRESSED, CLUSTERED</td>
</tr>
<tr>
<td></td>
<td>(NAME, VERSION)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNIQUE INDEX</td>
<td>ON SYSTEM.AREA</td>
<td>ASC, ASC</td>
<td>UNCOMPRESSED, CLUSTERED</td>
</tr>
<tr>
<td></td>
<td>(NAME, SEGMENT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNIQUE INDEX</td>
<td>IX-AREA</td>
<td>ASC, ASC</td>
<td>UNCOMPRESSED, IN SYSCAT.DDLCATX</td>
</tr>
<tr>
<td></td>
<td>(NAME, SEGMENT)</td>
<td></td>
<td>INDEX BLOCK CONTAINS 10 KEYS</td>
</tr>
<tr>
<td></td>
<td>IN SYSCAT.DDLCATX</td>
<td></td>
<td>DISPLACEMENT IS 3 PAGES</td>
</tr>
<tr>
<td>UNIQUE INDEX</td>
<td>IX-BUFFER</td>
<td>ASC, ASC</td>
<td>UNCOMPRESSED, IN SYSCAT.DDLCATX</td>
</tr>
<tr>
<td></td>
<td>(NAME, DMCL)</td>
<td></td>
<td>INDEX BLOCK CONTAINS 10 KEYS</td>
</tr>
<tr>
<td></td>
<td>IN SYSCAT.DDLCATX</td>
<td></td>
<td>DISPLACEMENT IS 3 PAGES</td>
</tr>
<tr>
<td>UNIQUE INDEX</td>
<td>IX-CONSTRAINT</td>
<td>ASC, ASC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(SCHEMA)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Field Descriptions

A description of the fields in QREPORT 007 follows:

- **SCHEMA NAME**
  The schema which owns the indexes listed as follows.

- **INDEX NAME**
  The name of the index that was created.

- **SCHEMA.TABLE**
  The schema and table name for which this index is defined.

- **INDEX COLUMNS**
  Specifies the column(s) used to create the index key.

- **ORDER**
  Indicates the sort order of the values in the named column(s), ascending or descending.

- **OTHER SPECIFICATIONS**
  Lists other specifications used to create the index, such as compressed/uncompressed, clustered, displacement.

QREPORT 008 - SQL Table Constraint Report

Contents

- Symbolic Parameter Overrides (see page 120)
- Job Submission (see page 120)
- Field Descriptions (see page 121)

QREPORT 008, the SQL Table Constraint report, lists all of the referential constraints in which each table participates. This report is sorted first by the schema and table names that were requested on the WITH VALUES clause, then the constraint relationship, and finally the constraint name. Constraints where the table being reported is the referencing table are listed first, followed by constraints where the table being reported is the referencing table.
Two tables are involved in each referential constraint. The dependent table is called the **referencing table**. Data values for certain columns of the referencing table are restricted to only those values that exist in corresponding columns of the **table being referenced**. These columns in the referencing table are often referred to as **foreign key columns**; they correspond to **referenced columns** in the referenced table.

Constraints where the table being reported is the referencing table are listed first in QREPORT 008. Finally, QREPORT 008 lists the constraints where the table being reported is the table being referenced.

### Symbolic Parameter Overrides

- **REQUESTED_DICTIONARY (&&1).**
  Enter the 1 to 8 character name of the SQL dictionary you wish to report on.

- **REQUESTED_SCHEMA (&&2).**
  Enter the 1 to 18 character name of an SQL schema, or use a like-predicate pattern to request more than one schema.

- **REQUESTED_TABLE (&&3).**
  Enter the 1 to 18 character name of an SQL table, or use a like-predicate pattern to request more than one table.

### Job Submission

The following example retrieves QRPT008 syntax from the CULLIB copybook library, and produces a Table Constraint Report for those tables in SYSTEM schema of TSTDICT that begin with the letter D.

```
DATABASE DICTNAME=dictionary-name
USE "QRPT008" WITH VALUES (REQUESTED_DICTIONARY='TSTDICT',
                           REQUESTED_SCHEMA='SYSTEM',
                           REQUESTED_TABLE='D%')
```

**Sample QREPORT 008:**

```
<table>
<thead>
<tr>
<th>TABLE NAME</th>
<th>CONSTRAINT NAME</th>
<th>TABLE REFERENCED</th>
<th>REFERENCING TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM.DBNAME</td>
<td>DBTABLE-DBNAME</td>
<td>SYSTEM.DBTABLE</td>
<td></td>
</tr>
<tr>
<td>DBNAME-DBSEGMENT</td>
<td>DBNAME-DBSEGMENT</td>
<td>SYSTEM.DBSEGMENT</td>
<td></td>
</tr>
<tr>
<td>DBNAME-DBSSC</td>
<td>DBNAME-DBSSC</td>
<td>SYSTEM.DBSSC</td>
<td></td>
</tr>
<tr>
<td>SYSTEM.DBTABLE</td>
<td>DBTABLE-DBNAME</td>
<td></td>
<td>SYSTEM.DBNAME</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>Field Descriptions</th>
</tr>
</thead>
</table>

A description of the fields in QREPORT 008 follows:

- **TABLE NAME**
  The name of the table which participates in constraints.

- **CONSTRAINT NAME**
  The name of the referential constraint.

- **TABLE REFERENCED**
  Specifies the table being referenced.

- **REFERENCING TABLE**
  Specifies the dependent table in the constraint.
CA IDMS Extractor Reports

This section describes the Extract Audit Report and the Load Audit Report produced by CA IDMS Extractor. Included is an example of each report along with a brief description of each field contained in the report.

- CA IDMS Extractor Audit Reports (see page 122)
- Extract Audit Report (see page 123)
- Load Audit Report (see page 127)

CA IDMS Extractor Audit Reports

CA IDMS Extractor produces two audit reports:

- EXTRACT audit report
- LOAD audit report

Each report displays information on run times, input parameters, processing statistics, processing options, database statistics, and any processing errors that may have occurred. See Error! Reference source not found. for complete error message information.

Common Report Header Information

The header portion of the Extract and Load Audit report is identical. Load Audit Reports are shown below.

<table>
<thead>
<tr>
<th>ID</th>
<th>RELEASE</th>
<th>CA IDMS/DATABASE EXTRACTOR</th>
<th>DATE</th>
<th>TIME</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rnn.nn</td>
<td>EXTRACT AUDIT REPORT</td>
<td>mm/dd/yy</td>
<td>hh:mm:ss</td>
<td>nnnn</td>
<td></td>
</tr>
</tbody>
</table>

Field Descriptions

The table below describes each of the header fields.

<table>
<thead>
<tr>
<th>Report Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rnn.nn</td>
<td>The version of CA IDMS Extractor that was installed, where nn.nn represents the release number and the subrelease number.</td>
</tr>
<tr>
<td>Report Title</td>
<td>The report shown below is titled: EXTRACT AUDIT REPORT. It is produced by the Database Extract Component of CA IDMS Extractor. The Database Load Component of CA IDMS Extractor produces the LOAD AUDIT REPORT.</td>
</tr>
<tr>
<td>mm/dd</td>
<td>The date on which the report was run where mm indicates the month, dd indicates the day, /yy and yy indicates the year.</td>
</tr>
</tbody>
</table>
Report Field Description

<table>
<thead>
<tr>
<th>hh:mm:ss</th>
<th>The time at which the report was run where hh indicates the hour, mm indicates the minute, and ss indicates the second. The time is shown in 24 hour format.</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnn</td>
<td>The page number within the report.</td>
</tr>
</tbody>
</table>

**Extract Audit Report**

The following section contains an example of the Extract Audit Report and a description of the fields that appear below the header information.

**Processing Messages and Statistics**

<table>
<thead>
<tr>
<th>ID</th>
<th>RELEASE</th>
<th>CA IDMS/DATABASE EXTRACTOR</th>
<th>DATE</th>
<th>TIME</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rnn.nn</td>
<td>EXTRACT AUDIT REPORT</td>
<td>mm/dd/yy</td>
<td>hh:mm:ss</td>
<td>nnnn</td>
<td></td>
</tr>
</tbody>
</table>

Extracts:

- NKWP091I INPUT PARAMETER STATEMENT
- NKWP092I CONTINUATION STATEMENT
- PARM002I NO PARAMETER ERRORS DETECTED
- PSUB003I SOURCE PSUB STORAGE REQUIRED...18,072 BYTES
- PSUB003I TARGET PSUB STORAGE REQUIRED...18,072 BYTES
- RETS005I RETRIEVING SUBSCHEMA TESTSUB1; USING DICTNAME **NONE**, DICTNODE **NONE**
- XTRC001I USING SUBSCHEMA TESTSUB1; COMPILED mm/dd/yy hh:mm:ss;
- XTRC002I EXTRACT STEP STARTED
- XTRC003I STORAGE-POOL SIZE.........5,856 BYTES
- XTRC004I INITIATING RECORD BINDS
- XTRC006I BEGINNING SWEEP OF AREA DEPT-REGION
- XTRC007I END OF AREA SWEEP
- XTRC012I EXTRACT STEP ENDED
- XTRC013I EXTRACTS WRITTEN
- DBX PROCESSING EXTRACTS

**Parameter Field Descriptions**

<table>
<thead>
<tr>
<th>Report Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input parameter statement ruler</td>
<td>Indicates the 80 positions of the parameter statement. Enter all syntax in columns 1 through 72.</td>
</tr>
<tr>
<td>List of all parameter statements input for this execution of CA IDMS Extractor.</td>
<td></td>
</tr>
<tr>
<td>Report Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>INPUT PARAMETER STATEMENTS</td>
<td></td>
</tr>
<tr>
<td>SOURCE PSUB STORAGE REQUIRED</td>
<td>The amount of storage, in bytes, required for building the pseudo subschema table of the source subschema.</td>
</tr>
<tr>
<td>TARGET PSUB STORAGE REQUIRED</td>
<td>The amount of storage, in bytes, required for building the pseudo subschema table of the target subschema.</td>
</tr>
<tr>
<td>Informative message</td>
<td>Indicates that the Selection Criteria Specification named on the PROCESS parameter statement was successfully retrieved from the CA IDMS Extractor database.</td>
</tr>
<tr>
<td>USING SUBSCHEMA</td>
<td>The name of the source subschema used by the Database Extract Component.</td>
</tr>
<tr>
<td>EXTRACT STEP STARTED</td>
<td>The date and time at which the EXTRACT step was started in the format mm/dd/yy hh:mm:ss.</td>
</tr>
<tr>
<td>STORAGE POOL SIZE</td>
<td>The size of the storage pool, in bytes, required for the extract record and save stack.</td>
</tr>
<tr>
<td>INITIATING READIES</td>
<td>The names of areas in the source database that were successfully readied for processing. These areas are always readied in shared retrieval mode.</td>
</tr>
<tr>
<td>BEGINNING SWEEP</td>
<td>The date and time CA IDMS Extractor began walking an index in the format mm/dd/yy hh:mm:ss.</td>
</tr>
<tr>
<td>END OF AREA SWEEP</td>
<td>The date and time CA IDMS Extractor ended walking of an index in the format mm/dd/yy hh:mm:ss.</td>
</tr>
<tr>
<td>EXTRACT STEP ENDED</td>
<td>The date and time at which the EXTRACT step ended in the format mm/dd/yy hh:mm:ss.</td>
</tr>
<tr>
<td>EXTRACTS WRITTEN</td>
<td>The total number of extracts written.</td>
</tr>
<tr>
<td>DBX PROCESSING EXTRACTS</td>
<td>The number of extracts required by the EXTRACT step to describe the Specification's Path (or paths).</td>
</tr>
</tbody>
</table>

### Areas/Records/Sets Extracted

The following fields display the areas, records, and sets that participated in the extraction. See the example shown below.

```plaintext
ID  RELEASE  CA IDMS/DATABASE EXTRACT AUDIT REPORT  DATE    TIME    PAGE
Rnn.nn  EXTRACTS WRITTEN  mm/dd/yy  hh:mm:ss  nnnn

SPECIFICATION: DEPT-TEACHER
******************************************************************************
A R E A S / R E C O R D S / S E T S E X T R A C T E D ******************************************************************************
SUBSCHEMA....TESTSUB1 DICTNAME **NONE**; DICTNODE **NONE**; DBNAME STUDTCHR; DBNODE **NONE**
AREA....CLASS-REGION  NOSWEEP LDAPAGE (**99,001) HIPAGE(**99,030) FROM(***********) TO(***********)
SET.....CLASS-SCHEDULE  WALKED FROM CLASS - OWNER TO MEMBER
                    FROM(**********96,001) TO(**99,028) DBKEY(***********)
                    SKIP(**********) LIMIT(**********) PERSET(**********) NTHWR(**********)
                    LEVL(**********)
```
CA IDMS - 19.0

### Field Descriptions 2

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBSCHE</td>
<td>The name of the source subschema named in the specification or overridden in the PROCESS statement.</td>
</tr>
<tr>
<td>DICTNA</td>
<td>The name of the source dictionary named in the specification or overridden in the PROCESS statement.</td>
</tr>
<tr>
<td>DICTNOD</td>
<td>The name of the source node named in the specification or overridden in the PROCESS statement.</td>
</tr>
<tr>
<td>DBNAME</td>
<td>The name of the source database named in the specification or overridden in the PROCESS statement.</td>
</tr>
<tr>
<td>DBNODE</td>
<td>The name of the source database node named in the specification or overridden in the PROCESS statement.</td>
</tr>
<tr>
<td>AREA</td>
<td>Name of the area containing records that were extracted.</td>
</tr>
<tr>
<td>Area Sweep</td>
<td>If there were entry records in the area, the indicator reads SWEEP. If all entry records were retrieved by either direct DBKEY or CALC key, or there were no Entry Records in the area, the indicator reads NOSWEEP.</td>
</tr>
<tr>
<td>LOPAGE</td>
<td>Low page number of the area.</td>
</tr>
<tr>
<td>HIPAGE</td>
<td>High page number of the area.</td>
</tr>
<tr>
<td>FROM</td>
<td>Page number at which area sweeping began.</td>
</tr>
<tr>
<td>TO</td>
<td>Page number at which area sweeping ended.</td>
</tr>
<tr>
<td>Set Name</td>
<td>Name of a set included in a path that has an owner or member residing in the named area.</td>
</tr>
<tr>
<td>Access Method</td>
<td>The method by which the set, listed in the SET Name field, was accessed.</td>
</tr>
<tr>
<td>Record Name</td>
<td>Name of a record residing in the area listed in the Area field. The record was either an Entry Record or an owner or a member of a set included in a path.</td>
</tr>
<tr>
<td>Record Level</td>
<td>A summary of the data entered on the Record Level Selection Criteria screen for the record listed in the Record Name field. If a particular selection criteria was not entered for the record, all asterisks appear within the parentheses.</td>
</tr>
</tbody>
</table>
Extract Statistics

The following fields of the Extract Audit Report display statistical information. See the example shown below.

<table>
<thead>
<tr>
<th>ID</th>
<th>RELEASE</th>
<th>CA IDMS/DATABASE EXTRACTOR</th>
<th>DATE</th>
<th>TIME</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rnn.nn</td>
<td>EXTRACT AUDIT REPORT</td>
<td>mm/dd/yy</td>
<td>hh:mm:ss</td>
<td>nnnn</td>
<td></td>
</tr>
</tbody>
</table>

*****************************************************************************************************************
************************************** EXTRACT STATISTICS *************************************
*****************************************************************************************************************

PAGES ENTRY RECORD

<table>
<thead>
<tr>
<th>AREA NAME</th>
<th>SWEPT</th>
<th>NAME</th>
<th>RETRIEVED</th>
<th>EXTRACTED</th>
<th>SET NAME</th>
<th>NAME OCCURRENCES</th>
<th>EXTRACTED</th>
</tr>
</thead>
</table>

CLASS-REGION 0

CLASS-SCHEDULE

SCHEDULE 71 0

DEPT-REGION 15

DEPT 16 16

DEPT-TEACHER

TEACHER 18 18

DEPT-ACADEMIC

SUBJECT 126 126

SUBJECT-CLASS

CLASS 20 20

ROOM-CLASS

ROOM 20 0

** TOTALS **

15 16 16 415 324

Field Descriptions 3

<table>
<thead>
<tr>
<th>Report Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREA NAME</td>
<td>Name of the area containing records which were extracted.</td>
</tr>
<tr>
<td>PAGES SWEPT</td>
<td>Number of pages swept within the area listed in the AREA NAME field. The number is zero if: All entry records were retrieved by Direct DBKEY or CALC key. Entry into an area was only by Entry Index(es).</td>
</tr>
<tr>
<td>ENTRY RECOR D NAME</td>
<td>Name of Entry Record residing in the area listed in the AREA NAME field.</td>
</tr>
<tr>
<td></td>
<td>Number of entry records retrieved by area sweeping.</td>
</tr>
</tbody>
</table>
## Load Audit Report

A description of the fields appearing on the Load Audit Report is given below.

<table>
<thead>
<tr>
<th>Report Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTRY RECORDS REtrieved</td>
<td>Number of entry records retrieved that met the Record Level Selection Criteria. An asterisk next to this field indicates that the extraction record limit was reached.</td>
</tr>
<tr>
<td>ENTRY RECORDS EXTRACTED</td>
<td>Note: If a set is walked from member to owner, the number of RECORDS RETRIEVED may be greater than the number of actual record occurrences on the database because this value reflects the number of owners retrieved when walking from all member record occurrences. If a set occurrence has 10 member records and the owner was obtained for each of the 10 member records, RECORDS RETRIEVED reflects a count of 10 even though there was only one physical occurrence of the owner record type. The same situation exists for RECORDS EXTRACTED.</td>
</tr>
<tr>
<td>SET NAME</td>
<td>Name of a set in which the record occurrence listed in the OWNER/MEMBER RECORDS NAME field is a participant. The set was included in a path.</td>
</tr>
<tr>
<td>OWNER/MEMBER RECORDS NAME</td>
<td>The name of the record occurrence that is an owner or member of the set listed in the SET NAME field.</td>
</tr>
<tr>
<td>OWNER/MEMBER RECORD OCCURRENCES</td>
<td>Number of record occurrences retrieved when walking set occurrences owned by the record listed in the RECORD NAME field.</td>
</tr>
<tr>
<td>OWNER/MEMBER RECORDS EXTRACTED</td>
<td>Number of records retrieved which met their Record Level Selection Criteria. An asterisk next to this field indicates that the extraction record limit was reached. See the note above for the ENTRY RECORDS EXTRACTED field.</td>
</tr>
</tbody>
</table>
### Processing Messages

<table>
<thead>
<tr>
<th>Report Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOAD STEP STARTED</td>
<td>The date and time at which the LOAD step was started in the format mm/dd/yy and hh:mm:ss.</td>
</tr>
<tr>
<td>TARGET PSUB STORAGE REQUIRED</td>
<td>The amount of storage, in bytes, required for building the pseudo subschema table of the target subschema.</td>
</tr>
<tr>
<td>EXTRACTED /LOADING USING SUBSCHEMA</td>
<td>The name of the source subschemas used by the EXTRACT step, and the target subschema used by the LOAD step, respectively.</td>
</tr>
<tr>
<td>INITIATING READIES</td>
<td>The names of areas in the target database that were successfully readied for processing, and the ready mode.</td>
</tr>
<tr>
<td>Report Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>STORE STEP STARTED/STORE STEP ENDED</td>
<td>The date and time CA IDMS Extractor began/ended storing records on the target database in the format mm/dd/yy and hh:mm:ss.</td>
</tr>
<tr>
<td>TOTAL EXTRACTS READ</td>
<td>The total number of extracts read.</td>
</tr>
<tr>
<td>DBX PROCESSING EXTRACTS READ</td>
<td>The number of processing extracts required by the LOAD step for determining what records and sets participated in the extraction in addition to the header and trailer extracts.</td>
</tr>
<tr>
<td>EXTRACTS WRITTEN TO WORKFILE</td>
<td>The number of extracts required by the connect step.</td>
</tr>
<tr>
<td>RECORDS STORED</td>
<td>The number of records stored on the target database.</td>
</tr>
<tr>
<td>RECORDS W/DUP CALC KEYS</td>
<td>The number of CALC records that would have violated a DUPLICATES NOT ALLOWED restriction. This situation occurs only if you load a populated database.</td>
</tr>
<tr>
<td>CONNECT STEP STARTED/CONNECT STEP ENDED</td>
<td>The date and time at which CA IDMS Extractor began/ended connected member records into their set occurrences in the format mm/dd/yy and hh:mm:ss.</td>
</tr>
<tr>
<td>OWNER RECORDS OBTAINED</td>
<td>The number of set occurrences loaded that had member records.</td>
</tr>
<tr>
<td>MEMBER RECORDS CONNECTED</td>
<td>The number of member records connected.</td>
</tr>
<tr>
<td>MANUAL INDEX SET CONNECTIONS</td>
<td>The number of member records, with a set connect option of MANUAL, connected to system-owned, integrated indexes. The records were connected to the same index in the source database.</td>
</tr>
<tr>
<td>LOAD STEP ENDED</td>
<td>The date and time at which the LOAD step ended in the format mm/dd/yy and hh:mm:ss.</td>
</tr>
</tbody>
</table>
CA IDMS Module Listing

This section lists the following CA IDMS AREPORTs, CREPORTs, DREPORTs, IREPORTs, JREPORTs, and SREPORTs in order by module number.

- AREPORT Listing (see page 130)
- CREPORT Listing (see page 130)
- DREPORT Listing (see page 132)
- IREPORT Listing (see page 134)
- JREPORT Listing (see page 134)
- QREPORT Listing (see page 135)
- SREPORT Listing (see page 135)

AREPORT Listing

The following table lists AREPORTs by module number.

<table>
<thead>
<tr>
<th>AREPORT Module</th>
<th>Report Name</th>
<th>KEY Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>ADS Dialogs and Their Components -- Detail</td>
<td></td>
</tr>
<tr>
<td>002</td>
<td>ADS Dialogs and Their Components -- Key</td>
<td>KEY PROG-NAME-051 'dialog-name'</td>
</tr>
<tr>
<td>003</td>
<td>ADS Dialogs by Process Key</td>
<td>KEY MOD-NAME-067 'process-name'</td>
</tr>
<tr>
<td>004</td>
<td>ADS Dialogs by Record Key</td>
<td>KEY RSYN-NAME-079 'record-name'</td>
</tr>
<tr>
<td>005</td>
<td>ADS Dialogs by Subschema Key</td>
<td>KEY SS-NAM-026 'subschema-name'</td>
</tr>
<tr>
<td>006</td>
<td>ADS Dialogs by Map Key</td>
<td>KEY MAP-NAME-098 'map-name'</td>
</tr>
</tbody>
</table>

CREPORT Listing

The following table lists CREPORTs by module number.

<table>
<thead>
<tr>
<th>Module Number</th>
<th>Report Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>Network Description by Line(1)</td>
</tr>
<tr>
<td>002</td>
<td>Network Description by Physical Terminal(1)</td>
</tr>
<tr>
<td>003</td>
<td>Network Description by Logical Terminal(1)</td>
</tr>
<tr>
<td>004</td>
<td>Program Description(1)</td>
</tr>
<tr>
<td>005</td>
<td>Task Description(1)</td>
</tr>
<tr>
<td>006</td>
<td>Queue Description(1)</td>
</tr>
<tr>
<td>007</td>
<td>Destination Report(1)</td>
</tr>
<tr>
<td>011</td>
<td>System Options(1)</td>
</tr>
<tr>
<td>Module Number</td>
<td>Report Name</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>014</td>
<td>Network Description by Line(2)</td>
</tr>
<tr>
<td>015</td>
<td>Network Description by Physical Terminal(2)</td>
</tr>
<tr>
<td>016</td>
<td>Physical Terminals Within Line(2)</td>
</tr>
<tr>
<td>017</td>
<td>Network Description by Logical Terminal(2)</td>
</tr>
<tr>
<td>018</td>
<td>Logical Terminal by Physical Terminal(2)</td>
</tr>
<tr>
<td>019</td>
<td>Program Description(2)</td>
</tr>
<tr>
<td>020</td>
<td>Task Description(2)</td>
</tr>
<tr>
<td>021</td>
<td>Task Description Within Program(2)</td>
</tr>
<tr>
<td>022</td>
<td>Queue Description(2)</td>
</tr>
<tr>
<td>023</td>
<td>Queue Description Within Task(2)</td>
</tr>
<tr>
<td>024</td>
<td>Destination Report(2)</td>
</tr>
<tr>
<td>025</td>
<td>System Options(2)</td>
</tr>
<tr>
<td>028</td>
<td>Defined Messages</td>
</tr>
<tr>
<td>029</td>
<td>Defined Devices</td>
</tr>
<tr>
<td>030</td>
<td>Map Record Indices</td>
</tr>
<tr>
<td>031</td>
<td>Map Field Indices</td>
</tr>
<tr>
<td>032</td>
<td>Listing of Maps by Panel</td>
</tr>
<tr>
<td>033</td>
<td>Listing of Maps</td>
</tr>
<tr>
<td>034</td>
<td>Listing of Maps by Record Name</td>
</tr>
<tr>
<td>035</td>
<td>Listing of Maps by Element Name</td>
</tr>
<tr>
<td>040</td>
<td>ADS OnLine Report(1)</td>
</tr>
<tr>
<td>041</td>
<td>OLQ Report(1)</td>
</tr>
<tr>
<td>043</td>
<td>Listing of Nodes</td>
</tr>
<tr>
<td>044</td>
<td>Listing of Defined Resources</td>
</tr>
<tr>
<td>045</td>
<td>ADS OnLine Report(2)</td>
</tr>
<tr>
<td>046</td>
<td>OLQ Report(2)</td>
</tr>
<tr>
<td>047</td>
<td>SQL CACHE Report(2)</td>
</tr>
<tr>
<td>048</td>
<td>SQL CACHE Report(1)</td>
</tr>
<tr>
<td>050</td>
<td>Load Area Modules</td>
</tr>
<tr>
<td>051</td>
<td>Module Text to Output File Utility</td>
</tr>
<tr>
<td>052</td>
<td>Module Text to Output File Utility</td>
</tr>
<tr>
<td>053</td>
<td>Symbol Table Report</td>
</tr>
</tbody>
</table>

**Note:**

(1) Object reports  (2) Source reports
DREPORT Listing

The following table lists DREPORTs by module number.

<table>
<thead>
<tr>
<th>Module Number</th>
<th>Report Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Housekeeping Module</td>
</tr>
<tr>
<td>001</td>
<td>Class Report -- Detail</td>
</tr>
<tr>
<td>002</td>
<td>Attribute Report -- Detail</td>
</tr>
<tr>
<td>003</td>
<td>System Report -- Detail</td>
</tr>
<tr>
<td>004</td>
<td>User Report -- Detail</td>
</tr>
<tr>
<td>005</td>
<td>Program Report -- Detail</td>
</tr>
<tr>
<td>006</td>
<td>Module Report -- Detail</td>
</tr>
<tr>
<td>007</td>
<td>File Report -- Detail</td>
</tr>
<tr>
<td>008</td>
<td>Record Report -- Detail</td>
</tr>
<tr>
<td>009</td>
<td>Element Report -- Detail</td>
</tr>
<tr>
<td>010</td>
<td>Inactive Element Report -- Detail</td>
</tr>
<tr>
<td>011</td>
<td>Task Report -- Detail</td>
</tr>
<tr>
<td>012</td>
<td>Queue Report -- Detail</td>
</tr>
<tr>
<td>013</td>
<td>Destination Report -- Detail</td>
</tr>
<tr>
<td>014</td>
<td>Logical Terminal Report -- Detail</td>
</tr>
<tr>
<td>015</td>
<td>Physical Terminal Report -- Detail</td>
</tr>
<tr>
<td>016</td>
<td>Line Report -- Detail</td>
</tr>
<tr>
<td>017</td>
<td>Panel Report -- Detail</td>
</tr>
<tr>
<td>018</td>
<td>Map Report -- Detail</td>
</tr>
<tr>
<td>019</td>
<td>User-Defined Entity Report -- Detail</td>
</tr>
<tr>
<td>020</td>
<td>File/Record Cross-Reference Report</td>
</tr>
<tr>
<td>021</td>
<td>File Synonym Cross-Reference Report</td>
</tr>
<tr>
<td>022</td>
<td>Record Synonym Cross-Reference Report</td>
</tr>
<tr>
<td>023</td>
<td>Element Synonym Reference Report</td>
</tr>
<tr>
<td>024</td>
<td>Element Description Cross-Reference Report</td>
</tr>
<tr>
<td>025</td>
<td>Element Designator Cross-Reference Report</td>
</tr>
<tr>
<td>026</td>
<td>File Activity Report</td>
</tr>
<tr>
<td>027</td>
<td>IDMS Set Activity Report</td>
</tr>
<tr>
<td>028</td>
<td>IDMS Record Activity Report</td>
</tr>
<tr>
<td>029</td>
<td>IDMS Area Activity Report</td>
</tr>
<tr>
<td>030</td>
<td>Element/Program Cross-Reference Report</td>
</tr>
<tr>
<td>038</td>
<td>Record/Attribute Report -- Key</td>
</tr>
<tr>
<td>039</td>
<td>Element/Attribute Report -- Detail</td>
</tr>
<tr>
<td>Module Number</td>
<td>Report Name</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>050</td>
<td>Level Number Report</td>
</tr>
<tr>
<td>051</td>
<td>Module Text to Card Utility</td>
</tr>
<tr>
<td>052</td>
<td>Module Text to Output File Utility</td>
</tr>
<tr>
<td>053</td>
<td>System Report -- Summary</td>
</tr>
<tr>
<td>054</td>
<td>User Report -- Summary</td>
</tr>
<tr>
<td>055</td>
<td>Program Report -- Summary</td>
</tr>
<tr>
<td>056</td>
<td>Module Report -- Summary</td>
</tr>
<tr>
<td>057</td>
<td>File Report -- Summary</td>
</tr>
<tr>
<td>058</td>
<td>Record Report -- Summary</td>
</tr>
<tr>
<td>059</td>
<td>Element Report -- Summary</td>
</tr>
<tr>
<td>061</td>
<td>Task Report -- Summary</td>
</tr>
<tr>
<td>062</td>
<td>Queue Report -- Summary</td>
</tr>
<tr>
<td>063</td>
<td>Destination Report -- Summary</td>
</tr>
<tr>
<td>064</td>
<td>Logical Terminal Report -- Summary</td>
</tr>
<tr>
<td>065</td>
<td>Physical Terminal Report -- Summary</td>
</tr>
<tr>
<td>066</td>
<td>Line Report -- Summary</td>
</tr>
<tr>
<td>067</td>
<td>Panel Report -- Summary</td>
</tr>
<tr>
<td>068</td>
<td>Map Report -- Summary</td>
</tr>
<tr>
<td>071</td>
<td>Class Report -- Key</td>
</tr>
<tr>
<td>072</td>
<td>Attribute Report -- Key</td>
</tr>
<tr>
<td>073</td>
<td>System Report -- Key</td>
</tr>
<tr>
<td>074</td>
<td>User Report -- Key</td>
</tr>
<tr>
<td>075</td>
<td>Program Report -- Key</td>
</tr>
<tr>
<td>076</td>
<td>Module Report -- Key</td>
</tr>
<tr>
<td>077</td>
<td>File Report -- Key</td>
</tr>
<tr>
<td>078</td>
<td>Record Report -- Key</td>
</tr>
<tr>
<td>079</td>
<td>Element Report -- Key</td>
</tr>
<tr>
<td>081</td>
<td>Task Report -- Key</td>
</tr>
<tr>
<td>082</td>
<td>Queue Report -- Key</td>
</tr>
<tr>
<td>083</td>
<td>Destination Report -- Key</td>
</tr>
<tr>
<td>084</td>
<td>Logical Terminal Report -- Key</td>
</tr>
<tr>
<td>085</td>
<td>Physical Terminal Report -- Key</td>
</tr>
<tr>
<td>086</td>
<td>Line Report -- Key</td>
</tr>
<tr>
<td>087</td>
<td>Panel Report -- Key</td>
</tr>
<tr>
<td>088</td>
<td>Map Report -- Key</td>
</tr>
<tr>
<td>089</td>
<td>User-Defined Entity Report -- Key</td>
</tr>
</tbody>
</table>
### IREPORT Listing

The following table lists IREPORTs by module number.

<table>
<thead>
<tr>
<th>Module Number</th>
<th>Report Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>Row Level Security Summary Report</td>
</tr>
<tr>
<td>002</td>
<td>Row Level Security Detail Report</td>
</tr>
<tr>
<td>003</td>
<td>Row Level Security Summary Report by User</td>
</tr>
<tr>
<td>004</td>
<td>Row Level Security Detail Report by User</td>
</tr>
<tr>
<td>005</td>
<td>Row Level Security Summary Report by Owner/Security Name</td>
</tr>
</tbody>
</table>

### JREPORT Listing

The following table lists JREPORTs by module number.

<table>
<thead>
<tr>
<th>Module Number</th>
<th>Report Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Housekeeping Module</td>
</tr>
<tr>
<td>001</td>
<td>Transaction Summary</td>
</tr>
<tr>
<td>002</td>
<td>Program Termination Statistics</td>
</tr>
<tr>
<td>003</td>
<td>Program I/O Statistics</td>
</tr>
<tr>
<td>004</td>
<td>Program Summary</td>
</tr>
<tr>
<td>005</td>
<td>Detail Area/Transaction</td>
</tr>
<tr>
<td>006</td>
<td>Detail Program/Area</td>
</tr>
<tr>
<td>007</td>
<td>Area Summary</td>
</tr>
<tr>
<td>008</td>
<td>Formatted Record Dump</td>
</tr>
<tr>
<td>009</td>
<td>User ID</td>
</tr>
<tr>
<td>010</td>
<td>External User Identity</td>
</tr>
</tbody>
</table>
### QREPORT Listing

The following table lists QREPORTs by module number.

<table>
<thead>
<tr>
<th>Module Number</th>
<th>Report Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>SQL Column Name Report</td>
</tr>
<tr>
<td>002</td>
<td>SQL Table Information Report</td>
</tr>
<tr>
<td>003</td>
<td>SQL Schema Information Report</td>
</tr>
<tr>
<td>004</td>
<td>SQL Access Module Information Report</td>
</tr>
<tr>
<td>005</td>
<td>SQL Table Access Report</td>
</tr>
<tr>
<td>006</td>
<td>SQL Table Syntax Report</td>
</tr>
<tr>
<td>007</td>
<td>SQL Table Index Report</td>
</tr>
<tr>
<td>008</td>
<td>SQL Table Constraint Report</td>
</tr>
</tbody>
</table>

### SREPORT Listing

The following table lists SREPORTs by module number.

<table>
<thead>
<tr>
<th>Module Number</th>
<th>Report Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Housekeeping Module</td>
</tr>
<tr>
<td>001</td>
<td>IDMS Statistics - Histogram Report</td>
</tr>
<tr>
<td>003</td>
<td>IDMS DC System Statistics</td>
</tr>
<tr>
<td>005</td>
<td>IDMS DC Statistics by User Id</td>
</tr>
<tr>
<td>006</td>
<td>IDMS DC Statistics by Lterm Id</td>
</tr>
<tr>
<td>007</td>
<td>IDMS DC Task Statistics by Task Code</td>
</tr>
<tr>
<td>008</td>
<td>IDMS DC ERUS Task Statistics by Accounting Data</td>
</tr>
<tr>
<td>009</td>
<td>IDMS DC ERUS Task Statistics by Program</td>
</tr>
<tr>
<td>010</td>
<td>IDMS DC Transactions Statistics by User Id</td>
</tr>
<tr>
<td>011</td>
<td>IDMS DC Transaction Statistics by Lterm Id</td>
</tr>
<tr>
<td>012</td>
<td>IDMS DC Task Summary</td>
</tr>
<tr>
<td>013</td>
<td>IDMS DC Program Summary</td>
</tr>
<tr>
<td>014</td>
<td>IDMS DC Queue Summary</td>
</tr>
<tr>
<td>015</td>
<td>IDMS DC Line Summary</td>
</tr>
<tr>
<td>016</td>
<td>IDMS DC Physical Terminal Summary</td>
</tr>
<tr>
<td>Module Number</td>
<td>Report Name</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>017</td>
<td>Summary of Records Read</td>
</tr>
<tr>
<td>018</td>
<td>ADS OnLine Statistics by User Id</td>
</tr>
<tr>
<td>019</td>
<td>ADS OnLine Statistics by Dialog and Version Number</td>
</tr>
<tr>
<td>020</td>
<td>ADS OnLine Statistics by Logical Terminal Id</td>
</tr>
<tr>
<td>021</td>
<td>IDMS DC Transaction Statistics by Dialog</td>
</tr>
<tr>
<td>099</td>
<td>Output File of Archived Statistics Records</td>
</tr>
</tbody>
</table>
CA IDMS reports z/VSE JCL

Running in Local Mode

JCL for executing CA IDMS reports in local mode is shown below:

```
CULPRIT (z/VSE)
// JOB     CULPRIT
// DLBL  idmsnnn,'idmsnnn.library'// EXTENT SYSnnn,nnnnnn,sssllll// ASSGN SYSnnn
   DISK,VOL=nnnnnn,SHR
// LIBDEF *,SEARCH=CA IDMS. nn.n libraries// EXEC PROC=IDMSLBLS
:// OPTION LOG,CATAL,DUMP
   // ASSGN SYS009,IGN
   // ASSGN SYS001,nnnnnn,sssllll
   // DLBL SORTWK1,'WORK',0
   // EXTENT SYS001,nnnnnn,sssllll
   // ASSGN SYS004,X'ppp'
   // ASSGN SYS005,X'ddd'
   // DLBL SYS005,'PARMS',0
   // EXTENT SYS005,nnnnnn,sssllll
   // ASSGN SYS006,X'ddd'
   // DLBL SYS006,'EXTRACT',0
   // EXTENT SYS006,nnnnnn,sssllll
   // ASSGN SYS007,X'ddd'
   // DLBL SYS007,'SORTCARD',0
   // EXTENT SYS007,nnnnnn,sssllll
   // ASSGN SYS008,X'ddd'
   // DLBL SYS008,'NOSORT',0
   // EXTENT SYS008,nnnnnn,sssllll
   // ASSGN SYS002,X'ttt'
   // TLBL SYS002,'user.keyfile'
   // ASSGN SYS010,X'ttt'
   // TLBL SYS010,'user.inputfil'
   // ASSGN SYSPCH,X'ccc'
   // ASSGN SYS020,X'ttt'
   // TLBL SYS020,'user.nonprint',15
// EXEC CULPRITCULPRIT parameters/*user input file, if on cards/*optional restart
   parameter/*
   /*
   idmsnnn  dtfname of the CA IDMS library
   n
   'idmsnnn
   library'
   data set name of CA IDMS libraries, as established during installation

   SYSnnn  Logical unit of the volume for which the extent is effective

   ccc  device assignment (channel and unit) for punched output (required for DREPORT 051 and
   CREPORT 051)
   culprit.  file-id of system library that contains parameters to be copied (ASSGN necessary if USE,
   srclib =COPY, or =MACRO is used; DLBL and EXTENT necessary if the parameters are maintained on
   a private library)
   ddd  device assignment (channel and unit) for disk files
```
IDMSLB Name of the procedure provided at installation that contains the file definitions for CA IDMS
LS
  dictionary, database, disk journal, and SYSIDMS file definition
  Note: For a complete listing of IDMSLBLS, see IDMSLBLS Procedure (https://docops.ca.com
/display/IDMS19/IDMSLBLS+Procedure) later in this section

<table>
<thead>
<tr>
<th>Illl</th>
<th>number of tracks assigned for file</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnnnnn</td>
<td>serial number for disk storage device</td>
</tr>
<tr>
<td>ppp</td>
<td>device assignment (channel and unit) for printed output</td>
</tr>
<tr>
<td>sssss</td>
<td>starting track of file</td>
</tr>
<tr>
<td>SYS009</td>
<td>logical unit of the CA IDMS tape journal file</td>
</tr>
<tr>
<td>SYS020</td>
<td>logical unit for first tape or disk output file</td>
</tr>
</tbody>
</table>
| ttt | device assignment (channel and unit) for tape files (files may use disk instead of tape, in
which case a device assignment, DLBL and EXTENT information are also required) |
| nnnnnn | volume serial number |
| user. | file-id of the input file (required for JREPORTS and SREPORTS) |
| inputfil | |
| user. | file-id of the key file (necessary only if a key file is input |
| keyfile | |
| user. | file-id and retention period for nonprint/nonpunch output (required for SREPORT 099, |
| nonpri
nt,15 | |
|  | |

Central Version Modifications

To run CA IDMS reports under the central version, add a SYSCTL file to your job control; for example:

```plaintext
// ASSGN SYSnnn,DISK,VOL=vvvvvvv,SHR
// DLBL SYSCTL,'idms.sysctl',2099/365,SD
// EXTENT SYSnnn,nnnnnnn,,ssss,2
```

IDMSLBLS Procedure Listing

```plaintext
* LIBDEFS
  // LIBDEF *,SEARCH=idmslib.sublib
  // LIBDEF *,CATALOG=user.sublib
  */ LABELS
  // DLBL idmslib,'idms.library',2099/365
  // EXTENT ,nnnnnnn,,ssss,1500
  // DLBL dccat,'idms.system.dccat',2099/365,DA
  // EXTENT SYSnnn,nnnnnnn,,ssss,31
  // ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
  // DLBL dccatl,'idms.system.dccatlod',2099/365,DA
  // EXTENT SYSnnn,nnnnnnn,,ssss,6
  // ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
  // DLBL dccatx,'idms.system.dccatx',2099/365,DA
  // EXTENT SYSnnn,nnnnnnn,,ssss,11
  // ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
  // DLBL dcdml,'idms.system.dcdml',2099/365,DA
  // EXTENT SYSnnn,nnnnnnn,,ssss,101
  // ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
  // DLBL dclod,'idms.system.dclod',2099/365,DA
```
// EXTENT SYSnnn,nnnnnnn,ssss,21
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL dclog,'idds.system.ddldclog',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,401
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL dcrun,'idds.system.ddldcrun',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,68
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL dcsr,'idds.system.ddldcsr',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,135
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL dcmgs,'idds.sysmsg.ddldcmgs',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,281
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL dlscrr,'idds.sysloc.ddldcmsg',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,6
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL dirldb,'idds.sysdirl.ddldml',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,201
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL dirllod,'idds.sysdirl.ddldclod',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,2
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL empdemo,'idds.empdemo1',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,6
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL insdemo,'idds.insdemo1',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,6
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL orgdemo,'idds.orgdemo1',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,6
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL emldem,'idds.sqldemo.empldemo',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,11
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL infodem,'idds.sqldemo.infodemo',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,6
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL projdem,'idds.projseg projdemo',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,6
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL indxdem,'idds.sqldemo.indxdemo',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,6
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL sysctl,'idds.sysctl',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,2
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL secdd,'idds.sysuser.ddlsec',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,26
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL dclldb,'idds.appdict.ddldml',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,51
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL dclld,'idds.appdict.ddldclod',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,51
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL sqldd,'idds.syssql.ddlcat',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,101
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL sqllod,'idds.syssql.ddlcatl',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,51
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL sqlxdd,'idds.syssql.ddlcatx',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,26
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL asfdml,'idds.asfdict.ddldml',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,481
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL asflod,'idds.asfdict.asflod',2009/365,DA
// EXTENT SYSnnn,nnnnnnn,ssss,201
// ASSGN SYSnnn,DISK,VOL=nnnnnnn,SHR
// DLBL asfdemo,'idds.asfdict.asfdata',2009/365,DA
idmslib.sublib  Name of the sublibrary within the library containing CA IDMS modules
user.sublib   Name of the sublibrary within the library containing user modules
idmslib      Filename of the file containing CA IDMS modules
idms.library File-ID associated with the file containing CA IDMS modules
SYSnnn      Logical unit of the volume for which the extent is effective
nnnnnn      Volume serial identifier of appropriate disk volume
ssss        Starting track (CKD) or block (FBA) of disk extent
dccat       Filename of the system dictionary catalog (DDLCAT) area
idms.system.dccat File-ID of the system dictionary catalog (DDLCAT) area
dccatl      Filename of the system dictionary catalog load (DDLCATLOAD) area
idms.system.dccatlod File-ID of the system dictionary catalog load (DDLCATLOAD) area
dccatx      Filename of the system dictionary catalog index (DDLCATX) area
idms.system.dccatx File-ID of the system dictionary catalog index (DDLCATX) area
dcdml       Filename of the system dictionary definition (DDLDMML) area
idms.system.ddldml File-ID of the system dictionary definition (DDLDMML) area
dclod       Filename of the system dictionary definition load (DD DLCLOD) area
idms.system.ddldclod File-ID of the system dictionary definition load (DD DLCLOD) area
dclog       Filename of the system log area (DDLDCLOG) area
idms.system.ddldclog File-ID of the system log (DDLDCLOG) area
dcrun       Filename of the system queue (DDLDCRUN) area
idms.system.ddldcrun File-ID of the system queue (DDLDCRUN) area
dcscr       Filename of the system scratch (DDLDCSCR) area
idms.system.ddldcscr File-ID of the system scratch (DDLDCSCR) area
dcmsg       Filename of the system message (DDLDCMSG) area
idms.sysmsg.ddldcmsg File-ID of the system message (DDLDCMSG) area
dclscr      Filename of the local mode system scratch (DDLOCS) area
idms.sysloc.ddlocscr File-ID of the local mode system scratch (DDLOCS) area
<table>
<thead>
<tr>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dirldb</td>
<td>Filename of the IDMSDIRL definition (DDLDML) area</td>
</tr>
<tr>
<td>idms.sysdir.ddldml</td>
<td>File-ID of the IDMSDIRL definition (DDLDML) area</td>
</tr>
<tr>
<td>dirlldod</td>
<td>Filename of the IDMSDIRL definition load (DDLDCLOD) area</td>
</tr>
<tr>
<td>idms.sysdir.dirlod</td>
<td>File-ID of the IDMSDIRL definition load (DDLDCLOD) area</td>
</tr>
<tr>
<td>empdemo</td>
<td>Filename of the EMPDEMO area</td>
</tr>
<tr>
<td>idms.empdemo1</td>
<td>File-ID of the EMPDEMO area</td>
</tr>
<tr>
<td>insdemo</td>
<td>Filename of the INSDEMO area</td>
</tr>
<tr>
<td>idms.insdemo1</td>
<td>File-ID of the INSDEMO area</td>
</tr>
<tr>
<td>orgdemo</td>
<td>Filename of the ORGDEMO area</td>
</tr>
<tr>
<td>idms.orgdemo1</td>
<td>File-ID of the ORGDEMO area</td>
</tr>
<tr>
<td>empldem</td>
<td>Filename of the EMPPLDEMO area</td>
</tr>
<tr>
<td>idms.sqldemo.empldemo</td>
<td>File-ID of the EMPPLDEMO area</td>
</tr>
<tr>
<td>infodemo</td>
<td>Filename of the INFODEMO area</td>
</tr>
<tr>
<td>idms.sqldemo.infodemo</td>
<td>File-ID of the INFODEMO area</td>
</tr>
<tr>
<td>projdem</td>
<td>Filename of the PROJDEMO area</td>
</tr>
<tr>
<td>idms.projseg.projdemo</td>
<td>File-ID of the PROJDEMO area</td>
</tr>
<tr>
<td>indxdem</td>
<td>Filename of the INDXDEMO area</td>
</tr>
<tr>
<td>idms.sqldemo.indxdemo</td>
<td>File-ID of the INDXDEMO area</td>
</tr>
<tr>
<td>sysctl</td>
<td>Filename of the SYSCTL file</td>
</tr>
<tr>
<td>idms.sysctl</td>
<td>File-ID of the SYSCTL file</td>
</tr>
<tr>
<td>secdd</td>
<td>Filename of the system user catalog (DDLSEC) area</td>
</tr>
<tr>
<td>idms.sysuser.ddlsec</td>
<td>File-ID of the system user catalog (DDLSEC) area</td>
</tr>
<tr>
<td>dictdb</td>
<td>Filename of the application dictionary definition area</td>
</tr>
<tr>
<td>idms.appldict.ddldml</td>
<td>File-ID of the application dictionary definition (DDLDML) area</td>
</tr>
<tr>
<td>dloddb</td>
<td>Filename of the application dictionary definition load area</td>
</tr>
<tr>
<td>idms.appldict.ddldclod</td>
<td>File-ID of the application dictionary definition load (DDLDCLOD) area</td>
</tr>
<tr>
<td>sqldd</td>
<td>Filename of the SQL catalog (DDLCAT) area</td>
</tr>
<tr>
<td>idms.syssql.ddlcat</td>
<td>File-ID of the SQL catalog (DDLCAT) area</td>
</tr>
<tr>
<td>sqllod</td>
<td>Filename of the SQL catalog load (DDLCATL) area</td>
</tr>
<tr>
<td>idms.syssql.ddlcatl</td>
<td>File-ID of SQL catalog load (DDLCATL) area</td>
</tr>
<tr>
<td>sqlxdd</td>
<td>Filename of the SQL catalog index (DDLCATX) area</td>
</tr>
<tr>
<td>idms.syssql.ddlcax</td>
<td>File-ID of the SQL catalog index (DDLCATX) area</td>
</tr>
<tr>
<td>asfdml</td>
<td>Filename of the asf dictionary definition (DDLDML) area</td>
</tr>
<tr>
<td>idms.asfdict.ddldml</td>
<td>File-ID of the asf dictionary definition (DDLDML) area</td>
</tr>
<tr>
<td>asflod</td>
<td>Filename of the asf dictionary definition load (ASFLOD) area</td>
</tr>
<tr>
<td>idms.asfdict.asflod</td>
<td>File-ID of the asf dictionary definition load (ASFLOD) area</td>
</tr>
<tr>
<td>asfdata</td>
<td>Filename of the asf data (ASFDATA) area</td>
</tr>
<tr>
<td>Filename</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>idms.asfdict.asfdata</td>
<td>File-ID of the asf data area (ASFDATA) area</td>
</tr>
<tr>
<td>ASFDEFN</td>
<td>Filename of the asf data definition (ASFDEFN) area</td>
</tr>
<tr>
<td>idms.asfdict.asfdefn</td>
<td>File-ID of the asf data definition area (ASFDEFN) area</td>
</tr>
<tr>
<td>j1jrnl</td>
<td>Filename of the first disk journal file</td>
</tr>
<tr>
<td>idms.j1jrnl</td>
<td>File-ID of the first disk journal file</td>
</tr>
<tr>
<td>j2jrnl</td>
<td>Filename of the second disk journal file</td>
</tr>
<tr>
<td>idms.j2jrnl</td>
<td>File-ID of the second disk journal file</td>
</tr>
<tr>
<td>j3jrnl</td>
<td>Filename of the third disk journal file</td>
</tr>
<tr>
<td>idms.j3jrnl</td>
<td>File-ID of the third disk journal file</td>
</tr>
<tr>
<td>sysidms parms</td>
<td>Filename of the SYSIDMS parameter file</td>
</tr>
</tbody>
</table>
Compliance Reporting

The following are considerations for journal record processing:

- If you want to capture information on retrieval only transactions, you must specify JOURNAL RETRIEVAL. While this will allow you to see what programs were used for retrieval activity, you will not be able to see what records were viewed.

- If you want to capture information from local batch jobs, you must capture the journal records (many sites take a backup of their database for recovery purposes when running a local mode update job rather than writing the local mode journal file.)

- Many web-based applications capture the user ID on the client side of the application and use a generic user ID to access the data from the database. In these cases, the actual user ID as captured on the client side can be captured and reported if a product like SiteMinder is used on the client side to make the actual user ID available to the backend system for recording in the BGIN journal checkpoint record.

Example 1

In this example, JREPORT 009 is used to determine who used CA IDMS DMLO to make updates to the database. The SELECT parameter selects only those BGIN records that have USDMAIN0 as the program name.

```
INPUT 19068 19068 UM(CULLJRNL)
JREPORT=009
SELECT PROGRAM EQ &sdq.USDMAIN0&sdq.
```

Example 2

Having run JREPORT 009, you found that CA IDMS DMLO was used to update the database, and the Transaction ID was 44568. You can find out what changes were made by running JREPORT 008. The SELECT parameter selects all the journal records for Transaction ID 44568.

```
INPUT 19068 19068 UM(CULLJRNL)
JREPORT=008
SELECT TRANSACT-ID EQ 44568
```

Example 3

You have a web-based application that uses a generic user ID to sign on to the backend CA IDMS system. You need to know the identities of the actual users of that web-based application. You can find this information out by running JREPORT 010 using a SELECT parameter to select only those BGIN journal records that contain the generic user ID used by that web-based application.

```
INPUT 19068 19068 UM(CULLJRNL)
JREPORT=010
SELECT TYPE EQ 'BGIN' AND USER-ID EQ 'WEB-USER'
```
DB Analyzer System Output Reports

The CA IDMS/DB Analyzer produces individual reports for database areas, record types, set types, and SR8 index sets. Statistics for these reports are drawn from a statistical file that you specify by using parameter statements. This option also produces reports that compare the data generated from a database structure at two different times. This section explains all reports in detail.

Important statistical information on the physical characteristics of four database structures--areas, record types, set types, and index sets--is collected by CA IDMS/DB Analyzer. This information is provided through individual and comparative reports. See Section 3, "Parameters".

Individual Reports

You can produce an individual report for every area, record type, set type, and index set in your database, provided you have created a CA IDMS/DB Analyzer statistical file containing the information necessary for each report.

Comparative Reports

In addition, you can produce a comparative report for each of the four structures (area, record type, set type, and index set). A comparative report is similar to an individual report, except that it shows information on a particular structure at two different times. Consequently, CA IDMS/DB Analyzer must draw statistics from two different statistical files to produce this type of report. A comparative report also displays changes between new statistical values and old statistical values.

This section describes both individual and comparative Area, Record, Set, and SR8 Index Reports. A description of each comparative report follows that of its corresponding individual report. For example, the Comparative Area Report description follows the Area Report description. When reading this section, be sure you are familiar with the description of an individual report before reading about its corresponding comparative report. For more information, see the following topics:

- Area Report (see page 145)
- Comparative Area Report (see page 157)
- Record Report (see page 164)
- Comparative Record Report (see page 171)
- Set Report (see page 176)
- Comparative Set Report (see page 185)
- SR8 Index Report (see page 192)
- Comparative SR8 Index Report (see page 197)
Area Report

You can produce an Area Report for every area, or portion of an area, in a database for which statistics have been accumulated. You cannot produce reports for areas, or portions of areas, excluded from accumulation by an AREA parameter statement. An Area Report is divided into three sections: the Area Statistics Section, the Distribution Section, and the Analysis by Record Section.

Area Report fields are shown in following screen displays. In this section are descriptions of two report identification fields plus fields used in the Area Statistics, Distribution, Histogram, and Analysis by Record Sections of the Area Report. The item numbers preceding each field name and description correspond with those in the referenced exhibits. When a percentage is given, the derivation is shown. The percentage is calculated by dividing the value indicated in the numerator by the value indicated in the denominator and multiplying the result by 100 (the asterisk indicates multiplication). The item number for each value used in calculating the percentage is given in parentheses after the name of the item.

Contents

- Area Report Field Descriptions (see page 145)
- Distribution Section Fields (see page 151)
- Histogram Section Fields (see page 154)
  - Analysis by Record Section Fields (see page 156)

Area Report Field Descriptions

The Area Statistic Section of the Area Report is shown in following two screen displays. Descriptions of the Identification and Statistic Section fields are provided following the exhibits.

<table>
<thead>
<tr>
<th>CA-TOOLS</th>
<th>RELEASE</th>
<th>CA IDMS/DB ANALYZER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rnn.nn</td>
<td></td>
<td>AREA REPORT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AREA STATISTICS SECTION</td>
</tr>
</tbody>
</table>

1. AREA NAME: STUDENT-REGION
2. FILE CREATION DATA: mm/dd/yy - hh:mm:ss

<table>
<thead>
<tr>
<th>ITEM</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>PAGE SIZE - IN BYTES</td>
</tr>
<tr>
<td>4</td>
<td>AREA - LOW PAGE</td>
</tr>
<tr>
<td>5</td>
<td>AREA - HIGH PAGE</td>
</tr>
<tr>
<td>6</td>
<td>AREA - TOTAL PAGES</td>
</tr>
<tr>
<td>7</td>
<td>AREA - TOTAL SPACE</td>
</tr>
<tr>
<td>8</td>
<td>SWEEP - LOW PAGE</td>
</tr>
<tr>
<td>9</td>
<td>SWEEP - HIGH PAGE</td>
</tr>
<tr>
<td>10</td>
<td>SWEEP - TOTAL PAGES</td>
</tr>
<tr>
<td>11</td>
<td>SWEEP - TOTAL SPACE</td>
</tr>
<tr>
<td>12</td>
<td>BYTES USED - POINTERS</td>
</tr>
<tr>
<td>13</td>
<td>BYTES USED - DATA</td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>14</td>
<td>Bytes Used - Line Indices</td>
</tr>
<tr>
<td>15</td>
<td>Bytes Used - Headers/Footers</td>
</tr>
<tr>
<td>16</td>
<td>Bytes Used - SR7 Records</td>
</tr>
<tr>
<td>17</td>
<td>Bytes Used - SR8 Records</td>
</tr>
<tr>
<td>18</td>
<td>Bytes Used - SMPS</td>
</tr>
<tr>
<td>19</td>
<td>Bytes Used - Total</td>
</tr>
<tr>
<td></td>
<td>Free Space</td>
</tr>
<tr>
<td></td>
<td>Avg/Page</td>
</tr>
<tr>
<td></td>
<td>Max/Page</td>
</tr>
<tr>
<td></td>
<td>Min/Page</td>
</tr>
<tr>
<td>20-21</td>
<td>Free Space Less Page Reserve</td>
</tr>
<tr>
<td></td>
<td>Avg/Page</td>
</tr>
<tr>
<td></td>
<td>Max/Page</td>
</tr>
<tr>
<td></td>
<td>Min/Page</td>
</tr>
<tr>
<td>22</td>
<td>Pages Physically Full</td>
</tr>
<tr>
<td>23</td>
<td>Pages Logically Full</td>
</tr>
<tr>
<td>24</td>
<td>Pages Empty</td>
</tr>
<tr>
<td>25</td>
<td>Space Management Pages</td>
</tr>
</tbody>
</table>

**CA-TOOLS**

**RELEASE**

**CA IDMS/DB ANALYZER**

**AREA REPORT**

**AREA STATISTICS SECTION**

**AREA NAME:** STUDENT-REGION

**FILE CREATION DATA:** mm/dd/yy - hh:mm:ss

**ITEM** | **TOTAL** | **180**
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>Number of Record Occurrences</td>
</tr>
<tr>
<td>27</td>
<td>Recs Out of Physical Sequence</td>
</tr>
<tr>
<td>28</td>
<td>LDEL Records ...............</td>
</tr>
<tr>
<td>29</td>
<td>Bytes To Store</td>
</tr>
<tr>
<td>30</td>
<td>Relocated Records ..........</td>
</tr>
<tr>
<td>31</td>
<td>Bytes To Store</td>
</tr>
<tr>
<td>32</td>
<td>Bytes That Could Be Returned</td>
</tr>
<tr>
<td>33</td>
<td>Fragmented Records</td>
</tr>
<tr>
<td>34</td>
<td>Bytes To Store</td>
</tr>
<tr>
<td>35</td>
<td>Bytes That Could Be Returned</td>
</tr>
<tr>
<td>36</td>
<td>Fragments Per Occurrence</td>
</tr>
<tr>
<td>37</td>
<td>Total Fragments in Area</td>
</tr>
<tr>
<td>38</td>
<td>Compressed Data Records ......</td>
</tr>
<tr>
<td>39</td>
<td>Savings Due to Compression</td>
</tr>
</tbody>
</table>
16-Jan-2018

### CA IDMS - 19.0

#### AVG | 371.00
#### MAX | 408
#### MIN | 0

<table>
<thead>
<tr>
<th>40</th>
<th>SR8 RECORDS</th>
<th>51</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>DECOMPRESSED BYTES TO STORE</td>
<td>2,228</td>
</tr>
<tr>
<td>42</td>
<td>COMPRESSED BYTES TO STORE</td>
<td>2,228</td>
</tr>
<tr>
<td>43</td>
<td>SAVINGS DUE TO COMPRESSION</td>
<td>0</td>
</tr>
</tbody>
</table>

---

**Note:** In a case where only a portion of an area has been swept for statistical accumulation, the field values described pertain only to the portion of the area swept. Exceptions are items 3, 4, 5, 6, and 7, which pertain to the entire area regardless of whether it was actually swept.

---

1 **AREA NAME**--The name of the area for the reported statistics.

2 **FILE CREATION DATA**--The date and time that the statistical file used for this report was created. This information is important for identifying which statistical file is being used since there can be more than one file in storage.

3 **PAGE SIZE - IN BYTES**--The number of bytes allocated to each page in the area.

4 **AREA - LOW PAGE**--The page number of the beginning page for the entire area.

5 **AREA - HIGH PAGE**--The page number of the ending page for the entire area.

6 **AREA - TOTAL PAGES**--The number of pages in the entire area.

7 **AREA - TOTAL SPACE**--The total number of bytes allocated to the entire area. It is equivalent to the number of pages in the area (item 6) multiplied by the page size (item 3).

8 **Sweep - Low Page**--The lowest page number when a portion of an area was swept. If the entire area was swept, this page number is identical to the AREA - LOW PAGE number (item 4).

9 **Sweep - High Page**--The highest page number when a portion of an area was swept. If the entire area was swept, this page number is identical to the AREA - HIGH PAGE number (item 5).

10 **Sweep - Total Pages**--The total number of pages swept in the area.

11 **Sweep - Total Space**--The total number of bytes allocated to the swept portion of the area and the percentage of space allocated to the swept portion of the area. The total equals the sum of BYTES USED - TOTAL (item 19) and BYTES OF FREE SPACE (item 20).

\[
PCT=100 \times \frac{\text{Sweep, Total Space} (11)}{\text{Area, Total Space} (7)}
\]

If the entire area was swept, the first value is identical to that shown for AREA - TOTAL SPACE (item 7) and the percentage is 100.

12 **Bytes Used - Pointers**--The number of bytes used by pointers in the swept area and the percentage of space used by pointers.

\[
PCT=100 \times \frac{\text{Bytes Used, Pointers} (12)}{\text{Sweep, Total Space} (11)}
\]
13 BYTES USED - DATA--The compressed number of bytes used by data (record occurrences) and the percentage of space used by data.

\[ PCT = 100 \times \frac{BYTES\ USED\ ,\ DATA}{SWEEP\ ,\ TOTAL\ SPACE} \]

14 BYTES USED - LINE INDICES--The number of bytes used by line indexes and the percentage of space used by line indexes.

\[ PCT = 100 \times \frac{BYTES\ USED\ ,\ LINE\ INDICES}{SWEEP\ ,\ TOTAL\ SPACE} \]

15 BYTES USED - HEADERS/FOOTERS--The number of bytes used by page headers plus the number of bytes used by page footers and the percentage of space used by both page headers and footers. Since each header uses 16 bytes and each footer uses 16 bytes, and there is one header and one footer per page, the first value is equivalent to 32 multiplied by the total number of pages swept in the area (item 10).

\[ PCT = 100 \times \frac{BYTES\ USED\ ,\ HEADERS/FOOTERS}{SWEEP\ ,\ TOTAL\ SPACE} \]

16 BYTES USED - SR7 RECORDS--The number of bytes used by SR7 records and the percentage of space used by SR7 records.

17 BYTES USED - SR8 RECORDS--The number of bytes used by SR8 records and the percentage of space used by SR8 records.

18 BYTES USED - SMPS--The number of bytes used by space management pages minus the number of bytes used for headers and footers for those pages and the percentage of space used by space management pages.

\[ PCT = 100 \times \frac{BYTES\ USED\ ,\ SMPS}{SWEEP\ ,\ TOTAL\ SPACE} \]

19 BYTES USED - TOTAL--The total number of bytes used in the area and the percentage of space used for this storage. The total number of bytes used is the sum of items 12, 13, 14, 15, 16, 17, and 18.

\[ PCT = 100 \times \frac{BYTES\ USED\ ,\ TOTAL}{SWEEP\ ,\ TOTAL\ SPACE} \]

20 BYTES OF FREE SPACE--The actual free space found in the area and the percentage of actual free space found in the area. Bytes of free space is the sum of the space available values located in the header record of each page swept in the area.

\[ PCT = 100 \times \frac{FREE\ SPACE}{SWEEP\ ,\ TOTAL\ SPACE} \]

Also exhibited are the average number of bytes of free space on a page (AVG/PAGE), the maximum number of bytes of free space on a page that is not empty (MAX/PAGE), and the minimum number of bytes of free space on a page that is not full (MIN/PAGE).

\[ AVG/PAGE = \frac{BYTES\ FREE\ SPACE}{TOTAL\ PAGES} - SMPS \]

\[ MAX/PAGE = \frac{BYTES\ FREE\ SPACE}{TOTAL\ PAGES} - SMPS \]

\[ MIN/PAGE = \frac{BYTES\ FREE\ SPACE}{TOTAL\ PAGES} - SMPS \]

Note: The AVG/PAGE may exceed the MAX/PAGE if the area has a sufficient number of empty pages.
Bytes of free space values indicate area density. Poor performance for some database update functions may occur when area density is 80% or greater.

**21 BYTES OF FREE SPACE LESS PAGE RESERVE** --The number of bytes available for data storage and the percentage of bytes available for data storage. The values represent the actual free space found in the area minus the page reserve.

\[
PCT=100 \quad * \quad \text{FREE SPACE-PAGE RESERVE (21)} \quad \text{SWEEP, TOTAL SPACE (11)}
\]

**22 PAGES PHYSICALLY FULL**--The number of pages whose header record space available values are zero and the percentage of pages whose space available values are zero.

\[
PCT=100 \quad * \quad \text{PAGES PHYSICALLY FULL (22)} \quad \text{TOTAL PAGES (10) - SMPS (23)}
\]

When pages become physically full, records will overflow to pages with available space. Performance degradation due to overflow records is likely to become significant as the percentage of physically full pages increases beyond 50%.

**23 PAGES LOGICALLY FULL**--The number of pages containing the maximum data record occurrences and the percentage of pages containing the schema defined maximum record occurrences.

\[
PCT=100 \quad * \quad \text{PAGES LOGICALLY FULL (23)} \quad \text{TOTAL PAGES (10) - SMPS (25)}
\]

A large number of logically full pages indicates that the page size is too large for the record types being stored in the area.

**24 PAGES EMPTY**--The number of pages that contain no record occurrences and the percentage of pages that contain no record occurrences.

\[
PCT=100 \quad * \quad \text{PAGES EMPTY (24)} \quad \text{TOTAL PAGES (10) - SMPS (25)}
\]

**25 SPACE MANAGEMENT PAGES**--The number of space management pages and the percentage of space management pages.

\[
PCT=100 \quad * \quad \text{SPACE MANAGEMENT PAGES (25)} \quad \text{TOTAL PAGES (10)}
\]

A space management page is a CA IDMS generated administrative page that contains records showing the number of bytes available for each page in a series of pages in an area.

**26 NUMBER OF RECORD OCCURRENCES**--The total number of record occurrences in the area swept, including logically deleted (LDEL) records.

**27 RECORDS OUT OF PHYSICAL SEQUENCE**--As of Release 12.0, this value is always zero.

**28 LDEL RECORDS IN AREA**--The number of logically deleted record occurrences in the area swept and the percentage of logically deleted record occurrences in the area swept.

\[
PCT=100 \quad * \quad \text{LDEL RECORDS IN AREA (28)} \quad \text{RECORD OCCURRENCES (26)}
\]

LDEL records cause unnecessary update and retrieval overhead. This overhead increases with the number of LDEL records.

**29 BYTES TO STORE (LDEL RECORDS)**--The number of bytes used to store all logically deleted record occurrences and the percentage of bytes used for LDEL records.
30 RELOCATED RECORDS IN AREA--The number of relocated record occurrences in the area swept and the percentage of relocated records.

Relocated records cause performance overhead. If the percentage of relocated records is greater than 20% and the percentage of bytes that could be returned (see item 32) exceeds 50%, consider reorganizing the database. If the percentage of relocated records is greater than 20% but the percentage of bytes that could be returned is less than 50%, consider expanding the database.

31 BYTES TO STORE (RELOCATED RECORDS)--The number of bytes used to store all relocated record occurrences (and associated pointers) and the percentage of bytes used for relocated records (SR2 and SR3 records).

32 BYTES THAT COULD BE RETURNED (RELOCATED RECORDS)--The number of bytes of relocated record storage that could be returned to the target page were a reorganization to take place and the percentage of bytes that could be returned.

33 FRAGMENTED RECORDS--The number of fragmented record occurrences and the percentage of fragmented record occurrences.

Record fragments cause performance overhead. If the percentage of fragments is greater than 20% and the percentage of bytes that could be returned (see item 35) exceeds 50%, consider reorganizing the database. If the percentage of record fragments is greater than 20% but the percentage of bytes that could be returned is less than 50%, consider expanding the database.

34 BYTES TO STORE (FRAGMENTED RECORDS)--The number of bytes used to store all record fragments and the percentage of bytes used to store all record fragments.

35 BYTES THAT COULD BE RETURNED (FRAGMENTED RECORDS)--The number of bytes of fragmented records that could be returned to the target page, were a reorganization to take place, and the percentage of bytes that could be returned.

36 FRAGMENTS PER OCCURRENCE--The number of fragments for each fragmented record occurrence expressed as the average number of fragments per occurrence (AVG), the greatest number of fragments into which a record occurrence is divided (MAX), and the least number of fragments into which a record occurrence is divided (MIN).

37 TOTAL FRAGMENTS IN AREA--The total number of record fragments in the area swept.
38 NUMBER OF RECORDS COMPRESSED--The number of compressed record occurrences and the percentage of compressed record occurrences. This does not include SR8 record compression.

\[ PCT = \frac{\text{RECORDS COMPRESSED}}{\text{RECORD OCCURRENCES}} \times 100 \]

This statistic is valid only if IDMSCOMP was used as the compression routine.

39 SAVINGS DUE TO COMPRESSION--The number of bytes saved due to the compression of record occurrences (decompressed bytes minus compressed bytes) and the percentage of savings due to compression. The compression of SR8 records is not included in this statistic.

\[ PCT = \frac{\text{SAVINGS DUE COMPRESSION}}{\text{TOTAL SPACE}} \times 100 \]

Also included are the average number of bytes of savings per compressed record (AVG), the greatest number of bytes of savings in a compressed record (MAX), and the least number of bytes of savings in a compressed record (MIN).

\[ \text{AVG} = \frac{\text{SAVINGS DUE COMPRESSION}}{\text{RECORDS COMPRESSED}} \]

This statistic is valid only if IDMSCOMP was used as the compression routine.

40 SR8 RECORDS--The total number of SR8 records located in this area, and the percentage of the records in the area that are SR8 records.

\[ PCT = \frac{\text{SR8 RECORDS}}{\text{RECORD OCCURRENCES}} \times 100 \]

41 DECOMPRESSED BYTES TO STORE--The maximum number of bytes required to store the SR8 records for this area when the automatic CA IDMS compression feature and the user-specified compression option are not used. You can expect to gain storage savings through compression when you specify SYMBOLIC KEY for the sort type.

42 COMPRESSED BYTES TO STORE--The number of bytes actually required to store the SR8 records for this area.

43 SAVINGS DUE TO COMPRESSION--The number of bytes saved through use of the CA IDMS automatic compression feature and/or the user-specified compression option.

\[ PCT = \frac{\text{SAVINGS DUE TO COMPRESSION}}{\text{COMPRESSED BYTES TO STORE}} \times 100 \]

Distribution Section Fields

The Distribution Section of the Area Report is shown in the following display. This section of the report displays the distribution of the following selected area features.
<table>
<thead>
<tr>
<th></th>
<th>PERCENT OF PAGE FULL</th>
<th>PERCENT RANGE</th>
<th>PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 - 10</td>
<td>11 - 20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>21 - 30</td>
<td>31 - 40</td>
<td></td>
</tr>
<tr>
<td></td>
<td>41 - 50</td>
<td>61 - 70</td>
<td></td>
</tr>
<tr>
<td></td>
<td>81 - 90</td>
<td>91 - 100</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>RECORD OCCURRENCES</th>
<th>PAGES</th>
<th>OCCURRENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>91001 - 91008</td>
<td>61</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>91009 - 91016</td>
<td>71</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>91017 - 91024</td>
<td>24</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>91025 - 91032</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>91033 - 91040</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>91041 - 91048</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>91049 - 91056</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>91057 - 91064</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>91065 - 91072</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>91073 - 91080</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>FRAGMENTS</th>
<th>PAGES</th>
<th>FRAGMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>91001 - 91008</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>91009 - 91016</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>91017 - 91024</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>91025 - 91032</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>91033 - 91040</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>91041 - 91048</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>91049 - 91056</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>91057 - 91064</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>91065 - 91072</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>91073 - 91080</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>FREE SPACE</th>
<th>PAGES</th>
<th>BYTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>91001 - 91008</td>
<td>16.880</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>91009 - 91016</td>
<td>18.944</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>RANGE</td>
<td>TOTAL</td>
<td>PCT</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>-----</td>
<td></td>
</tr>
<tr>
<td>0 - 0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1 - 50</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>51 - 100</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>101 - 150</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>151 - 200</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>201 - 250</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>251 - 300</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>301 - 350</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>351 - 400</td>
<td>28</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>401 - 450</td>
<td>23</td>
<td>11</td>
<td></td>
</tr>
</tbody>
</table>

Note: In a case where only a portion of an area has been swept for statistical accumulation, the field values described pertain only to the portion of the area swept.

1 PERCENT OF PAGE FULL--The distribution of pages by range of percent physically full is shown here. Ranges of percent full are listed under the column headed RANGE. The number of pages found in each range of percent full is listed under the column headed TOTAL. The percentage of pages found in each page range is listed under the column headed PCT.

\[
PCT = \frac{\text{NUMBER OF PAGES IN RANGE}}{\text{TOTAL PAGES} \times 100}
\]

The distribution of pages by percent of page full is based upon data taken from the SR1 record for each page in the area swept.

2 RECORD OCCURRENCES--The distribution of record occurrences by page range is shown here. Page ranges are listed under the column headed RANGE. The number of occurrences found in each page range is listed in the column headed TOTAL. The percentage of occurrences found in each page range is listed in the column headed PCT.

\[
PCT = \frac{\text{OCCURRENCES IN PAGE RANGE}}{\text{RECORD OCCURRENCES} \times 100}
\]
3 FRAGMENTS--The distribution of record fragments by page range is shown here. Page ranges are listed in the column headed RANGE. The total number of fragments found in each page range is listed in the column headed TOTAL. The percentage of fragments found in each page range is listed in the column headed PCT.

\[ \text{PCT} = 100 \times \frac{\text{FRAGMENTS, AREA}}{\text{AREA Stats 37}} \]

4 FREE SPACE--The distribution of free space by page range is shown here. Page ranges are listed in the column headed RANGE. The amount of free space (in numbers of bytes) found in each page range is listed in the column headed TOTAL. The percentage of free space found in each page range is listed in the column headed PCT.

\[ \text{PCT} = 100 \times \frac{\text{FREE SPACE IN PAGE RANGE}}{\text{AREA Stats 20}} \]

5 SAVINGS DUE TO COMPRESSION--The distribution of compressed record occurrences by range of bytes saved is shown here. Ranges of bytes saved are listed in the column headed RANGE. The number of compressed record occurrences found in each range is listed in the column headed TOTAL. The percentage of compressed record occurrences found in each range is listed in the column headed PCT.

\[ \text{PCT} = 100 \times \frac{\text{RECS COMPRESSED, AREA}}{\text{AREA Stats 38}} \]

This statistic is valid only if IDMSCOMP was used as the compression routine.

The Histogram Section of the Area Report is shown in the following display. This section of the report is a graphic representation and a quick reference version of the Distribution Section. Distribution Section percentages are shown for each range in the form of a bar chart.
### Record Occurrences - Percentage by Distribution Range

<table>
<thead>
<tr>
<th>Pages</th>
<th>0-9</th>
<th>10-19</th>
<th>20-29</th>
<th>30-39</th>
<th>40-49</th>
<th>50-59</th>
<th>60-69</th>
<th>70-79</th>
<th>80-89</th>
<th>90-99</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>91001 - 91008</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>91009 - 91016</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>91017 - 91024</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>91025 - 91032</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>91033 - 91040</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>91041 - 91048</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>91049 - 91056</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>91057 - 91064</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>91065 - 91072</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>91073 - 91080</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

### Fragments - Percentage by Distribution Range

<table>
<thead>
<tr>
<th>Pages</th>
<th>0-9</th>
<th>10-19</th>
<th>20-29</th>
<th>30-39</th>
<th>40-49</th>
<th>50-59</th>
<th>60-69</th>
<th>70-79</th>
<th>80-89</th>
<th>90-99</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>91001 - 91008</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>91009 - 91016</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>91017 - 91024</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>91025 - 91032</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>91033 - 91040</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>91041 - 91048</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>91049 - 91056</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>91057 - 91064</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>91065 - 91072</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>91073 - 91080</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

### Free Space - Percentage by Distribution Range

<table>
<thead>
<tr>
<th>Pages</th>
<th>0-9</th>
<th>10-19</th>
<th>20-29</th>
<th>30-39</th>
<th>40-49</th>
<th>50-59</th>
<th>60-69</th>
<th>70-79</th>
<th>80-89</th>
<th>90-99</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>91001 - 91008</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>91009 - 91016</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>91017 - 91024</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>91025 - 91032</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>91033 - 91040</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>91041 - 91048</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>91049 - 91056</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>
Analysis by Record Section Fields

The Analysis by Record Section of the Area Report is shown in the following display. This section of the report provides summary statistics by record type for both active and logically deleted (LDEL) records. A description of each field is given below.
Note: In a case where only a portion of an area has been swept for statistical accumulation, the field values described pertain only to the portion of the area swept.

1 RECORD NAME--Listed under this heading is the record name for each record type in the area swept.

2 LOCATION MODE--The location mode under which the record is stored. The value will be either V (VIA), C(CALC), or D(DIRECT).

3 OCCURRENCES--The values listed under this heading are the total number of active record occurrences and the percentage of active record occurrences for each record type.

4 SPACE USED--The values listed under this heading are the total number of bytes used by active record occurrences and the percentage of bytes used by active record occurrences for each record type.

5 LENGTH--The values listed under this heading are the average, maximum, and minimum lengths (number of bytes used) of active record occurrences for each record type. Fixed length record types show identical average, maximum, and minimum values.

6 FRAGMENTED RECORDS--The values listed under this heading are the total number of active fragmented record occurrences and the percentage of active fragmented record occurrences for each record type.

7 COMPRESSION BYTES SAVED--The values listed under this heading are the total number of bytes saved by active compressed record occurrences for each record type.

8 COUNT--The values listed under this heading are the total number of logically deleted record occurrences for each record type.

9 SPACE USED--The values listed under this heading are the total number of bytes used by logically deleted records and the percentage of space used by logically deleted records for each record type.
Analysis by Record Section Fields (see page 163)

A Comparative Area Report is used to assess changes that have occurred in the physical organization of an area, or portion of an area. Consequently, this type of report uses data from two different statistical files. The data from these files describe the area at two different points in time.

Similar to the Area Report, the Comparative Area Report is divided into four sections--the Area Statistics Section, the Distribution Section, the Histogram Section, and the Analysis by Record Section.

The format of each section in this report is the same as that of an Area Report except that fields have been added to show “old” and “new” statistical file values and, when possible, the amount of change between these values.

Comparative Area Report Fields are shown in the following screen displays. Fields at the top of the Comparative Area Report are described below. See Area Report Field Descriptions (see page 145).

### Comparative Area Report Field Descriptions

The Report Identification fields and the Area Statistics Section fields are described below and are shown in the Comparative Area Report.

<table>
<thead>
<tr>
<th>CA-TOOLS</th>
<th>RELEASE</th>
<th>CA IDMS/DB ANALYZER</th>
<th>AREA REPORT</th>
<th>AREA STATISTICS SECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA-TOOLS</td>
<td>Rnn.nn</td>
<td>AREA REPORT</td>
<td>dd/mm/yy</td>
<td>mm/dd/yy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AREA NAME: STUDENT-REGION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEW FILE CREATION DATA: mm/dd/yy - hh:mm:ss</td>
</tr>
<tr>
<td>OLD FILE CREATION DATA: mm/dd/yy - hh:mm:ss</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ITEM</th>
<th>NEW</th>
<th>PCT</th>
<th>OLD</th>
<th>PCT</th>
<th>VARIANCE</th>
<th>PCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAGE SIZE - IN BYTES</td>
<td>3,156</td>
<td>1</td>
<td>3,156</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AREA - LOW PAGE</td>
<td>91,001</td>
<td>1</td>
<td>91,001</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AREA - HIGH PAGE</td>
<td>91,000</td>
<td>1</td>
<td>91,000</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AREA TOTAL PAGES</td>
<td>89</td>
<td>1</td>
<td>89</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AREA TOTAL SPACE</td>
<td>252,480</td>
<td>100</td>
<td>252,480</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SWEEP - LOW PAGE</td>
<td>92,001</td>
<td>1</td>
<td>91,001</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SWEEP - HIGH PAGE</td>
<td>91,000</td>
<td>1</td>
<td>91,000</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SWEEP TOTAL PAGES</td>
<td>89</td>
<td>1</td>
<td>89</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SWEEP TOTAL SPACE</td>
<td>52,480</td>
<td>100</td>
<td>252,480</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BYTES USED - POINTERS</td>
<td>1,392</td>
<td>1</td>
<td>3,908</td>
<td>2</td>
<td>-2,516</td>
<td>1</td>
</tr>
<tr>
<td>BYTES USED - DATA</td>
<td>3,900</td>
<td>2</td>
<td>5,420</td>
<td>2</td>
<td>-1,520</td>
<td>0</td>
</tr>
<tr>
<td>BYTES USED - LINE INDICES</td>
<td>896</td>
<td>1</td>
<td>1,664</td>
<td>1</td>
<td>-768</td>
<td>1</td>
</tr>
<tr>
<td>BYTES USED - HEADERS/FOOTERS</td>
<td>2,560</td>
<td>1</td>
<td>2,560</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BYTES USED - SR7 RECORDS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BYTES USED - SR8 RECORDS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BYTES USED - SMPS</td>
<td>3,124</td>
<td>1</td>
<td>3,124</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BYTES USED - TOTAL</td>
<td>52,480</td>
<td>100</td>
<td>252,480</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FREE SPACE PLUS PAGE RESERVE</td>
<td>240,608</td>
<td>95</td>
<td>233,576</td>
<td>93</td>
<td>+7,032</td>
<td>2</td>
</tr>
<tr>
<td>AVG/PAGE</td>
<td>3,045.67</td>
<td>2,956.66</td>
<td>+89.01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX/PAGE</td>
<td>3,056</td>
<td>3,056</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIN/PAGE</td>
<td>1,984</td>
<td>800</td>
<td>+1,184</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FREE SPACE LESS PAGE RESERVE</td>
<td>240,608</td>
<td>95</td>
<td>233,576</td>
<td>93</td>
<td>+7,032</td>
<td>2</td>
</tr>
<tr>
<td>AVG/PAGE</td>
<td>3,045.67</td>
<td>2,956.66</td>
<td>+89.01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX/PAGE</td>
<td>3,056</td>
<td>3,056</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIN/PAGE</td>
<td>1,984</td>
<td>800</td>
<td>+1,184</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAGES PHYSICALLY FULL</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PAGES LOGICALLY FULL</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PAGES EMPTY</td>
<td>66</td>
<td>84</td>
<td>61</td>
<td>77</td>
<td>+5</td>
<td>7</td>
</tr>
<tr>
<td>SPACE MANAGEMENT PAGES</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>NUMBER OF RECORD OCCURRENCES</td>
<td>112</td>
<td>208</td>
<td>-96</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RECS OUT OF PHYSICAL SEQUENCE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LDEL RECORDS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BYTES TO STORE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BYTES THAT COULD BE RETURNED</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
**Distribution Section Fields**

The Distribution Section of the Comparative Area Report is shown in the following display. See Area Report Field Descriptions (see page 145) for descriptions of fields not discussed here.
<table>
<thead>
<tr>
<th>RECORD OCCURRENCES</th>
<th>PAGES</th>
<th>OCCURRENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>91001 - 91008</td>
<td>42</td>
<td>61</td>
</tr>
<tr>
<td>91009 - 91016</td>
<td>55</td>
<td>71</td>
</tr>
<tr>
<td>91017 - 91024</td>
<td>14</td>
<td>24</td>
</tr>
<tr>
<td>91025 - 91032</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>91033 - 91040</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>91041 - 91048</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>91049 - 91056</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>91057 - 91064</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>91065 - 91072</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>91073 - 91080</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FRAGMENTS</th>
<th>PAGES</th>
<th>FRAGMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>91001 - 91008</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>91009 - 91016</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>91017 - 91024</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>91025 - 91032</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>91033 - 91040</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>91041 - 91048</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>91049 - 91056</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>91057 - 91064</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>91065 - 91072</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>91073 - 91080</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FREE SPACE</th>
<th>PAGES</th>
<th>BYTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>91001 - 91008</td>
<td>19,556</td>
<td>8,166,880</td>
</tr>
<tr>
<td>91009 - 91016</td>
<td>21,960</td>
<td>9,18,944</td>
</tr>
<tr>
<td>91017 - 91024</td>
<td>24,216</td>
<td>10,22,876</td>
</tr>
<tr>
<td>91025 - 91032</td>
<td>24,992</td>
<td>10,24,992</td>
</tr>
<tr>
<td>91033 - 91040</td>
<td>24,992</td>
<td>10,24,992</td>
</tr>
<tr>
<td>91041 - 91048</td>
<td>24,924</td>
<td>10,24,924</td>
</tr>
</tbody>
</table>
1 NEW PCT--The values listed under this heading are raw counts and percentages for each range. These values are taken from the new (STAT1) statistics file. See Comparative Area Report Identification Fields, item 2.

2 OLD PCT--The values listed under this heading are raw counts and percentages for each range. These values are taken from the old (STAT2) statistics file. See Comparative Area Report Identification Fields, item 3.

3 VARIANCE PCT--Listed under this heading are the differences between new and old raw counts and the percent variance between the old and new values. Positive values indicate the amount by which new statistical file values are greater than old. Negative values indicate the amount by which new file values are less than old.

For variance to be meaningful, both the old and new statistical files must contain identical ranges. If they do not, no values are listed under this heading.

Histogram Section Fields

A portion of the Histogram Section of the Comparative Area Report is shown in the following display. This section of the report is a graphic representation and a quick reference version of the Distribution Section. Old (O) and new (N) values of Distribution Section percentages for each range are shown on two lines in the form of a bar chart.

<table>
<thead>
<tr>
<th>SAVINGS DUE TO COMPRESSION</th>
<th>BYTES SAVED</th>
<th>OCCURRENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 - 50</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>51 - 100</td>
<td>45</td>
<td>40</td>
</tr>
<tr>
<td>101 - 150</td>
<td>48</td>
<td>43</td>
</tr>
<tr>
<td>151 - 200</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>201 - 250</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>251 - 300</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>301 - 350</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>351 - 400</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>401 - 450</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

16-Jan-2018 161/898
<table>
<thead>
<tr>
<th>PERCENT RANGE</th>
<th>PERCENT OF PAGE FULL</th>
<th>PERCENTAGE BY DISTRIBUTION RANGE</th>
</tr>
</thead>
</table>
| 0 - 10        | NNAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Analysis by Record Section Fields

The Analysis by Record Section of the Comparative Area Report is shown in the following display. This section of the report is identical to its Area Report counterpart except that it lists statistics from both the old and new statistical files. See Area Report Field Descriptions (see page 145) for a description of fields in this section.
Record Report

Contents

- Record Report Field Descriptions (see page 164)
- Distribution Section Fields (see page 168)
- Histogram Section Fields (see page 170)

A Record Report can be produced for every record type residing in an area, or portion of an area, for which statistics have been accumulated. Reports for record types residing in an area, or portion of an area, excluded from statistical accumulation by an AREA statement cannot be produced.

A Record Report is divided into three sections: the Record Statistics Section, the Distribution Section, and the Histogram Section.

Record Report Field Descriptions

Record Report fields are shown in the following screen displays. Following are descriptions of two report identification fields plus fields used in the Record Statistics and Distribution Sections of the Record Report.
**RECORD STATISTICS SECTION**

<table>
<thead>
<tr>
<th>ITEM</th>
<th>TOTAL</th>
<th>PCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. RECORD ID/NAME:</td>
<td>1002 - SUBJECT</td>
<td></td>
</tr>
<tr>
<td>2. FILE CREATION DATA:</td>
<td>mm/dd/yy - hh:mm:ss</td>
<td></td>
</tr>
<tr>
<td>3. RECORD LOCATION MODE</td>
<td>CALC</td>
<td></td>
</tr>
<tr>
<td>4. NUMBER OF OCCURRENCES</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>5. BYTES TO STORE ALL OCCURRENCES</td>
<td>2,912</td>
<td></td>
</tr>
<tr>
<td>6. RECS OUT OF PHYSICAL SEQUENCE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>7. NUMBER OF LDEL OCCURRENCES</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>8. BYTES TO STORE LDEL RECORDS</td>
<td>1,400</td>
<td></td>
</tr>
<tr>
<td>9. OCCURRENCE LENGTH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>COMPRESSED</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AVG</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>MAX</td>
<td>188</td>
</tr>
<tr>
<td></td>
<td>MIN</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>DECOMPRESSED</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AVG</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>MAX</td>
<td>188</td>
</tr>
<tr>
<td></td>
<td>MIN</td>
<td>56</td>
</tr>
<tr>
<td>10. RELOCATED OCCURRENCES</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11. BYTES TO STORE RELOCATED RECS</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>12. BYTES THAT COULD BE RETURNED</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>13. OCCURRENCES WITH FRAGMENTS</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>14. FRAGMENTS PER OCCURRENCE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AVG</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>MAX</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>MIN</td>
<td>0</td>
</tr>
<tr>
<td>15. NUMBER OF FRAGMENTS</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>16. BYTES TO STORE ALL FRAGMENTS</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>17. BYTES THAT COULD BE RETURNED</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>18. NUMBER OF RECORDS COMPRESSED</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>19. SAVINGS DUE TO COMPRESSION</td>
<td>0.00</td>
<td>0</td>
</tr>
</tbody>
</table>

**Note:** In a case where only a portion of an area has been swept for statistical accumulation, the field values described pertain only to the portion of the area swept. Remember that statistics shown on the Record Report apply only to the record type indicated in the RECORD ID/NAME field of the report (item 1).

1. **RECORD ID/NAME**--The identification number and name of the record type for the reported statistics.

2. **FILE CREATION DATA**--The date and time that the statistical file used for this report was created. This information helps identify which statistical file is being used since there can be more than one file in storage.

3. **RECORD LOCATION MODE**--The location mode for the record type as defined in the schema is shown here.

4. **NUMBER OF OCCURRENCES**--The total number of record occurrences. This value includes logically deleted (LDEL) records.
5 BYTES TO STORE ALL OCCURRENCES--The total number of bytes used to store all record occurrences.

6 RECORDS OUT OF PHYSICAL SEQUENCE--As of Release 12.0, this value is always zero.

   PCT=100 *   RECS OUT OF PHYS SEQUENCE(6) NUMBER OF OCCURRENCES (4)

7 NUMBER OF LDEL OCCURRENCES--The number of logically deleted record occurrences in the area swept for this record type and the percentage of total records of this type that are logically deleted.

   PCT=100 *   NUMBER OF LDEL OCCURRENCES(7) NUMBER OF OCCURRENCES (4)

8 BYTES TO STORE LDEL RECORDS--The number of bytes used to store all logically deleted record occurrences and the percentage of total bytes used for this record type that are for logically deleted occurrences.

   PCT=100 *   BYTES TO STORE LDEL RECS(8) BYTES TO STORE ALL OCCURRENCES (5)

9 OCCURRENCE LENGTH (if fixed)--If occurrences for the record type are of fixed length and are not compressed, the number of bytes used for occurrence length is shown here.

   If the record type is of variable length or is compressed, the fixed occurrence length is not shown on the report. For variable length, see next item.

9 OCCURRENCE LENGTH (if variable)--If occurrences for the record type are of variable length or are compressed, the average, maximum, and minimum number of bytes used for both compressed and decompressed occurrence lengths is shown here. The decompressed occurrence length is only valid if IDMSCOMP was used as the compression routine.

   If the record type is of fixed length and is not compressed, these fields are not shown on the report.

10 RELOCATED OCCURRENCES--The total number of relocated record occurrences and the percentage of relocated record occurrences.

   PCT=100 *   RELOCATED OCCURRENCES (10) NUMBER OF OCCURRENCES (4)

11 BYTES TO STORE ALL RELOCATED RECS--The total number of bytes used to store all relocated record occurrences (and associated pointers) and the percentage of bytes used for relocated record occurrences.

   PCT=100 *   BYTES, STORE ALL RELOC RECS(11) BYTES TO STORE ALL RELOCATED RECS (5)

12 BYTES THAT COULD BE RETURNED(RELOCATED RECORDS)--The total number of bytes of relocated records that could be returned to the target page, were a reorganization to take place, and the percentage of bytes that could be returned.

   PCT=100 *   BYTES COULD BE RET,RELOC RECS (12) BYTES TO STORE ALL RELOC RECS (11)

13 OCCURRENCES WITH FRAGMENTS--The total number of fragmented record occurrences and the percentage of fragmented record occurrences.

   PCT=100 *   OCCURRENCES WITH FRAGMENTS(13) NUMBER OF OCCURRENCES (4)
If the record is a fixed length record, this field is not shown on the report since fixed length records cannot be fragmented.

14 FRAGMENTS PER OCCURRENCE--The number of fragments for each fragmented record occurrence expressed as the average number of fragments per occurrence (AVG), the greatest number of fragments into which a record occurrence is divided (MAX), and the least number of fragments into which a record occurrence is divided (MIN).

\[
AVG = \frac{\text{NUMBER OF FRAGMENTS (15)}}{\text{NUMBER OF OCCURRENCES (4)}}
\]

If the record is a fixed length record, this field is not shown on the report since fixed length records cannot be fragmented.

15 NUMBER OF FRAGMENTS--The total number of record fragments. If the record is a fixed length record, this field is not shown on the report since fixed length records cannot be fragmented.

16 BYTES TO STORE ALL FRAGMENTS--The number of bytes used to store all fragments and the percentage of bytes used to store all fragments.

\[
PCT = 100 \times \frac{\text{BYTES TO STORE ALL FRAGMENTS (16)}}{\text{BYTES TO STORE ALL OCS (5)}}
\]

If the record is a fixed length record, this field is not shown on the report since fixed length records cannot be fragmented.

17 BYTES THAT COULD BE RETURNED (FRAGMENTED RECORDS)--The total number of bytes of fragmented records that could be returned to the target page were a reorganization to take place and the percentage of bytes that could be returned.

\[
PCT = 100 \times \frac{\text{BYTES COULD BE RET, FRAGRECS (17)}}{\text{BYTES TO STORE ALL FRAGMENTS (16)}}
\]

If the record is a fixed length record, this field is not shown on the report since fixed length records cannot be fragmented.

18 NUMBER OF RECORDS COMPRESSED--The number of compressed records and the percentage of compressed records.

\[
PCT = 100 \times \frac{\text{NUMBER RECS COMPRESSED (18)}}{\text{NUMBER OF OCCURRENCES (4)}}
\]

This statistic is valid only if IDMSCOMP was used as the compression routine.

19 SAVINGS DUE TO COMPRESSION--The total number of bytes saved due to the compression of certain record occurrences.

Also included are the average number of bytes of savings per compressed record (AVG), the greatest number of bytes of savings in a compressed record (MAX), and the least number of bytes of savings in a compressed record that has compression savings (MIN).

\[
AVG = \frac{\text{SAVINGS DUE TO COMPRESSION (19)}}{\text{NUMBER RECORDS COMPRESSED (18)}}
\]

If the record is a fixed length record, this field is not shown on the report because compressed records are defined internally as variable length.
This statistic is valid only if IDMSCOMP was used as the compression routine.

**Distribution Section Fields**

The Distribution Section of the Record Report is shown in the following display. This section of the report displays the distribution of the following selected record characteristics.

<table>
<thead>
<tr>
<th>Item</th>
<th>Records</th>
<th>Pages</th>
<th># of Occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. RECORD OCCURRENCES</td>
<td>88001 - 88002</td>
<td>7</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>88003 - 88004</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>88005 - 88006</td>
<td>5</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>88007 - 88008</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>88009 - 88010</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>88011 - 88012</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>88013 - 88014</td>
<td>5</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>88015 - 88015</td>
<td>6</td>
<td>16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Item</th>
<th>Records</th>
<th>Pages</th>
<th># of Fragments</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. FRAGMENTS</td>
<td>88001 - 88002</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>88003 - 88004</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>88005 - 88006</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>88007 - 88008</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>88009 - 88010</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>88011 - 88012</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>88013 - 88014</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>88015 - 88015</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Item</th>
<th>Bytes Saved</th>
<th># of Occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. SAVINGS DUE TO COMPRESSION</td>
<td>0 - 0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1 - 22</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>23 - 44</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>45 - 66</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>67 - 88</td>
<td>0</td>
</tr>
</tbody>
</table>
## Note:
In a case where only a portion of an area has been swept for statistical accumulation, the field values described pertain only to the portion of the area swept.

Remember that statistics shown on the Record Report apply only to the record type indicated in the RECORD ID/NAME field of the report. See Record Report Field Descriptions (see page 164), Identification Fields, item 1.

### 1 RECORD OCCURRENCES
The distribution of record occurrences by page range is shown here. Page ranges are listed under the column headed RANGE. The number of occurrences found in each page range is listed under the column headed TOTAL. The percentage of occurrences found in each page range is listed under the column headed PCT.

\[
PCT = \frac{\text{NUMBER OCCURRENCES IN PAGE RANGE}}{\text{Rec Stats 4}} \times 100
\]

### 2 FRAGMENTS
The distribution of record fragments by page range is shown here. Page ranges are listed under the column headed RANGE. The number of fragments found in each page range is listed under the column headed TOTAL. The percentage of fragments found in each page range is listed under the column headed PCT.

\[
PCT = \frac{\text{NUMBER OF FRAGMENTS}}{\text{FRAGMENTS IN PAGE RANGE}} \times 100
\]

If the record is a fixed length record, this field is not shown on the report since fixed length records cannot be fragmented.

### 3 SAVINGS DUE TO COMPRESSION
The distribution of compressed record occurrences by range of bytes saved is shown here. Ranges of bytes saved are listed in the column headed RANGE. The number of compressed record occurrences found in each range is listed in the column headed TOTAL. The percentage of compressed record occurrences found in each range is listed in the column headed PCT.

\[
PCT = \frac{\text{NUMBER RECS COMPRESSED OCCURRENCES IN BYTE RANGE}}{\text{Rec Stats 18}} \times 100
\]

If the record is a fixed length record, this field is not shown on the report, since fixed length records cannot be compressed. This statistic is valid only if IDMSCOMP was used as the compression routine.
Histogram Section Fields

The Histogram Section of the Record Report is shown in the following display. This section of the report is a graphic representation and a quick reference version of the Distribution Section. Distribution Section percentages are shown for each range in the form of a bar chart.

<table>
<thead>
<tr>
<th>PAGES</th>
<th>RECOR D OCCURRENC ES - PERCENTAGE BY DISTRIBUTION RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>88001 - 88002</td>
<td>.NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN . .</td>
</tr>
<tr>
<td>88003 - 88004</td>
<td>. . . . . .</td>
</tr>
<tr>
<td>88005 - 88006</td>
<td>.NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN . .</td>
</tr>
<tr>
<td>88007 - 88008</td>
<td>.NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN . .</td>
</tr>
<tr>
<td>88009 - 88010</td>
<td>.NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN . .</td>
</tr>
<tr>
<td>88011 - 88012</td>
<td>.NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN . .</td>
</tr>
<tr>
<td>88013 - 88014</td>
<td>.NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN . .</td>
</tr>
<tr>
<td>88015 - 88015</td>
<td>.NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN . .</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PAGES</th>
<th>FRAGMENTS - PERCENTAGE BY DISTRIBUTION RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>88001 - 88002</td>
<td>. . . . . .</td>
</tr>
<tr>
<td>88003 - 88004</td>
<td>. . . . . .</td>
</tr>
<tr>
<td>88005 - 88006</td>
<td>. . . . . .</td>
</tr>
<tr>
<td>88007 - 88008</td>
<td>. . . . . .</td>
</tr>
<tr>
<td>88009 - 88010</td>
<td>. . . . . .</td>
</tr>
<tr>
<td>88011 - 88012</td>
<td>. . . . . .</td>
</tr>
<tr>
<td>88013 - 88014</td>
<td>. . . . . .</td>
</tr>
<tr>
<td>88015 - 88015</td>
<td>. . . . . .</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PAGES</th>
<th>BYTES SAVED</th>
<th>SAVINGS DUE TO COMPRESSION - PERCENTAGE BY DISTRIBUTION RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>88001 - 88002</td>
<td>. . . . . .</td>
<td></td>
</tr>
<tr>
<td>88003 - 88004</td>
<td>. . . . . .</td>
<td></td>
</tr>
<tr>
<td>88005 - 88006</td>
<td>. . . . . .</td>
<td></td>
</tr>
<tr>
<td>88007 - 88008</td>
<td>. . . . . .</td>
<td></td>
</tr>
<tr>
<td>88009 - 88010</td>
<td>. . . . . .</td>
<td></td>
</tr>
<tr>
<td>88011 - 88012</td>
<td>. . . . . .</td>
<td></td>
</tr>
<tr>
<td>88013 - 88014</td>
<td>. . . . . .</td>
<td></td>
</tr>
<tr>
<td>88015 - 88015</td>
<td>. . . . . .</td>
<td></td>
</tr>
</tbody>
</table>
A Comparative Record Report is used to assess changes that have occurred in the physical organization for a record type. Consequently, this type of report uses data from two different statistical files. The data from these files describes the area at two different points in time.

Similar to the Record Report, the Comparative Record Report is divided into three sections--the Record Statistics Section, the Distribution Section, and the Histogram Section.

The format of each section in this report is the same as that of a Record Report except that fields have been added to show "old" and "new" statistical file values and, when possible, the amount of change between these values.

### Comparative Report Field Descriptions

Comparative Record Report Fields are shown in the following screen displays. Fields near the top of the Comparative Record Report are described below. See Area Report Field Descriptions (see page 145) descriptions of the other fields.
### RECORD LOCATION MODE

<table>
<thead>
<tr>
<th></th>
<th>CALC</th>
<th>CALC</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMBER OF OCCURRENCES</td>
<td>37</td>
<td>32</td>
</tr>
<tr>
<td>BYTES TO STORE ALL OCCURRENCES</td>
<td>2,912</td>
<td>4,280</td>
</tr>
<tr>
<td>RECS OUT OF PHYSICAL SEQUENCE</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>NUMBER OF LDEL OCCURRENCES</td>
<td>25</td>
<td>68</td>
</tr>
<tr>
<td>BYTES TO STORE LDEL RECORDS</td>
<td>1,400</td>
<td>48</td>
</tr>
</tbody>
</table>

### OCCURRENCE LENGTH

<table>
<thead>
<tr>
<th></th>
<th>COMPRESSED</th>
<th>DECOMPRESSED</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG</td>
<td>79</td>
<td>79</td>
</tr>
<tr>
<td>MAX</td>
<td>188</td>
<td>188</td>
</tr>
<tr>
<td>MIN</td>
<td>56</td>
<td>56</td>
</tr>
</tbody>
</table>

### FRAGMENTS PER OCCURRENCE

<table>
<thead>
<tr>
<th></th>
<th>AVG</th>
<th>MAX</th>
<th>MIN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.00</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### BYTES TO STORE ALL FRAGMENTS

|                      | 0     | 0    |

### RECORD ID/NAME

1 RECORD ID/NAME--The name of the record type whose change in physical organization is to be assessed.

### NEW FILE CREATION DATA

2 NEW FILE CREATION DATA--The date and time of creation of the file which is listed in the JCL as STAT1.
3 OLD FILE CREATION DATA--The date and time of creation of the statistical file which is listed in the JCL as STAT2.

4 NEW PCT--The values listed under this heading are raw counts and percentages taken from the new statistics file (item 2). See Record Report for descriptions of each field listed.

5 OLD PCT--The values listed under this heading are raw counts and percentages taken from the new statistical file (item 3). See Record Report for a description of each field listed.

6 VARIANCE PCT--Listed under this heading are the differences between new and old raw counts and the percent variance between the old and new values. Positive values indicate the amount by which new statistical file values are greater than old. Negative values indicate the amount by which new file values are less than old.

Distribution Section Fields

The Distribution Section of the Comparative Record Report is shown in the following display. See Area Report Field Descriptions (see page 145), for descriptions of fields not discussed here.

<table>
<thead>
<tr>
<th>ITEM</th>
<th>RANGE</th>
<th>NEW</th>
<th>PCT</th>
<th>OLD</th>
<th>VARIANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>88001 - 88002</td>
<td>PAGES</td>
<td>7</td>
<td>19</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>88003 - 88004</td>
<td>PAGES</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>88005 - 88006</td>
<td>PAGES</td>
<td>5</td>
<td>14</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>88007 - 88008</td>
<td>PAGES</td>
<td>4</td>
<td>11</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>88009 - 88010</td>
<td>PAGES</td>
<td>4</td>
<td>11</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>88011 - 88012</td>
<td>PAGES</td>
<td>5</td>
<td>14</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>88013 - 88014</td>
<td>PAGES</td>
<td>6</td>
<td>16</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>88015 - 88015</td>
<td>PAGES</td>
<td>6</td>
<td>16</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>88011 - 88012</td>
<td>PAGES</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
1 NEW PCT--The values listed under this heading are raw counts and percentages for each range. These values are taken from the new statistics file. See Comparative Record Report (see page 171) Identification Fields, item 2.

2 OLD PCT--The values listed under this heading are raw counts and percentages for each range. These values are taken from the old statistics file. See Comparative Record Report Identification Fields, item 3.

3 VARIANCE PCT--Listed under this heading are the differences between new and old raw counts and the percent variance between the old and new values. Positive values indicate the amount by which new statistical file values are greater than old. Negative values indicate the amount by which new file values are less than old. For a variance to be meaningful, both the old and new statistical files must contain identical ranges. If they do not, no values are listed under this heading.

Histogram Section Fields

The following screen display shows a portion of the Histogram Section of the Comparative Record Report. This section is a graphic representation and a quick reference version of the Distribution Section. Old (O) and new (N) values of Distribution Section percentages for each range are shown on two lines of a bar chart.
Set Report

Contents

- Set Report Field Descriptions (see page 176)
- Distribution Section Fields (see page 181)
- Histogram Section Fields (see page 183)
- Multi-Member Set Analysis Section Fields (see page 184)

A Set Report can be produced for every set type (including SR8 index sets) for which statistics have been accumulated.

Set statistics are accumulated for a set type when all owner records in the set type reside in an area swept for statistical accumulation and all member records in the set type reside either in an area swept or in an area that is accessible.

An accessible area is an area that you have not excluded by using the NOMEMBERSET option in an AREA parameter statement when processing a database for statistical accumulation.

If a member record in any set occurrence in a set type resides in an inaccessible area (excluded by using the NOMEMBERSET option), no statistics for the entire set type are accumulated. Consequently, a Set Report for such a set type is not available. In addition, a Set Report for a set type excluded from processing by a SET parameter statement cannot be produced. See Section 3, "Parameters".

A Set Report is divided into four sections--the Set Statistics Section, the Distribution Section, Histogram Section and the Multi-Member Set Analysis Section.

Set Report fields are shown in the following screen displays. Following are descriptions of two report identification fields plus fields used in the Set Statistics, Distribution, Histogram, and Multi-Member Set Analysis Sections of the Set Report. The item numbers preceding each field name and description correspond with those in the referenced exhibits.

### Set Report Field Descriptions

Identification and Set Statistic Section fields are described below.

<table>
<thead>
<tr>
<th>CA-TOOLS</th>
<th>RELEASE</th>
<th>CA IDMS/DB ANALYZER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rnn.nn</td>
<td></td>
<td>SET REPORT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SET STATISTICS SECTION</td>
</tr>
</tbody>
</table>

1. **SET NAME**: ACTIVITY - TYPES
2. **FILE CREATION DATA**: mm/dd/yy - hh:mm:ss
3. **OWNER RECORD NAME**: ACTIVITY
4. **MEMBER RECORD NAME**: MULTI-MEMBER SET
5. **SET OCCURRENCES**: 3
6. **SET OCCURRENCES WITH MEMBERS**: 3 100
7. **MEMBER RECORD OCCURRENCES**: 9
8. **LDEL OCCURRENCES**: 0 0
9. **LENGTH OF SET OCCURRENCES**:
### SET NAME

The name of the set type for the reported statistics. CALC sets are identified as such and reported by area.

#### Statistics

<table>
<thead>
<tr>
<th>Item</th>
<th>Total</th>
<th>PCT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Owner Record Name</strong></td>
<td>SR7</td>
<td></td>
</tr>
<tr>
<td><strong>Member Record Name</strong></td>
<td>STUDENT</td>
<td></td>
</tr>
<tr>
<td><strong>Set Occurrences</strong></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td><strong>Set Occurrences with Members</strong></td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td><strong>Member Record Occurrences</strong></td>
<td>51</td>
<td></td>
</tr>
<tr>
<td><strong>LDEL Occurrences</strong></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td><strong>Length of Set Occurrences</strong></td>
<td>51.00</td>
<td></td>
</tr>
<tr>
<td><strong>Bytes to Store Owners</strong></td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td><strong>Bytes to Store Members</strong></td>
<td>4,600</td>
<td>69</td>
</tr>
<tr>
<td><strong>Bytes to Store LDELS</strong></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Bytes to Store SR8 Records</strong></td>
<td>2,060</td>
<td>31</td>
</tr>
<tr>
<td><strong>Total Bytes to Store Set</strong></td>
<td>6,692</td>
<td></td>
</tr>
</tbody>
</table>

---

1. **Bytes to Store Owners**: The minimum number of bytes required to store the owner records.
2. **Bytes to Store Members**: The number of bytes required to store the member records.
3. **Bytes to Store LDELS**: The number of bytes required to store the LDELS (Large Data Elements).
4. **Bytes to Store SR8 Records**: The number of bytes required to store the SR8 records.
5. **Total Bytes to Store Set**: The total number of bytes required to store the set.

---

**Page Changes**

<table>
<thead>
<tr>
<th>Item</th>
<th>Average</th>
<th>Maximum</th>
<th>Minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Set Changes</strong></td>
<td>102.00</td>
<td>102</td>
<td>102</td>
</tr>
</tbody>
</table>

---

**Cluster Page Spread**

<table>
<thead>
<tr>
<th>Item</th>
<th>Average</th>
<th>Maximum</th>
<th>Minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Set Changes</strong></td>
<td>18.00</td>
<td>18</td>
<td></td>
</tr>
</tbody>
</table>

---

**Set Statistics Section**

- **Set Name**: IX - STUD - LNAME
- **File Creation Data**: mm/dd/yy - hh:mm:ss
- **Number of Set Occurrences**: 1
- **Number of Member Record Occurrences**: 51
- **Average Length of Set Occurrences**: 51.00
- **Minimum Number of Bytes to Store Owners**: 32
- **Maximum Number of Bytes to Store Members**: 4,600
- **Minimum Number of Bytes to Store LDELS**: 0
- **Total Bytes to Store SR8 Records**: 2,060
- **Total Bytes to Store Set**: 6,692
- **Average Page Changes**: 102.00
- **Minimum Page Changes**: 102
- **Maximum Page Changes**: 102

---

**CA-TOOLS Release**: CA IDMS/DB ANALYZER

**Rnn.nn**

**SET REPORT**

**SET STATISTICS SECTION**
2 FILE CREATION DATA--The date and time that the statistical file used for this report was created. This information helps identify which statistical file is being used since there can be more than one file in storage.

⚠️ Note: In a case where only a portion of an area has been swept forstatistical accumulation, set report field values pertain only to the portion of the area swept. In other words, as long as member records reside in accessible areas, set statistics are accumulated when only a portion of an area is swept. However, statistics for any set occurrences whose owner records reside outside the portion of the area swept are not accumulated and are not reflected in the values whose descriptions follow.

Remember that statistics shown on the Set Report apply only to the set indicated in the SET NAME field of the report. See Set Report Identification Fields, item 1.

3 OWNER RECORD NAME--The name of the owner record in this set. For the CALC set, the owner name will be listed as SR1.

4 MEMBER RECORD NAME--The name of the member record in this set. For multi-member sets, the member name will be listed as Multi-member Set. All of the member names in a multi-member set will be listed in the multi-member set analysis section.

5 SET OCCURRENCES--The total number of owner record occurrences.

6 SET OCCURRENCES WITH MEMBERS--The total number of owner record occurrences that own at least one member record occurrence and the percentage of these non-null occurrences.

\[
PCT=100 \times \frac{SET\ OCCURRENCES\ WITH\ MBRS(6)}{SET\ OCCURRENCES(5)}
\]

7 MEMBER RECORD OCCURRENCES--The total number of member record occurrences.

8 LDEL OCCURRENCES--The number of logically deleted member record occurrences for the set type and the percentage of all member occurrences that are logically deleted.

\[
PCT=100 \times \frac{LDEL\ OCCURRENCES\ (8)}{MEMBER\ RECORD\ OCCURRENCES\ (7)}
\]

9 LENGTH OF SET OCCURRENCES--The average (AVG), maximum (MAX), and minimum (MIN) number of member record occurrences (set length) for owner record occurrences that own at least one member record.

\[
AVG = \frac{MEMBER\ RECORD\ OCCURRENCES\ (7)}{SET\ OCCURRENCES\ WITH\ MEMBERS\ (6)}
\]

Set types in sorted order with high average lengths and high average cluster page spread (item 14) are likely to have unwanted performance overhead during update operations. In such a case, consider reorganization or database expansion.

10 BYTES TO STORE OWNERS--The number of bytes used to store owner record occurrences and the percentage of bytes used to store owner record occurrences.
11 BYTES TO STORE MEMBERS--The number of bytes used to store member record occurrences and the percentage of bytes used to store member record occurrences.

PCT=100 * BYTES TO STORE MEMBERS (11) TOTAL BYTES TO STORE SET (13)

12 BYTES TO STORE LDELS--The number of bytes used to store logically deleted record occurrences and the percentage of bytes used to store logically deleted record occurrences.

PCT=100 * BYTES TO STORE LDELS (12) TOTAL BYTES TO STORE SET (13)

13 TOTAL BYTES TO STORE SET--The total number of bytes used to store owner, member, and logically deleted record occurrences. It is the sum of items 10, 11, and 12 (and, if the set is an SR8 index set, item 20).

14 CLUSTER PAGE SPREAD--The average (AVG) and maximum (MAX) number of different pages on which member record occurrences reside for all set occurrences.

AVG = SET OCCURRENCE PAGES WITH SET MEMBERS (6)

where PAGES WITH SET MEMBERS is derived by taking the number of different overflow pages (on which member records reside) for each set occurrence, and then finding the sum of those numbers of pages for all set occurrences in the set type.

⚠️ Note: Only those set occurrences that have members residing on overflow pages are included in this statistic.

Determine the optimum average cluster page spread by dividing the average set length by the number of record occurrences that can fit on one page. If the actual average cluster page spread value is significantly higher than the optimum, performance when retrieving an entire set occurrence is likely to be poor due to excessive input/output operations. In such a case, consider database reorganization.

15 MEMBERS ON TARGET PAGE--The total number of member record occurrences residing on target page and the percentage of member occurrences residing on target page. (A target page is the page on which the system designated that a record be located if there is space available.)

PCT=100 * MEMBERS ON TARGET PAGE (15) MEMBER RECORD OCCURRENCES (7)

Also included are the average number of member occurrences on target page (AVG), the maximum number of member occurrences on target page for a single set occurrence (MAX), and the minimum number of member occurrences on target page for a single set occurrence (MIN) that had at least one member on the target page.

AVG = MEMBERS ON TARGET PAGE (15) SET OCCURRENCES WITH MEMBERS (6)

These fields appear only when at least one set member is stored via this set.
Note: If there are a large number of set occurrences without members, it is possible for the AVG members on target page to be less than the MIN members on target page.

16 MEMBERS ON OVERFLOW--The total number of member record occurrences on overflow and the percentage of member occurrences on overflow. (A record occurrence located on a page other than its target page is said to be on overflow.)

\[ \text{PCT} = \frac{\text{MEMBERS ON OVERFLOW (16)}}{\text{MEMBERS ON OVERFLOW (16)}} \times 100 \]

Also included are the average number of member records on overflow (AVG), the maximum number of member records on overflow for a single set occurrence (MAX), and the minimum number of member records on overflow for a single set occurrence that had at least one member on overflow (MIN).

\[ \text{AVG} = \frac{\text{MEMBERS ON OVERFLOW (16)}}{\text{SET OCCURRENCES WITH MEMBERS (6)}} \]

These fields appear only when at least one set member is stored VIA this set.

Note: If there are a large number of set occurrences without members, it is possible for the AVG members on overflow to be less than the MIN members on overflow.

17 MEMBERS NOT VIA THIS SET--The total number of member record occurrences that are members of this set but are not VIA this set and the percentage of the total number of member records that are not VIA this set.

\[ \text{PCT} = \frac{\text{MEMBERS NOT VIA THIS SET (17)}}{\text{MEMBERS NOT VIA THIS SET (17)}} \times 100 \]

Also included are the average number of member records not VIA this set (AVG), the maximum number of member records not VIA this set for a single set occurrence (MAX), and the minimum number of member records not VIA this set for a single set occurrence that had at least one member not VIA the set (MIN).

This field appears only on multi-member set reports.

\[ \text{AVG} = \frac{\text{MEMBERS NOT VIA THIS SET (17)}}{\text{SET OCCURRENCES WITH MEMBERS (6)}} \]

This field will only be printed for non-CALC, VIA sets.

Note: If there are a large number of set occurrences without members, it is possible for the AVG number of members not VIA this set to be less than the MIN number of members not VIA this set.
18 OCCURRENCES W/O PAGE CHANGES—The number of set occurrences whose members are contained on the same page as their owner records (hence no page changes while walking the set) and the percentage of total set occurrences with members that do not have page changes while walking the set occurrence.

\[ \text{PCT} = 100 \times \frac{\text{OCCURRENCES W/O PAGE CHNGS (18)}}{\text{SET OCCURRENCE S WITH MBERS (6)}} \]

19 PAGE CHANGES—The average number of page changes required to walk set occurrences (AVG), the greatest number of page changes required to walk a single set occurrence (MAX), and the fewest number of page changes required while walking a set occurrence with at least one page change (MIN). If any of the member records are fragmented, the user may encounter more page changes when walking the set in CA IDMS; CA IDMS/DB Analyzer does not recompose fragments during SET-WALK processing.

\[ \text{AVG} = \frac{\text{PAGE CHANGES, ALL SET OCcs (6)}}{\text{SET OCCURRENCE S WITH MBERS (6)}} \]

If the average number of page changes is significantly higher than the optimum average cluster page spread (see item 14), performance when retrieving an entire set occurrence is likely to be poor. In such a case, consider database reorganization.

**Note:** If there are a large number of set occurrences with no page changes, it is possible for the AVG page changes to be less than the MIN page changes.

20 BYTES TO STORE SR8 RECORDS—The number of bytes used to store SR8 index record occurrences and the percentage of bytes used to store SR8 record occurrences.

\[ \text{PCT} = 100 \times \frac{\text{BYTES TO STORE SR8 RECORDS (20)}}{\text{TOTAL BYTES TO STORE SET (13)}} \]

This field will only be present if the report is for an SR8 (integrated index) set.

**Distribution Section Fields**

The Distribution Section of the Set Report is shown in the following display. This section of the report displays the distribution of the following set features.

<table>
<thead>
<tr>
<th>CA-TOOLS</th>
<th>RELEASE</th>
<th>CA IDMS/DB ANALYZER</th>
<th>SET REPORT</th>
<th>DISTRIBUTION SECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rnn.nn</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SET NAME: IX - STUD - LNAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FILE CREATION DATA: mm/dd/yy - hh:mm:ss</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ITEM</th>
<th>RANGE</th>
<th>TOTAL</th>
<th>PCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CLUSTER PAGE SPREAD</td>
<td>PAGES</td>
<td># SET OCCURRENCES</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
### Note

In a case where only a portion of an area has been swept for statistical accumulation, set report field values pertain only to the portion of the area swept. In other words, as long as member records reside in accessible areas, set statistics are accumulated when only a portion of an area is swept. However, statistics for any set occurrences whose owner records reside outside the portion of the area swept are not accumulated and are not reflected in the values whose descriptions follow.

Remember that statistics shown on the Set Report apply only to the set type indicated in the SET NAME field of the report. See Set Report Identification Fields, item 1.

### 1 CLUSTER PAGE SPREAD

The distribution of set occurrences by number of different overflow pages on which member record occurrences reside is shown here. The number of different overflow pages are listed under the column headed RANGE. The number of set occurrences with all members on the corresponding number of overflow pages is listed in the column headed TOTAL. The percentage of set occurrences found in each page range is listed in the column headed PCT.

<table>
<thead>
<tr>
<th>RANGE</th>
<th>SET OCS WITH MEMBERS</th>
<th>PCT=100</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>27</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>31</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>32</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>33</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>34</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>35</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>36</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>37</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>38</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>39</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>41</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>42</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>43</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>44</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>46</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>47</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>48</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>49</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>51</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>52</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>53</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>54</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>55</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>56</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>57</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>58</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>59</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>60</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>61</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>62</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>63</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>64</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>65</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>66</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>67</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>68</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>69</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>70</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>71</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>72</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>73</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>74</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>75</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>76</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>77</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>78</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>79</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>80</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>81</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Note: Set occurrences with all members on the target page are not included in this statistic.

2 LENGTH OF SET OCCURRENCES—The distribution by number of member record occurrences per set occurrence is shown here. Ranges of member occurrences are listed under the column headed RANGE. The number of set occurrences whose member record count falls within each range is listed in the column headed TOTAL.

The percentage of set occurrences falling within each range is listed in the column headed PCT.

Histogram Section Fields

The Histogram Section of the Set Report is shown in the following display. This section of the report is a graphic representation and a quick reference version of the Set Report Distribution Section. Distribution Section percentages are shown for each range in the form of a bar chart.

<table>
<thead>
<tr>
<th>CA-TOOLS</th>
<th>RELEASE</th>
<th>CA IDMS/DB ANALYZER</th>
<th>SET REPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>HISTOGRAM SECTION</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| SET NAME: IX - STUD - LNAME |
| FILE CREATION DATA: mm/dd/yy - hh:mm:ss |</p>
<table>
<thead>
<tr>
<th>MBR OCCURRENCES</th>
<th>CLUSTER PAGE SPREAD</th>
<th>PERCENTAGE BY DISTRIBUTION RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 10</td>
<td>0 - 10</td>
<td>0 - 60</td>
</tr>
<tr>
<td>1 - 20</td>
<td>1 - 20</td>
<td>1 - 60</td>
</tr>
<tr>
<td>21 - 9999</td>
<td>21 - 9999</td>
<td>21 - 60</td>
</tr>
</tbody>
</table>

MBR OCCURRENCES | LENGTH OF SET OCCURRENCES | PERCENTAGE BY DISTRIBUTION RANGE |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 10</td>
<td>0 - 10</td>
<td>0 - 60</td>
</tr>
<tr>
<td>1 - 10</td>
<td>1 - 10</td>
<td>1 - 60</td>
</tr>
</tbody>
</table>
Multi-Member Set Analysis Section Fields

The Multi-Member Set Analysis Section of the Set Report presents an analysis by record type of record occurrences found in a multi-member set. The Section fields are shown in the following display.

Note: When only a portion of an area has been swept for statistical accumulation, set report field values pertain only to the portion of the area swept; i.e., as long as member records reside in accessible areas, set statistics are accumulated when only a portion of an area is swept. However, statistics for any set occurrences whose owner records reside outside the portion of the area swept are not accumulated and are not reflected in the values described below.

Remember that statistics shown on the Set Report apply only to the set indicated in the SET NAME field of the report. See Set Report Identification Fields, item 1.
1 RECORD ID--Listed under this heading are the record identification numbers for those record types whose occurrences act as member records for this set.

2 RECORD NAME--Listed under this heading are the record names for each record type.

3 OCCURRENCES--The values listed under this heading are the total number of active multi-member record occurrences for each record type and the percentage of active multi-member record occurrences.

   PCT=100 * OCCURRENCES PER RECORD TYPE MEMBER REC OCS (Set Stats 7)

4 SPACE USED--The values listed under this heading are the total number of bytes used by active multi-member record occurrences and the percentage of bytes used by active multi-member record occurrences for each record type.

   PCT=100 * ACTIVE SPACE USED PER REC TYPE BYTES STORE MEMBERS (Set Statistics 11)

5 MEMBERS PER SET OCCURRENCE--The values listed under this heading are the average number of active multi-member record occurrences in a set occurrence (AVG), the number of active multi-member record occurrences in the set occurrence with the greatest number of multi-member record occurrences (MAX), and the number of active multi-member record occurrences in the set occurrence with the least number of multi-member record occurrences (MIN) for each record type.

   AVG = OCCURRENCES PER RECORD TYPE (3) SET OCS, MEMBERS (Set Statistics 6)

   Caution: If there are a large number of set occurrences without members, it is possible for the AVG number of members per set occurrence to be less than the MIN number of members per set occurrence.

6 OCCURRENCES--The values listed under this heading are the total number of logically deleted multi-member record occurrences for each record type.

7 SPACE USED--The values listed under this heading are the total number of bytes used by logically deleted multi-member record occurrences and the percentage of bytes used by logically deleted multi-member record occurrences for each record type.

   PCT=100 * LDEL SPACE USED PER REC TYPE BYTES TO STORE LDELS (Set Statistics 12)

---

**Comparative Set Report**

**Contents**

- Set Statistic Section Fields (see page 186)
- Distribution Section Fields (see page 188)
- Histogram Section Fields (see page 190)
A Comparative Set Report is used to assess changes that have occurred in the physical contents for a set type. Consequently, this type of report uses data from two different statistical files. The data from these files describes the set type at two different points in time.

Similar to the Set Report, the Comparative Set Report is divided into four sections: the Set Statistics Section, the Distribution Section, the Histogram Section, and the Multi-Member Set Analysis Section.

The format of each section in this report is the same as that of a Set Report except that fields have been added to show "old" and "new" statistical file values and, when possible, the amount of change between these values.

Set Comparative Report Fields are shown in the following screen displays. Fields near the top of the Comparative Set Report are described below. See Set Report (see page 176) for descriptions of fields identical to those found in the Set Report.

**Set Statistic Section Fields**

<table>
<thead>
<tr>
<th>CA-TOOLS</th>
<th>RELEASE</th>
<th>CA IDMS/DB ANALYZER</th>
<th>SET REPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rnn.nn</td>
<td>SET REPORT</td>
<td>dd/mm/yy</td>
<td>hh:mm:ss</td>
</tr>
<tr>
<td>STATISTICS SECTION</td>
<td></td>
<td>STATISTICS SECTION</td>
<td></td>
</tr>
<tr>
<td>COMPARATIVE</td>
<td></td>
<td>COMPARATIVE</td>
<td></td>
</tr>
<tr>
<td>SET NAME: ACTIVITY-TYPES</td>
<td></td>
<td>ACTIVITY-TYPES</td>
<td></td>
</tr>
<tr>
<td>NEW FILE CREATION DATA: mm/dd/yy - hh:mm:ss</td>
<td></td>
<td>mm/dd/yy - hh:mm:ss</td>
<td></td>
</tr>
<tr>
<td>OLD FILE CREATION DATA: mm/dd/yy - hh:mm:ss</td>
<td></td>
<td>mm/dd/yy - hh:mm:ss</td>
<td></td>
</tr>
<tr>
<td>ITEM</td>
<td>NEW</td>
<td>PCT</td>
<td>OLD</td>
</tr>
<tr>
<td>-- --</td>
<td>---</td>
<td>----</td>
<td>-----</td>
</tr>
<tr>
<td>OWNER RECORD NAME</td>
<td>ACTIVITY</td>
<td>ACTIVITY</td>
<td></td>
</tr>
<tr>
<td>MEMBER RECORD NAME</td>
<td>MEMBER</td>
<td>MEMBER</td>
<td></td>
</tr>
<tr>
<td>SET OCCURRENCES</td>
<td>OCCURRENCES</td>
<td>OCCURRENCES</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>SET OCCURRENCES W/MEMBERS</td>
<td>3</td>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>MEMBER RECORD OCCURRENCES</td>
<td>9</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>LDEL OCCURRENCES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LENGTH OF SET OCCURRENCES</td>
<td>AVG</td>
<td>MAX</td>
<td>MIN</td>
</tr>
<tr>
<td>3.00</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>MAX</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>MIN</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>BYTES TO STORE OWNERS</td>
<td>180</td>
<td>27</td>
<td>180</td>
</tr>
<tr>
<td>BYTES TO STORE MEMBERS</td>
<td>480</td>
<td>73</td>
<td>480</td>
</tr>
<tr>
<td>BYTES TO STORE LDELS</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TOTAL BYTES TO STORE SET</td>
<td>660</td>
<td>660</td>
<td></td>
</tr>
<tr>
<td>CLUSTER PAGE SPREAD</td>
<td>AVG</td>
<td>MAX</td>
<td></td>
</tr>
<tr>
<td>1.67</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Item</td>
<td>NEW</td>
<td>PCT</td>
<td>OLD</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td><strong>MEMBERS OF TARGET PAGE</strong></td>
<td>6</td>
<td>67</td>
<td>6</td>
</tr>
<tr>
<td>AVG</td>
<td>2.00</td>
<td></td>
<td>2.00</td>
</tr>
<tr>
<td>MAX</td>
<td>3</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>MIN</td>
<td>3</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td><strong>MEMBERS ON OVERFLOW</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AVG</td>
<td>0.00</td>
<td></td>
<td>0.00</td>
</tr>
<tr>
<td>MAX</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>MIN</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td><strong>MEMBERS NOT VIA THIS SET</strong></td>
<td>3</td>
<td>33</td>
<td>3</td>
</tr>
<tr>
<td>AVG</td>
<td>1.00</td>
<td></td>
<td>1.00</td>
</tr>
<tr>
<td>MAX</td>
<td>3</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>MIN</td>
<td>3</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td><strong>OCCURRENCES W/O PAGE CHANGES</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PAGE CHANGES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td>1.67</td>
<td></td>
<td>1.67</td>
</tr>
<tr>
<td>MAX</td>
<td>3</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>MIN</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td><strong>CLUSTER PAGE SPREAD</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td>11.00</td>
<td></td>
<td>18.00</td>
</tr>
<tr>
<td>MAX</td>
<td>11</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td><strong>OCCURRENCES W/O PAGE CHANGES</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PAGE CHANGES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td>32.00</td>
<td></td>
<td>102.00</td>
</tr>
<tr>
<td>MAX</td>
<td>32</td>
<td></td>
<td>102</td>
</tr>
<tr>
<td>MIN</td>
<td>32</td>
<td></td>
<td>102</td>
</tr>
</tbody>
</table>

**CA-TOOLS RELEASE**

**CA-IDMS/DB ANALYZER**

**SET REPORT**

**STATISTICS SECTION**

**COMPARATIVE**

**SET NAME:** IX-STUD-LNAME

**NEW FILE CREATION DATA:** mm/dd/yy - hh:mm:ss

**OLD FILE CREATION DATA:** mm/dd/yy - hh:mm:ss

**ITEM**

**OWNER RECORD NAME**

SR7

**MEMBER RECORD NAME**

STUDENT

**SET OCCURRENCES**

1

**SET OCCURRENCES W/MEMBERS**

1 100 1 100

**MEMBER RECORD OCCURRENCES**

16 51

**LDEL OCCURRENCES**

0 0 0 0

**LENGTH OF SET OCCURRENCES**
### Distribution Section Fields

The Distribution Section of the Comparative Set Report is shown in the following display. See Area Report Field Descriptions (see page 145) for descriptions of fields not discussed here.

<table>
<thead>
<tr>
<th>CA-TOOLS</th>
<th>RELEASE</th>
<th>CA IDMS/DB ANALYZER</th>
<th>DATE</th>
<th>TIME</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rnn.nn</td>
<td></td>
<td>SET REPORT</td>
<td>dd/mm/yy</td>
<td>hh:mm:ss</td>
<td>nnnn</td>
</tr>
</tbody>
</table>

#### CA IDMS - 19.0

<table>
<thead>
<tr>
<th>AVG</th>
<th>16.00</th>
<th>51.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX</td>
<td>16</td>
<td>51</td>
</tr>
<tr>
<td>MIN</td>
<td>16</td>
<td>51</td>
</tr>
<tr>
<td>BYTES TO STORE OWNERS</td>
<td>32</td>
<td>2</td>
</tr>
<tr>
<td>BYTES TO STORE MEMBERS</td>
<td>1,464</td>
<td>70</td>
</tr>
<tr>
<td>BYTES TO STORE LDELS</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BYTES TO STORE SR8 RECORDS</td>
<td>588</td>
<td>28</td>
</tr>
<tr>
<td>TOTAL BYTES TO STORE SET</td>
<td>2,084</td>
<td>6,692</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AVG</th>
<th>11.00</th>
<th>18.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX</td>
<td>11</td>
<td>18</td>
</tr>
<tr>
<td>OCCURRENCES W/O PAGE CHANGES</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PAGE CHANGES</td>
<td>AVG</td>
<td>32.00</td>
</tr>
<tr>
<td></td>
<td>MAX</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>MIN</td>
<td>32</td>
</tr>
</tbody>
</table>

1 **SET NAME**--The name of the set type whose change in physical organization is to be assessed.

2 **NEW FILE CREATION DATA**--The date and time of creation of the statistical file listed in the z/OS and OS/390 JCL as STAT1, and in the VSE/ESA JCL as SYS010.

3 **OLD FILE CREATION DATA**--The date and time of creation of the statistical file listed in the z/OS and OS/390 JCL as STAT2, and in the VSE/ESA JCL as SYS011.

4 **NEW PCT**--The values listed under this heading are raw counts and percentages taken from the new statistical file (item 2). See Set Report (see page 176) for descriptions of each field listed.

5 **OLD PCT**--The values listed under this heading are raw counts and percentages taken from the old statistical file (item 3). See Area Report Field Descriptions (see page 145) for a description of each field listed.

6 **VARIANCE PCT**--Listed under this heading are the differences between new and old raw counts and the percent variance between the old and new values. Positive values indicate the amount by which new statistical file values are greater than old. Negative values indicate the amount by which new file values are less than old.
## DISTRIBUTION SECTION

### COMPARATIVE

**SET NAME:** IX-STUD-LNAME  
**NEW FILE CREATION DATA:** mm/dd/yy - hh:mm:ss  
**OLD FILE CREATION DATA:** mm/dd/yy - hh:mm:ss

<table>
<thead>
<tr>
<th>ITEM</th>
<th>RANGE</th>
<th>NEW</th>
<th>PCT</th>
<th>OLD</th>
<th>PCT</th>
<th>VARIANCE</th>
<th>PCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLUSTER PAGE SPREAD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>15</td>
<td>1</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>100</td>
</tr>
<tr>
<td>16</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>100</td>
<td>-1</td>
<td>100</td>
</tr>
<tr>
<td>21</td>
<td>99999</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LENGTH OF SET OCCURRENCES</th>
<th>MBR OCCURRENCES</th>
<th># SET OCCURRENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>21</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>31</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>41</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>51</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>61</td>
<td>70</td>
<td>0</td>
</tr>
<tr>
<td>71</td>
<td>80</td>
<td>0</td>
</tr>
<tr>
<td>81</td>
<td>99999</td>
<td>0</td>
</tr>
</tbody>
</table>

1 **NEW PCT**--The values listed under this heading are raw counts and percentages for each range. These values are taken from the new statistical file. See Comparative Set Report (see page 185) Identification Fields, item 2.

2 **OLD PCT**--The values listed under this heading are raw counts and percentages for each range. These values are taken from the old statistical file. See Area Report Field Descriptions (see page 145) Identification Fields, item 3.

3 **VARIANCE PCT**--Listed under this heading are the differences between new and old raw counts and the percent variance between the old and new values. Positive values indicate the amount by which new statistical file values are greater than old. Negative values indicate the amount by which new file values are less than old.
For variance to be meaningful, both the old and new statistical files must contain identical ranges. If they do not, no values are listed under this heading.

Histogram Section Fields

A portion of the Histogram Section of the Comparative Set Report is shown in the following display. This section of the report is a graphic representation and a quick reference version of the Distribution Section. Old (O) and new (N) values of Distribution Section percentages for each range are shown on two lines in the form of a bar chart.

<table>
<thead>
<tr>
<th>CA-TOOLS</th>
<th>RELEASE</th>
<th>CA IDMS/DB ANALYZER</th>
<th>SET REPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SET NAME:</td>
<td>IX - STUD - LNAME</td>
<td>NEW FILE CREATION DATA: mm/dd/yy - hh:mm:ss</td>
<td>OLD FILE CREATION DATA: mm/dd/yy - hh:mm:ss</td>
</tr>
<tr>
<td>MBR OCCURRENCES</td>
<td>CLUSTER PAGE SPREAD - PERCENTAGE BY DISTRIBUTION RANGE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

0----+----10----+----20----+----30----+----40----+----50----+----60----+----70----+----80----+----90----+----100

1 - 1 . . . . . . . . . . . .
1 - 1 . . . . . . . . . . . .
2 - 2 . . . . . . . . . . . .
2 - 2 . . . . . . . . . . . .
3 - 3 . . . . . . . . . . . .
3 - 3 . . . . . . . . . . . .
4 - 4 . . . . . . . . . . . .
4 - 4 . . . . . . . . . . . .
5 - 5 . . . . . . . . . . . .
5 - 5 . . . . . . . . . . . .
6 - 7 . . . . . . . . . . . .
6 - 7 . . . . . . . . . . . .
8 - 10 . . . . . . . . . . . .
8 - 10 . . . . . . . . . . . .
11 - 15 . . . . . . . . . . . .
11 - 15 . . . . . . . . . . . .
16 - 20 . . . . . . . . . . . .
16 - 20 . . . . . . . . . . . .
21 - 99999 . . . . . . . . . . . .
21 - 99999 . . . . . . . . . . . .

0----+----10----+----20----+----30----+----40----+----50----+----60----+
Multi-Member Set Analysis Section Fields

The Multi-Member Set Analysis Section of the Comparative Set Report is shown in the following display. This section of the report is identical to its Set Report counterpart except that it lists statistics from both the old and new statistical files. See Set Report (see page 176) for a description of fields in this section.

<table>
<thead>
<tr>
<th>MBR OCCURRENCES</th>
<th>LENGTH OF SET OCCURRENCES</th>
<th>PERCENTAGE BY DISTRIBUTION RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>30</td>
<td>40</td>
<td>60</td>
</tr>
<tr>
<td>50</td>
<td>70</td>
<td>90</td>
</tr>
<tr>
<td>80</td>
<td>100</td>
<td>120</td>
</tr>
<tr>
<td>110</td>
<td>130</td>
<td>150</td>
</tr>
<tr>
<td>170</td>
<td>190</td>
<td>210</td>
</tr>
<tr>
<td>230</td>
<td>250</td>
<td>270</td>
</tr>
<tr>
<td>290</td>
<td>310</td>
<td>330</td>
</tr>
<tr>
<td>350</td>
<td>370</td>
<td>390</td>
</tr>
<tr>
<td>410</td>
<td>430</td>
<td>450</td>
</tr>
<tr>
<td>470</td>
<td>490</td>
<td>510</td>
</tr>
<tr>
<td>530</td>
<td>550</td>
<td>570</td>
</tr>
<tr>
<td>590</td>
<td>610</td>
<td>630</td>
</tr>
<tr>
<td>650</td>
<td>670</td>
<td>690</td>
</tr>
<tr>
<td>710</td>
<td>730</td>
<td>750</td>
</tr>
<tr>
<td>770</td>
<td>790</td>
<td>810</td>
</tr>
<tr>
<td>830</td>
<td>850</td>
<td>870</td>
</tr>
<tr>
<td>890</td>
<td>910</td>
<td>930</td>
</tr>
<tr>
<td>950</td>
<td>970</td>
<td>990</td>
</tr>
</tbody>
</table>

Multi-Member Set Analysis Section Fields

<table>
<thead>
<tr>
<th>CA-TOOLS</th>
<th>RELEASE</th>
<th>CA IDMS/DB ANALYZER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rnn.nn</td>
<td>SET REPORT</td>
<td></td>
</tr>
</tbody>
</table>

MULTI-MEMBER SET ANALYSIS SECTION

COMPARATIVE

SET NAME: STUDENT-REPORTS

NEW FILE CREATION DATA: mm/dd/yy - hh:mm:ss
OLD FILE CREATION DATA: mm/dd/yy - hh:mm:ss

RECORD RECORD *************ACTIVE RECORDS************
SR8 Index Report

Contents

- SR8 Index Report Fields (see page 192)
- Table Entry Distribution Section Fields (see page 195)
- Histogram Section Fields (see page 196)

You can produce an SR8 Index Report for every SR8 index set for which statistics have been accumulated. Reports for SR8 index sets excluded from accumulation by a SET parameter statement cannot be produced.

Additional information on each SR8 (integrated index) set can be found in the Set Report.

An SR8 Index Report is divided into three sections: the SR8 Index Statistics Section, the Table Entry Distribution Section, and the Histogram Section.

SR8 Index Report Fields

SR8 Index Report fields are shown in the following screen displays. Following are descriptions of two report identification fields plus fields used in the Index Statistics, Distribution, and Histogram Sections of the Index Report. The item numbers preceding each field name and description correspond with those in the referenced exhibits.

Listed by record type are statistics from the new statistical file.

Listed by record type are statistics from the old statistical file.
<table>
<thead>
<tr>
<th></th>
<th>DESCRIPTION</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>MEMBERSHIP OPTION</td>
<td>MANDATORY AUTO</td>
</tr>
<tr>
<td>6</td>
<td>DUPLICATES OPTION</td>
<td>DUPS NOT</td>
</tr>
<tr>
<td>7</td>
<td>SEQUENCE OPTION</td>
<td>DESCENDING SEQ</td>
</tr>
<tr>
<td>8</td>
<td>COMPRESSION OPTION</td>
<td>NO</td>
</tr>
<tr>
<td>9</td>
<td>SET ORDER OPTION</td>
<td>SORTED</td>
</tr>
<tr>
<td>10</td>
<td>SORT TYPE</td>
<td>SORTED DBKEY</td>
</tr>
<tr>
<td>11</td>
<td>MAX NUMBER OF TABLE ENTRIES</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>MAX LENGTH OF TABLE ENTRY</td>
<td>8</td>
</tr>
<tr>
<td>13</td>
<td>SYMBOLIC KEY LENGTH</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>MAX SR8 RECORD SIZE</td>
<td>72</td>
</tr>
<tr>
<td>15</td>
<td>TOTAL BYTES TO STORE INDEX</td>
<td>2,060</td>
</tr>
<tr>
<td>16</td>
<td>PAGE NUMBER - 1ST SR8</td>
<td>92,129</td>
</tr>
<tr>
<td>17</td>
<td>NUMBER OF BOTTOM SR8 IN TOP</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>SR8 DISPLACEMENT</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>NUMBER OF LEVELS</td>
<td>5</td>
</tr>
<tr>
<td>20</td>
<td>TOTAL ORPHAN COUNT</td>
<td>124</td>
</tr>
</tbody>
</table>

**UPPER LEVEL STATISTICS**

<table>
<thead>
<tr>
<th></th>
<th>DESCRIPTION</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>SR8 RECORDS</td>
<td>16</td>
</tr>
<tr>
<td>22</td>
<td>PAGE SPREAD</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>USED TABLE ENTRIES</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>AVG</td>
<td>2.31</td>
</tr>
<tr>
<td></td>
<td>MAX</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>MIN</td>
<td>2</td>
</tr>
<tr>
<td>24</td>
<td>ORPHAN COUNT</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>AVG</td>
<td>1.63</td>
</tr>
<tr>
<td></td>
<td>MAX</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>MIN</td>
<td>1</td>
</tr>
</tbody>
</table>

**BOTTOM LEVEL STATISTICS**

<table>
<thead>
<tr>
<th></th>
<th>DESCRIPTION</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>SR8 RECORDS</td>
<td>22</td>
</tr>
<tr>
<td>26</td>
<td>PAGE SPREAD</td>
<td>1</td>
</tr>
<tr>
<td>27</td>
<td>USED TABLE ENTRIES</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>AVG</td>
<td>2.32</td>
</tr>
<tr>
<td></td>
<td>MAX</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>MIN</td>
<td>2</td>
</tr>
<tr>
<td>28</td>
<td>ORPHAN COUNT</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>AVG</td>
<td>4.45</td>
</tr>
<tr>
<td></td>
<td>MAX</td>
<td>10</td>
</tr>
</tbody>
</table>

**INDEX NAME**--The name of the SR8 index set for the reported statistics.

**FILE CREATION DATA**--The date and time that the statistical file used for this report was created. This information helps identify which statistical file is being used since there can be more than one file in storage.

**OWNER RECORD NAME**--The name of the index owned by the system or specified by the user.

**MEMBER RECORD NAME**--The name of the record type that is being indexed.

**MEMBERSHIP OPTION**--The name of the actual membership option for the index set. One of the following four options appears on the report:

- MANDATORY-AUTO
- OPTIONAL-AUTO
- OPTIONAL-MAN
- MANDATORY-MAN

**DUPLICATES OPTION**--The name of the duplicates option specified in the source code for the set section of the schema. One of four options appears on the report:

- N/A
- DUPS FIRST
- DUPS LAST
- DUPS NOT
7 **SEQUENCE OPTION**--Either ASCENDING or DESCENDING depending on the order in which member record occurrences are to be stored in the index set.

8 **COMPRESSION OPTION**--Either YES or NO, depending on whether front-end db-key compression was specified.

9 **SET ORDER OPTION**--The name of the sort option specified for the SR8 index set. One of three options appears on the report:

   SORTED
   NEXT/FIRST
   PRIOR/LAST

10 **SORT TYPE**--The name of the sort type specified for the SR8 index set. One of three options appears on the report:

   SORTED DBKEY
   SORTED SYMBOLIC KEY
   UNSORTED

11 **MAX NUMBER OF TABLE ENTRIES**--The maximum number of table entries that can be made in an SR8 index record.

12 **MAX LENGTH OF TABLE ENTRY**--The maximum length of each entry that can be entered into an SR8 index record. The length depends upon which sort type is specified.

13 **SYMBOLIC KEY LENGTH**--The maximum number of bytes defined for the sort key specified.

14 **MAX SR8 RECORD SIZE**--The size of the largest SR8 index record encountered in this SR8 index set.

15 **TOTAL BYTES TO STORE INDEX**--The total storage space requirements for all SR8 and SR7 or user-specified owner records of the index set.

16 **PAGE NUMBER - 1ST SR8**--The page number of the first SR8 record encountered that is pointed to by an owner record.

17 **NUMBER OF BOTTOM SR8 IN TOP**--The number of bottom level SR8 records stored in the top level of the index set.

18 **SR8 DISPLACEMENT**--The number of pages the bottom level SR8 index records are stored from their owners.

19 **NUMBER OF LEVELS**--The maximum number of levels in the hierarchy for any owner occurrence within this SR8 index set.

20 **TOTAL ORPHAN COUNT**--The total number of orphan counts within this SR8 index set. This number is the sum of the upper level (item 24) and bottom level (item 28) orphan counts.

21 **SR8 RECORDS**--The number of SR8 index record occurrences in the upper level of the index set.

22 **PAGE SPREAD**--The number of pages on which upper level SR8 index record occurrences occur.

23 **USED TABLE ENTRIES**--The number of filled table entries for all upper level SR8 record occurrences in the SR8 index set.
Also included are the average (AVG) number of filled table entries, the greatest number of filled table entries (MAX), and the fewest number of filled table entries (MIN) in an upper level SR8 index record occurrence that had at least one filled table entry.

\[
AVG = \frac{\text{USED TABLE ENTRIES} (23)}{\text{Upper SR8 INDEX RECORDS} (21)}
\]

**24 ORPHAN COUNT**—The total number of orphan counts for all SR8 records within the upper level of this set.

Also included are the average (AVG) number of orphans per upper level SR8 index record occurrence, the greatest number of orphans (MAX), and the fewest number of orphans (MIN) per upper level SR8 index record occurrence.

\[
AVG = \frac{\text{Upper ORPHAN COUNT} (24)}{\text{Upper SR8 INDEX RECORDS} (21)}
\]

**25 SR8 RECORDS**—The number of SR8 index record occurrences in the bottom level of the index set.

**26 PAGE SPREAD**—The number of pages on which bottom level SR8 record occurrences occur.

**27 USED TABLE ENTRIES**—The number of filled table entries for all bottom level SR8 index record occurrences in the index set.

Also included are the average (AVG) number of filled table entries, the greatest number of filled table entries (MAX), and the fewest number of filled table entries (MIN) in a bottom level SR8 index record occurrence that had at least one filled table entry.

\[
AVG = \frac{\text{USED TABLE ENTRIES} (27)}{\text{Bottom SR8 INDEX RECORDS} (25)}
\]

**28 ORPHAN COUNT**—The total number of orphan counts for all SR8 records within the bottom level of this set.

Also included are the average (AVG) number of orphans per bottom level SR8 index record occurrence, the greatest number of orphans (MAX), and the fewest number of orphans (MIN) per bottom level SR8 index record occurrence.

\[
AVG = \frac{\text{Bottom ORPHAN COUNT} (28)}{\text{Bottom SR8 INDEX RECORDS} (25)}
\]

### Table Entry Distribution Section Fields

The Distribution Section of the SR8 Index Report is shown in the following display. This section of the report displays the distribution of the following selected SR8 index features.

<table>
<thead>
<tr>
<th>CA-TOOLS</th>
<th>RELEASE</th>
<th>CA IDMS/DB ANALYZER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rnn.nn</td>
<td></td>
<td>SR8 INDEX REPORT</td>
</tr>
</tbody>
</table>

**INDEX NAME:** IX-STUD-LNAME

**FILE CREATION DATA:** mm/dd/yy - hh:mm:ss

**ITEM**

- ------------------------
- RANGE TOTAL P

---

16-Jan-2018 195/898
1  UPPER LVL TBL ENTRIES USED  TABLE ENTRIES  # OF OCCURRENCES
   0 - 0  0
   1 - 1  0
   2 - 2  11
   3 - 3  5

2  BOTTOM LVL TBL ENTRIES USED  TABLE ENTRIES  # OF OCCURRENCES
   0 - 0  0
   1 - 1  0
   2 - 2  15
   3 - 3  7
   401-450 23

1  UPPER LVL TBL ENTRIES USED--The distribution of upper level SR8 index record occurrences by number of table entries used is shown here. Numbers of table entries used are listed under the column headed TABLE ENTRIES. The number of SR8 index record occurrences using the corresponding number of table entries is listed under TOTAL. The percentage of SR8 index records using each number of table entries is listed in the column headed PCT.

PCT=100  *  UP LVL TBL ENTS USED PER RANGE  Upper SR8 RECO
RDS (Index Stats 21)

2  BOTTOM LVL TBL ENTRIES USED--The distribution of bottom level SR8 index record occurrences by number of table entries used is shown here. Numbers of table entries used are listed under the column headed TABLE ENTRIES. The number of SR8 index record occurrences using the corresponding number of table entries is listed under TOTAL. The percentage of SR8 index records using each number of table entries is listed in the column headed PCT.

PCT=100  *  BOT LVL TBL ENTS USED PER RNGE  Bottom SR8 REC
ORDS (Index Stats 25)

Histogram Section Fields

The Histogram Section of the SR8 Index Report is shown in the following display. This section of the report is a graphic representation and a quick reference version of the Distribution Section. Distribution Section percentages are shown for each range in the form of a bar chart.
Comparative SR8 Index Report

Contents
- Comparative SR8 Index Report Fields (see page 197)
- Table Entry Distribution Section Fields (see page 199)
- Histogram Section Fields (see page 200)

A Comparative SR8 Index Report is used to assess changes that have occurred in the physical organization of an SR8 index or portion of an SR8 index. Consequently, this type of report uses data from two different statistical files. The data from these files describes the SR8 index set at two different times.

The Comparative SR8 Index Report is divided into three sections: the Index Statistics Section, the Table Entry Distribution Section, and the Histogram Section.

The format of each section is the same as that of the corresponding section in an SR8 Index Report except that fields have been added to show old and new statistical file values and the changes between these values.

Comparative SR8 Index Report Fields

Comparative Index Report Fields are shown in the following screen displays. Identification and SR8 Index Statistic Section fields of the Comparative SR8 Index Report are described below. See SR8 Index Report (see page 192) for descriptions of fields identical to those found in the Comparative SR8 Index Report.

```plaintext
+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+
| CA-TOOLS | DATABASE | RELEASE | CA-PRAT | DATE | TIME | PAGE |
+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+----------+
| nn        | mm:ss    | Rnn,     | SR8 INDEX REPORT | dd/mm/yy | hh:mm:ss |
| nn        | nnnn     |          |              |        |       | |
|           |          |          | SR8 INDEX STATISTICS SECTION |          | |
| COMPARATIVE | IX-STUD-LNAME |                | | |
| NEW FILE CREATION DATA: mm/dd/yy - hh:mm:ss | |
| OLD FILE CREATION DATA: mm/dd/yy - hh:mm:ss | |
```
### ITEM

<table>
<thead>
<tr>
<th>ITEM</th>
<th>NEW</th>
<th>OLD</th>
<th>VARIANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER RECORD NAME</td>
<td>SR7</td>
<td>SR7</td>
<td></td>
</tr>
<tr>
<td>MEMBER RECORD NAME</td>
<td>STUDENT</td>
<td>STUDENT</td>
<td></td>
</tr>
<tr>
<td>MEMBERSHIP OPTION</td>
<td>MANDATORY AUTO</td>
<td>MANDATORY AUTO</td>
<td></td>
</tr>
<tr>
<td>DUPLICATES OPTION</td>
<td>DUPS LAST</td>
<td>DUPS NO</td>
<td></td>
</tr>
<tr>
<td>SEQUENCE OPTION</td>
<td>DESCENDING SEQ</td>
<td>DESCENDING SEQ</td>
<td></td>
</tr>
<tr>
<td>COMPRESSION OPTION</td>
<td>NO</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>SET ORDER OPTION</td>
<td>SORTED</td>
<td>SORTED</td>
<td></td>
</tr>
<tr>
<td>SORT TYPE</td>
<td>SORTED DBKEY</td>
<td>SORTED DBKEY</td>
<td></td>
</tr>
<tr>
<td>MAX NUMBER OF TABLE ENTRIES</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>MAX LENGTH OF TABLE ENTRY</td>
<td>8</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>SYMBOLIC KEY LENGTH</td>
<td>4</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>MAX SR8 RECORD SIZE</td>
<td>72</td>
<td>72</td>
<td>0</td>
</tr>
<tr>
<td>TOTAL BYTES TO STORE INDEX</td>
<td>588</td>
<td>2,060</td>
<td>-1,472</td>
</tr>
<tr>
<td>PAGE NUMBER - 1ST SR8</td>
<td>92,129</td>
<td>92,129</td>
<td>0</td>
</tr>
<tr>
<td>NUMBER OF BOTTOM SR8 IN TOP</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SR8 DISPLACEMENT</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>NUMBER OF LEVELS</td>
<td>3</td>
<td>5</td>
<td>-2</td>
</tr>
<tr>
<td>NUMBER OF ORPHANS</td>
<td>24</td>
<td>124</td>
<td>0</td>
</tr>
</tbody>
</table>

### UPPER LEVEL STATISTICS

<table>
<thead>
<tr>
<th>ITEM</th>
<th>NEW</th>
<th>OLD</th>
<th>VARIANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR8 RECORDS</td>
<td>4</td>
<td>16</td>
<td>-12</td>
</tr>
<tr>
<td>PAGE SPREAD</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>USED TABLE ENTRIES</td>
<td>10</td>
<td>37</td>
<td>-27</td>
</tr>
<tr>
<td>AVG</td>
<td>2.50</td>
<td>2.31</td>
<td>+0.19</td>
</tr>
<tr>
<td>MAX</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>MIN</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>ORPHAN COUNT</td>
<td>8</td>
<td>26</td>
<td>0</td>
</tr>
<tr>
<td>AVG</td>
<td>2.00</td>
<td>1.63</td>
<td>+0.38</td>
</tr>
<tr>
<td>MAX</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>MIN</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

### BOTTOM LEVEL STATISTICS

<table>
<thead>
<tr>
<th>ITEM</th>
<th>NEW</th>
<th>OLD</th>
<th>VARIANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR8 RECORDS</td>
<td>7</td>
<td>22</td>
<td>-15</td>
</tr>
<tr>
<td>PAGE SPREAD</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>USED TABLE ENTRIES</td>
<td>16</td>
<td>51</td>
<td>-35</td>
</tr>
<tr>
<td>AVG</td>
<td>2.29</td>
<td>2.32</td>
<td>-0.03</td>
</tr>
<tr>
<td>MAX</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>MIN</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>ORPHAN COUNT</td>
<td>16</td>
<td>98</td>
<td>0</td>
</tr>
<tr>
<td>AVG</td>
<td>2.29</td>
<td>4.45</td>
<td>-2.17</td>
</tr>
<tr>
<td>MAX</td>
<td>6</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>MIN</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

1 **INDEX NAME**--The name of the SR8 index set for the reported statistics.

2 **NEW FILE CREATION DATA**--The date and time of creation of the statistical file which is listed in the z/OS and OS/390 JCL as STAT1, and in the VSE/ESA JCL as SYS010.

3 **OLD FILE CREATION DATA**--The date and time of creation of the statistical file which is listed in the z/OS and OS/390 JCL as STAT2, and in the VSE/ESA JCL as SYS011.

4 **NEW**--The values listed under this heading are taken from the new statistics file (item 2). See SR8 Index Report for a description of each field listed.

5 **OLD**--The values listed under this heading are taken from the old statistical file (item 3). See SR8 Index Report for a description of each field listed.

6 **VARIANCE**--Listed under this heading are the differences between new and old raw counts. Positive values indicate the amount by which new statistical file values are greater than old. Negative values indicate the amount by which new file values are less than old file values.
**Table Entry Distribution Section Fields**

The Distribution Section of the Comparative SR8 Index Report is shown in the following display. See the SR8 Index Report for descriptions of fields not discussed here.

<table>
<thead>
<tr>
<th>CA-TOOLS</th>
<th>RELEASE</th>
<th>CA IDMS</th>
<th>DATE</th>
<th>TIME</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>/DB ANALYZER</td>
<td>Rnn.</td>
<td>SR8 INDEX REPORT</td>
<td>dd/mm/yy</td>
<td>hh:mm:ss</td>
<td>nnn</td>
</tr>
</tbody>
</table>

**INDEX NAME:** IX-STUD-LNAME

**NEW FILE CREATION DATA:** mm/dd/yy - hh:mm:ss

**OLD FILE CREATION DATA:** mm/dd/yy - hh:mm:ss

<table>
<thead>
<tr>
<th>ITEM</th>
<th>RANGE</th>
<th>NEW</th>
<th>PCT</th>
<th>OLD</th>
<th>PCT</th>
<th>VARIANCE</th>
<th>PCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>VARIANCE</td>
<td>PCT</td>
<td>TABLE</td>
<td>ENTRIES</td>
<td># OF OCCURRENCES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-----</td>
<td>-------</td>
<td>----------</td>
<td>--------------</td>
<td>---</td>
<td>---------</td>
<td>---</td>
</tr>
<tr>
<td>TABLE ENTRY DISTRIBUTION SECTION COMPARATIVE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UPPER LVL TBL ENTRIES USED</th>
<th>TABLE ENTRIES</th>
<th># OF OCCURRENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>0 0</td>
<td>1 1</td>
<td>0 0</td>
</tr>
<tr>
<td>-9 19</td>
<td>2 2</td>
<td>50 69</td>
</tr>
<tr>
<td>-3 19</td>
<td>3 3</td>
<td>50 31</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BOTTOM LVL TBL ENTRIES USED</th>
<th>TABLE ENTRIES</th>
<th># OF OCCURRENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>0 0</td>
<td>1 1</td>
<td>0 0</td>
</tr>
<tr>
<td>-10 3</td>
<td>2 2</td>
<td>5 71</td>
</tr>
<tr>
<td>-5 3</td>
<td>3 3</td>
<td>2 29</td>
</tr>
</tbody>
</table>

1 NEW PCT—The values listed under this heading are raw counts and percentages for each range. These values are taken from the new statistics file. See Comparative SR8 Index Report Identification Fields, item 2.

2 OLD PCT—The values listed under this heading are raw counts and percentages for each range. These values are taken from the old statistics file. See Comparative SR8 Index Report Identification Fields, item 3.

3 VARIANCE PCT—Listed under this heading are the differences between new and old raw counts and the percent variance between the old and new values. Positive values indicate the amount by which new statistical file values are greater than old. Negative values indicate the amount by which new file values are less than old file values.

For variance to be meaningful, both the old and new statistical files must contain identical ranges. If they do not, no values are listed under this heading.
Histogram Section Fields

A portion of the Histogram Section of the Comparative SR8 Index Report is shown in the following display. This section of the report is a graphic representation and a quick reference version of the Distribution Section. Old (O) and new (N) values of Distribution Section percentages for each range are shown on two lines in the form of a bar chart.
4.4.4.4.4.4.4.4.4.

0---++---10---++---20---++---30---++---40---++---50---++---60---
++---70---++---80---++---90---++---100
The following table lists the reports that document these statistics in order by report module number:

<table>
<thead>
<tr>
<th>SREPORT Module</th>
<th>SREPORT Category</th>
<th>Statistics Report Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td></td>
<td>Startup Records Read (required with remaining modules)</td>
</tr>
<tr>
<td>001</td>
<td>Histogram</td>
<td>IDMS Statistics Histogram Report (system and task)</td>
</tr>
<tr>
<td>003</td>
<td>System</td>
<td>IDMS DC System Statistics</td>
</tr>
<tr>
<td>004</td>
<td>Task</td>
<td>IDMS DC Task Statistics by IP Address</td>
</tr>
<tr>
<td>005</td>
<td>Task</td>
<td>IDMS DC Task Statistics by User Id</td>
</tr>
<tr>
<td>006</td>
<td>Task</td>
<td>IDMS DC Task Statistics by Lterm Id</td>
</tr>
<tr>
<td>007</td>
<td>Task</td>
<td>IDMS DC Task Statistics by Task Code</td>
</tr>
<tr>
<td>008</td>
<td>Task</td>
<td>IDMS DC ERUS Task Statistics by Accounting Data</td>
</tr>
<tr>
<td>009</td>
<td>Task</td>
<td>IDMS DC ERUS Task Statistics by Program Name</td>
</tr>
<tr>
<td>010</td>
<td>Transaction</td>
<td>IDMS DC Transaction Statistics by User Id</td>
</tr>
<tr>
<td>011</td>
<td>Transaction</td>
<td>IDMS DC Transaction Statistics by Lterm Id</td>
</tr>
<tr>
<td>012</td>
<td>System</td>
<td>IDMS DC Task Summary</td>
</tr>
<tr>
<td>013</td>
<td>System</td>
<td>IDMS DC Program Summary</td>
</tr>
<tr>
<td>014</td>
<td>System</td>
<td>IDMS DC Queue Summary</td>
</tr>
<tr>
<td>015</td>
<td>System</td>
<td>IDMS DC Line Summary</td>
</tr>
<tr>
<td>016</td>
<td>System</td>
<td>IDMS DC Physical Terminal Summary</td>
</tr>
<tr>
<td>017</td>
<td>Record</td>
<td>Summary of Records Read</td>
</tr>
<tr>
<td>018</td>
<td>CA ADS</td>
<td>ADS OnLine Statistics by User Id</td>
</tr>
<tr>
<td>019</td>
<td>CA ADS</td>
<td>ADS OnLine Statistics by Dialog and Version Number</td>
</tr>
<tr>
<td>020</td>
<td>CA ADS</td>
<td>ADS OnLine Statistics by Logical Terminal Id</td>
</tr>
<tr>
<td>021</td>
<td>Transaction</td>
<td>IDMS DC Transaction Statistics by Dialog</td>
</tr>
<tr>
<td>022</td>
<td>Transaction</td>
<td>IDMS DC Transaction Statistics by IP Address</td>
</tr>
<tr>
<td>099</td>
<td></td>
<td>No listing (creates an output file of archive statistics records)</td>
</tr>
</tbody>
</table>

Uses for Statistics Reports

Systems administrators use statistics reports to monitor system activity. Statistics reports can be used to:

- Assist in system tuning and maintenance
- Evaluate processing efficiency at the system level, task level, transaction level, and dialog level
Monitor system activity within certain time intervals

Other Tools Available

Other tools available to monitor system activity are the PRINT LOG utility, PLOG, and DCMT STATISTICS commands. These tools are discussed in more detail in Other CA IDMS Reporting Facilities (see page 659).

What Follows

This section discusses types of statistics records, input parameters needed to process a statistics report, and sample output.

Note: For more information on statistics, see the Administering section.

Types of Statistics Records

A statistics record is one of the DC/UCF system log record types. A statistics record is variable in length and contains the following type of information:

- Information such as the date and time.
- Statistical data for the 35 types of statistics records. For a list of the statistics records types, see Record Summary Statistics Report (see page 252).

The following table lists the DSECTs that define the layout of the statistics records. Each of these DSECTs is presented in DSECT Reference (https://docops.ca.com/display/IDMSCU/DSECT+Reference).

<table>
<thead>
<tr>
<th>DSECT</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#LGR</td>
<td>The log record header.</td>
</tr>
<tr>
<td>DS</td>
<td></td>
</tr>
<tr>
<td>#STLD</td>
<td>The statistics log record header. This DSECT redefines 256 bytes of the log record header starting at offset X '14' and contains a description of 35 statistics record sub-types.</td>
</tr>
<tr>
<td>S</td>
<td></td>
</tr>
<tr>
<td>#STRD</td>
<td>The statistics record. This DSECT describes the layout of the statistics in the log record.</td>
</tr>
<tr>
<td>S</td>
<td>It contains a field that indicates the release level of the DC/UCF system for the statistics log record. To determine the release level, check the STRRID field in #STRDS. For example, STRRID=C'Rnnn', where nnn is the release level.</td>
</tr>
<tr>
<td>#HST</td>
<td>The histogram record. This DSECT describes the layout of histogram data in the log record.</td>
</tr>
<tr>
<td>DS</td>
<td>Histogram log records are variable length. #HSTDS redefines 244 bytes of the statistics log record header starting at offset X '0C'.</td>
</tr>
</tbody>
</table>
Layout of Statistics Log Records

For all statistics log records, the first two DSECTs are #LGRDS and #STLDS. These two DSECTs are followed by one or more occurrences of #STRDS, which varies depending on the type of statistics being saved.

The following figure illustrates the order of DSECTs for most statistics log records:

The order of DSECTs for task and transaction, system, and histogram statistics are described in the sections that follow.

Task or Transaction Statistics

There are five #STRDS DSECT types, in the following order, for task or transaction statistics:

- **STRDC**
  The first DSECT. Describes the layout of DC statistics.

- **STRTDB**
  The second DSECT. Describes the layout of database statistics.

- **STRTHDR**
  The third DSECT. Describes header information.

- **STRTSQL**
  The fourth DSECT. Describes the layout of SQL statistics.

- **STRDCX**
  The fifth DSECT. Describes the layout of the DC extended statistics.

To determine whether the current log record is for task or transaction statistics, check the STLTYPE field in #STLDS:
Order of DSECTs for Task and Transaction Statistics

The following figure illustrates the order of DSECTs for task and transaction statistics log records:

System Statistics

There are five #STRDS DSECT types, in the following order, for system statistics:

- **STRTSYS**
  The first DSECT. Describes the layout of system statistics.

- **STRTDC**
  The second DSECT. Describes the layout of IDMS-DC statistics.

- **STRTHDB**
  The third DSECT. Describes the layout of database statistics.

- **STRTSQL**
  The fourth DSECT. Describes the layout of SQL statistics.

- **STRTDXC**
  The fifth DSECT. Describes the layouts of the DC extended statistics.

To determine whether the current log record is for system statistics, check the STLTYPE field in #STLDS. For system statistics, STLTYPE=STLTCST.

Order of DSECTs for System Statistics

The following figure illustrates the order of DSECTs for system statistics log records:
Histograms

There are two #STRDS DSECT types, in the following order, for histograms:

- STRTHIS
- HTRDS

To determine whether the current log record is for a histogram, follow these steps:

1. Check the STLTYPE field in #STLDS. It should be either STLTCST or STLTPLE.
2. Check the STRTYPE field in #STRDS. It should be STRTHIS.

Order of DSECTs for Histograms

The following figure illustrates the order of DSECTs for histogram log records:

Number of Statistics Log Records

A statistics block can contain more data than can be written in one log record. When this occurs, DC/UCF uses up to three log records to write the statistics to the log.

To determine whether a particular statistics log record contains only part of the data from a statistics block, check the LGRTYPE field in #LGRDS:

- If LGRTYPE=X'76', this log record is one of the following:
  - The first of two log records used to write one statistics block
The first of three log records used to write one statistics block

The second of three log records used to write one statistics block

*If LGRTYPE=*F6*, this log record is one of the following:

- The only log record used for that particular statistics block
- The second of two log records used to write one statistics block
- The third of three log records used to write one statistics block

**Note:** Statistics log records have a maximum length of 276 bytes. Programs that read statistics from the DC/UCF log should reserve twice that amount of space to handle statistics that span two log records.

---

**SREPORTs for Statistics Records**

**SREPORT 099**

Statistics records output by SREPORT 099 are 280-byte fixed-length records with the same layout as records written to the archived system log file. CA Culprit for CA IDMS copies the RDW associated with the variable-length records to bytes 1 through 4 of the fixed-length records.

**SREPORT 000**

SREPORT 000 contains the CA Culprit for CA IDMS REC parameters that define the fields in each type of statistics record. Each REC parameter defines the start position, length, and data type of a field within the record. The following considerations apply:

- The field names assigned in the CA Culprit for CA IDMS report are not always the same as those assigned in the DSECTS.
- Each field defined by a CA Culprit for CA IDMS REC parameter has a start position 5 bytes more than in the DSECT.

---

**Producing Statistics Reports**

**Contents**

- Syntax (see page 208)
- Parameters (see page 208)
- Examples (see page 210)
- Operating System Considerations (see page 211)
SYSTEM Statements Determine How to Log Statistics

DC/UCF can log system resource statistics to the database, to a single file, or to alternate files as
defined at system configuration by the LOG clause of the SYSTEM statement. When statistics are
logged to the database, they are stored in DDLDCLOG, the log area of the data dictionary.

Reports Document Statistics Logged to Database

Statistics reports document statistics logged to the database. The ARCHIVE LOG utility offloads the
statistics from the DDLDCLOG area to an archived system log file; for more information on this utility,
see the CA IDMS Administrating section.

Input to Statistics Reports

Input to the statistics reports is either the archived system log file or a file created by executing
SREPORT 099. SREPORT 099 reads the archived system log file and creates a file of archived statistics
records.

Syntax

Syntax for the CA Culprit for CA IDMS parameters is shown followed by examples of CA Culprit for CA
IDMS code. Coding is freeform except that each parameter must be coded starting in column 2.

Parameters

Syntax rules appear in Section 1, "Introduction," except as described as follows:
**Input**
Specifies the CA Culprit for CA IDMS parameter that designates the physical characteristics of the input file.

**280**
Specifies the record length, in bytes.

**V block-size**
Specifies the record type (variable length) and the block size if the archived system log file is used as input:
- In z/OS systems, the block size is informational
- In z/VSE systems, block-size-n must match the actual block size of the archived log file (the default block size of the archive file is 6000).

> **Note:** For more information on creating output files with CA IDMS utilities in a z/VSE environment, see the *CA IDMS Administrating section*.

*Block-size* must be greater than or equal to the actual block size. (The block size for the archived log file is specified within the JCL used by the ARCHIVE LOG utility to create the file.)

**F 8120**
Specifies the record type (fixed length) and the block size (8120) if the output file produced by SREPORT 099 is used as input.

> **Note:** In z/OS systems, the block size can be omitted.

**USE 'SREPORT 000'**
Requests SREPORT 000; SREPORT 000 contains CA Culprit for CA IDMS REC parameters that define fields used in other statistics report modules. This parameter must be included in all statistics report runs.

**begin-date**
Specifies the begin date for the period to be covered by the requested reports. If no time period is specified, the requested reports will cover the entire period represented by the input archive file or by the input file from SREPORT 099.

> **Note:** A time period and a nonzero session indicator are required for SREPORTs 001, 003, and 012 through 016.

*Begin-date* must be in the Julian form *yyddd*, where *yy* is the last two digits of the year and *ddd* is the day.
• **begin-time**
  Specifies the begin time for the period to be covered by the requested reports. *Begin-time* must be in the form `hhmm`, where *hh* is hours based on a 24-hour clock, and *mm* is minutes.

• **end-date**
  Specifies the end date for the period to be covered by the requested reports. *End-date* must be in the Julian form `yyddd`, where *yy* is the last two digits of the year and *ddd* is the day.

• **end-time**
  Specifies the end time for the period to be covered by the requested reports. *End-time* must be in the form `hhmm`, where *hh* is hours based on a 24-hour clock, and *mm* is minutes.

• **session-indicator-number**
  Specifies that the requested reports are to cover the indicated occurrence of a DC/UCF session within the specified time period. A DC/UCF session is the period of time from one system startup to the next; the first session within a specified time period begins with the first startup record. SREPORT 000 lists each startup record that exists in the archive file and when the record was logged.
  *Session-indicator-number* must be an integer in the range 0 through 9999. If *session-indicator-number* is not specified or is equal to zero, the requested reports will cover the entire period specified.

  Note: A time period and a nonzero session indicator are required for SREPORTs 001, 003, and 012 through 016.

• **SELECT/BYPASS BUFFER WHEN boolean-expression**
  Specifies optional selection criteria to be applied during the extract phase of processing. If a SELECT or BYPASS parameter is coded, it must follow the USE 'SREPORT 000' parameter.

• **USE 'SREPORT sreport-number'**
  Specifies a report module name. *Sreport-number* must be a report number, as specified in the module name. Multiple USE parameters can be included; however, each one must be specified on a separate line.

### Examples

**Example 1**

SREPORT 003 is requested for the first DC/UCF session occurring between 8:30 and 9:50 PM on March 5, 1999.

```
DATABASE DICTNAME=CULPDICT
INPUT 280 F 8120
USE 'SREPORT 000' (99064 2030 99064 2150 1)
USE 'SREPORT 003'
```

As shown in the output generated for SREPORT 000, the first session within this time frame begins at 20:32 PM with startup record number 5703.
Example 2

SREPORT 012 is requested for the second DC/UCF session occurring in the period beginning at 6:00 AM on 10/19/99 and ending at 6:00 PM on 10/19/99. Input consists of the archived system log file created by means of the ARCHIVE LOG utility. Input parameters will not be listed.

PARAM=NOLIST
IN 280 V
USE ‘SREPORT 000’ (99292 0600 99292 1800 2)
USE ‘SREPORT 012’

Example 3

SREPORTs 010 and 005 are requested for all DC/UCF sessions in the period beginning at 8:00 AM on 11/22/99 and ending at 4:00 PM on 11/22/99. Input consists of the output file created by a previous run of SREPORT 099; PS indicates a sequential file. By default, input parameters will be listed.

IN 280 F 8120 PS
USE ‘SREPORT 000’ (99326 0800 99326 1600)
USE ‘SREPORT 010’
USE ‘SREPORT 005’

Example 4

Statistics report modules SREPORT 000 and SREPORT 012 are stored in the CULPDICT data dictionary. The SELECT parameter selects only those records stored in the input buffer that specify user ids TAF and TNC. The code for SREPORT 012 identifies TSTUSID as the user-id field name.

DATABASE DICTNAME=CULPDICT
IN 280 V
USE ‘SREPORT 000’
SELECT BUFFER WHEN TSTUSID EQ (‘TAF’ ‘TNC’)
USE ‘SREPORT 012’

Operating System Considerations

JCL Coding Considerations

Statistics reports can run either in local mode or under the central version. The JCL to run statistics reports appears in Appendixes A through D for z/OS, z/VSE and z/VM operating systems, respectively. The following considerations apply to coding the JCL for running statistics reports:

- The input file containing the statistics records must be defined with ddname/filename/linkname SYS010 (the input file can be either the archived DC/UCF system log file or the output file produced by SREPORT 099).

- When SREPORT 099 is run, the output file must be defined with ddname/filename/linkname SYS020.
Input File JCL Modifications

For the input file, modify the JCL in Appendices A through D, as follows:

- **For z/OS operations systems**, modify ddname SYS010 in Appendix A as follows:

  ```
  //SYS010 DD DSN=user.inputfil,DISP=OLD,UNIT=tape,VOL=SER=nnnnnn
  ```

  - *user.inputfil*: data set name of the archived system log file or SREPORT 099 output file
  - *tape*: symbolic device name of a disk or tape input file
  - *nnnnnn*: volume serial number of the input file

- **For z/VSE operating systems**, modify filename SYS010 in Appendix B as follows:

  ```
  // TLBL SYS010,'user.inputfil',,,nnnnnn,,f
  // ASSGN SYS010,X'ttt'
  ```

  - *user.inputfil*: file-id of the archived system log file or SREPORT 099 output file
  - *nnnnnn*: volume serial number of the input file
  - *f*: file number of the input file
  - *ttt*: device assignment (channel and unit) for tape files (files may be disk instead of tape, in which case a device assignment, DLBL and EXTENT information are also required)

- **For z/VM and z/VM operating systems**, modify the SYS010 command in Appendix C, as follows:

  ```
  For input from the archived system log file:
  ```

  - For tape files: `FILEDEF SYS010 TAP1 SL VOLID nnnnnn
    (RECFM VB LRECL 280 BLKSIZE bbbb
  ```

  - For disk files: `FILEDEF SYS010 DISK input file a
    (RECFM VB LRECL 280 BLKSIZE bbbb
  ```

  - *nnnnnn*: volume serial number of the archived system log file
  - *bbbbb*: block size of the input file
  - *input file a*: filename, filetype, filemode of archived system log file

  ```
  For input from the file created by SREPORT 099:
  ```

  ```
  FILEDEF SYS010 DISK input file a
    (RECFM FB LRECL 280 BLKSIZE 8120
  ```

  - *input file a*: filename, filetype, filemode of archived system log file

SREPORT 099 Output File JCL Modifications

To create a file of statistics records using SREPORT 099, modify the JCL in Appendices A through D, as follows:
For z/OS operations systems, modify ddname SYS020 in Appendix A, as follows:

```plaintext
//SYS020 DD DSN=user.nonprint,DISP=(NEW,CATLG),
      SPACE=(TRK,(10,10)),UNIT=tape,VOL=SER=nnnnnn
      DCB=(DSORG=PS,RECFM=FB,LRECL=280,BLSIZE=8120)
```

<table>
<thead>
<tr>
<th>user.nonprint</th>
<th>data set name for nonprint output</th>
</tr>
</thead>
<tbody>
<tr>
<td>tape</td>
<td>symbolic device name of the nonprint output file</td>
</tr>
<tr>
<td>nnnnnn</td>
<td>volume serial number of the nonprint output file</td>
</tr>
</tbody>
</table>

For z/VSE operating systems modify file name SYS020 in Appendix B as follows:

```plaintext
// ASSGN SYS020,X'ttt'
// TLBL SYS020,'user.nonprint',15
```

<table>
<thead>
<tr>
<th>ttt</th>
<th>device assignment (channel and unit) for tape files (files may be disk instead of tape, in which case a device assignment, DLBL, and EXTENT information are also required.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>user. nonprint</td>
<td>file-id and retention period for nonprint/nonpunch output</td>
</tr>
<tr>
<td>,15</td>
<td>15</td>
</tr>
</tbody>
</table>

For z/VM and z/VM operating systems, modify the SYS020 command in Appendix C as follows:

```plaintext
FILEDEF SYS020 nonprint file a (RECFM FB LRECL 280 BLKSIZE 8120)
```

| nonprint file a | filename, filetype, filemode of nonprint output |

---

**DC/UCF System Statistics Reports**

**Contents**

- SREPORT 003 - IDMS DC System Statistics (see page 214)
- SREPORT 012 - IDMS DC Task Summary (see page 224)
- SREPORT 013 - IDMS DC Program Summary (see page 225)
- SREPORT 014 - IDMS DC Queue Summary (see page 227)
- SREPORT 015 - IDMS DC Line Summary (see page 228)
- SREPORT 016 - IDMS DC Physical Terminal Summary (see page 230)

**Record Systemwide Data**

System statistics record systemwide data. DC/UCF always collects system statistics (they are not optional) because they require minimal overhead and provide valuable information for tuning and maintaining the DC/UCF system.

**When Statistics Are Logged**

System statistics are logged to the DC/UCF log file at the following times:

- At normal system shutdown
At the statistics interval established at system generation by the STATISTICS parameter of the SYSTEM statement. The statistics interval can be varied at run time by means of the DCMT VARY STATISTICS command.

Upon explicit request by means of a DCMT WRITE STATISTICS command.

Summary of System Statistics Reports

System statistics are collected and written for six categories:

<table>
<thead>
<tr>
<th>Category</th>
<th>Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>General systemwide statistics</td>
<td>SREPORT 003</td>
</tr>
<tr>
<td>Systemwide task statistics</td>
<td>SREPORT 012</td>
</tr>
<tr>
<td>Systemwide program statistics</td>
<td>SREPORT 013</td>
</tr>
<tr>
<td>Systemwide queue statistics</td>
<td>SREPORT 014</td>
</tr>
<tr>
<td>Systemwide line statistics</td>
<td>SREPORT 015</td>
</tr>
<tr>
<td>Systemwide physical terminal (including UCF PTERMs) statistics</td>
<td>SREPORT 016</td>
</tr>
</tbody>
</table>

SREPORT 003 - IDMS DC System Statistics

Contents

SREPORT 003, the IDMS DC System Statistics report, summarizes all systemwide statistics and thus provides an overview of system performance.

Sample SREPORT 003:

```
REPORT NO. 03                              IDMS-DC SYSTEM STATISTICS Rnn.n mm/dd/yy PAGE 1
SELECTED FROM: yyddd hh:mm                 TO: yyddd hh:mm
CV SYSTEM START: yyddd hh:mm               TO: yyddd hh:mm
CV JOB NAME: jobname                        CV NUMBER: nnn

SYSTEM STATISTICS
116 TOTAL TASKS                             0 STD PGMPOOL LOADS
72 TOTAL SYSTEM TASKS                       0 STD PGMPOOL WAITS
0 TASKS ABENDED                            0 STD PGM PAGES LOADED
0 RUNAWAY TASKS ABORTED                     29 RENTPOOL LOADS
0 TIMES AT MAX TASK                         0 RENTPOOL WAITS
0 SHORT ON STORAGE                         842 RENT PGM PGS LOADED
0 OVER RLE THRESH                          0 XA PGMPOOL LOADS
0 OVER RCE THRESH                          0 XA PGMPOOL WAITS
0 OVER DPE THRESH                          0 XA PGM PGS LOADED
0 OVER ILE THRESH                          169 XA RENTPOOL LOADS
0 STORAGE POOL WAITS                       0 XA RENTPOOL WAITS
1,624 STG REQS - PASS 1                    24,987 XA RENT PGS LOADED
1,103 STG REQS - PASS 2                    0 PAGE RELEASE RQSTS
0 PUT JOURNALS                             0 PAGES RELEASED
0 SET TIME WAITS                           0 PAGE FIX RQSTS
236 SET TIME POSTS                        0 PAGES PFIXED
1 SET TIME STRTASKS                        0 PAGE FREE RQSTS
234 SET TIME CANCELS                       0 PAGES PGFREED
0 AUTOSTART TASKS                          0

DC STATISTICS
19.8805 USER MODE CPU TIME                  3.8358 SYSTEM MODE CPU TIME
659,356 DC SERVICE REQUESTS                64 GET SCRATCHES
659,928 DB SERVICE REQUESTS                106 PUT SCRATCHES
245 PROGRAMS CALLED                        64 DELETE SCRATCHES
375 MAX # RLE'S USED                       10 GET QUEUES
329 MAX # RCE'S USED                       0 PUT QUEUES
308 MAX # DPE'S USED                       0 DELETE QUEUES
1,337 STACK HI WATERMARK                   8 GET TIMES
```
Field Descriptions

A description of the fields in the IDMS DC System Statistics report follows:

- **SELECTED FROM/TO**
  Specifies the time period specified on the USE 'SREPORT 000' parameter where SELECTED FROM specifies the beginning date and time and SELECTED TO specifies the ending date and time. The date is in Julian form, yyddd, where yy is the last two digits of the year and ddd is the day. The time is in hh:mm form, where hh is hours based on a 24-hour clock and mm is minutes.

- **CV SYSTEM START/INTERVAL START/TO**
  Specifies the actual time period for the report generated. If the USE 'SREPORT 000' parameter does not specify a time range, CV SYSTEM START/INTERVAL START and TO represent the time range for the entire input archive file or input file created by SREPORT 099. When the input file contains cumulative statistics, the CV SYSTEM START date and time will be displayed. When the input file contains interval based statistics, the INTERVAL START date and time will be displayed.

- **CV JOB NAME**
  Specifies the starting job name of Central Version which collected the statistics.

- **CV NUMBER**
  Specifies the number of Central Version which collected the statistics.

- **TOTAL TASKS**
  Indicates the total number of system and user tasks, including external request units, executed for this session.

- **TOTAL SYSTEM TASKS**
  Indicates the number of system tasks active at the conclusion of this session. System-initiated tasks include FACTOTUM, MASTER, DBRC, print, and line drivers.

- **TASKS ABENDED**
  Indicates the number of tasks that abended during this session.
RUNAWAY TASKS ABORTED
Indicates the number of tasks that terminated abnormally because their execution time exceeded the maximum amount of time defined in the RUNAWAY INTERVAL clause of the SYSTEM statement.

TIMES AT MAX TASK
Indicates the number of times a new task could not start because the system was processing the maximum number of tasks specified at system generation in the MAXIMUM ERUS and MAXIMUM TASK clauses of the SYSTEM statement.
Interpretation: Generally, this value should be close to 0. Values equal to 0 may indicate that the overall system size is larger than necessary; values greater than 0 may represent peaks in system loads or chronic system overload.
To reduce the number of times at maximum tasks, make one or more of the following system adjustments:
- Increase the MAX ERUS and MAX TASKS thresholds
- Decrease the limit specified for the MAXIMUM CONCURRENT THREADS clause of the TASK statement at system generation

SHORT ON STORAGE
Indicates the number of times program storage requests were not satisfied. Generally, the value should be close to zero; values greater than zero should represent peaks in storage utilization, rather than chronic shortages.
Interpretation: Values consistently greater than zero indicate the size of the storage pool or storage cushion is too small. The size of the storage pool is defined at system generation with the STORAGE POOL and XA STORAGE POOL clauses of the SYSTEM system generation statement; the size of the cushion is defined with the CUSHION clause of the same statement.

OVER RLE THRESH
Indicates how many times the number of resource link elements (RLEs) exceeded the value allocated in the SYSTEM statement at system generation. When the threshold is reached, the task that is executing abends; if the task is a system task, the system abends. Ideally, this value should be 0.

OVER RCE THRESH
Indicates how many times the number of resource control elements (RCEs) exceeded the value allocated in the SYSTEM statement at system generation. When the threshold is reached, the task that is executing abends; if the task is a system task, the system abends. Ideally, this value should be 0.

OVER DPE THRESH
Indicates how many times the number of deadlock prevention elements (DPEs) exceeded the value allocated in the SYSTEM statement at system generation. When the threshold is reached, the task that is executing abends; if the task is a system task, the system abends. Ideally, this value should be 0.

OVER ILE THRESH
Indicates the number of time the internal lock elements (ILEs) exceeded their allocated value. When the threshold is reached, the task that is executing abends. If the task is a system task, the system abends. Ideally, this value should be zero (0).
• **STORAGE POOL WAITS**
  Indicates the number of times tasks had to wait for a sufficient amount of contiguous storage. This value should be as low as possible; if the value is large or increasing, increase the size of the storage cushion in the CUSHION clause of the STORAGE POOL statement at system generation.

• **STG REQS - PASS 1**
  Indicates the number of storage requests that allocated space on a previously allocated page.

• **STG REQS - PASS 2**
  Indicates the number of storage requests that allocated space on a previously allocated page and on a contiguous new page.

• **SET TIME WAITS**
  Indicates the number of program requests to place a task in a wait state. The program issues the request with a SET TIMER WAIT DML statement.

• **SET TIME POSTS**
  Indicates the number of program requests to post a user-specified event control block (ECB) after the specified time interval elapses. The program issues the request with a SET TIMER POST DML statement.

• **SET TIME STRTTASKS**
  Indicates the number of program requests to initiate a user-specified task after the specified time interval elapses. The program issues the request with a SET TIMER START DML statement.

• **SET TIME CANCELS**
  Indicates the number of program requests to cancel the effect of a previously issued SET TIMER request. The program issues the request with a SET TIMER CANCEL DML statement.

• **AUTOSTART TASKS**
  (DC/UCF only) Indicates the number of times the associated task of each queue was invoked to process queue records. The task is invoked each time the queue threshold is exceeded. The threshold is defined with the THRESHOLD clause of the QUEUE system generation statement; the task is identified by the INVOKES TASK clause of the same statement.

• **STD PGMPOOL LOADS**
  Indicates the number of nonresident modules loaded into the 24-bit program pool; nonresident modules include programs, CA ADS and CA ADS batch dialogs, maps (DC/UCF only), and tables.

• **STD PGMPOOL WAITS**
  Indicates the number of times program loads were delayed due to insufficient space in the 24-bit program pool. Ideally, this value should be zero; a value greater than zero indicates insufficient space in the program pool for the volume of program load activity in the system.
  Interpretation: If this value is high, the size of the 24-bit program pool should be increased or nonresident programs should be redefined as either resident or reentrant. Program pool usage can be observed dynamically by means of the OPER WATCH PR command. For more information on program pools, see the Administrating section.

• **STD PGM PAGES LOADED**
  Indicates the number of pages used by programs loaded into the 24-bit program pool. The size of a page equals 4K. Only one program can occupy a program pool page; that is, a 4.5K program uses two pages of program pool.
- **RENTPOOL LOADS**
  Indicates the number of reentrant programs loaded into the 24-bit reentrant pool; reentrant programs include CA ADS dialogs and subschemas.

- **RENTPOOL WAITS**
  Indicates the number of times reentrant program loads were delayed due to insufficient space in the 24-bit reentrant pool. Ideally, this value should be zero; a value greater than zero indicates the size of the reentrant pool should be increased.

- **RENT PGM PGS LOADED**
  Indicates the number of pages used by reentrant programs loaded into the 24-bit reentrant pool. A page is 512 bytes (0.5K).

- **XA PGMPool LOADS**
  (Extended addressing only) Indicates the number of nonresident modules loaded into the 31-bit XA program pool; nonresident modules include programs, subschemas, maps, database procedures, and tables that have been assigned an RMODE of ANY.

- **XA PGMPool WAITS**
  (Extended addressing only) Indicates the number of times program loads were delayed due to insufficient space in the 31-bit XA program pool. Ideally, this value should be zero; a value greater than zero indicates insufficient space in the XA program pool for the volume of program load activity in the system.

- **XA PGM PGS LOADED**
  (Extended addressing only) Indicates the number of pages used by nonresident modules loaded into the 31-bit XA program pool. The size of a page equals 4K. Only one program can occupy a program pool page; that is, a 4.5K program uses two pages of program pool.

- **XA RENTPOOL LOADS**
  (Extended addressing only) Indicates the number of reentrant modules loaded into the 31-bit XA reentrant pool; reentrant modules include reentrant programs, subschemas, CA ADS dialogs, and DC/UCF maps that have been assigned an RMODE of ANY.

- **XA RENTPOOL WAITS**
  (Extended addressing only) Indicates the number of times program loads were delayed due to insufficient space in the 31-bit XA reentrant pool. Ideally, this value should be zero; a value greater than zero indicates insufficient space in the XA reentrant pool for the volume of program load activity in the system.

- **XA RENT PGS LOADED**
  (Extended addressing only) Indicates the number of pages used by reentrant modules loaded into the 31-bit XA reentrant pool. The size of a page equals 512 bytes (0.5K).

- **PAGE RELEASE RQSTS**
  Indicates the number of requests to release 4K virtual pages.

- **PAGES RELEASED**
  Indicates the number of 4K byte pages actually released.

- **PAGE FIX RQSTS**
  (VS systems only) Indicates the number of requests to fix 4K virtual pages allocated to storage pools defined to the system.
- **PAGES PFIXED**
  (VS systems only) Indicates the number of 4K virtual pages fixed in storage pools defined to the system.

- **PAGE FREE RQSTS**
  Indicates the number of requests to release 4K fixed virtual pages allocated to storage pools defined to the system, thereby making storage eligible for paging out.

- **PAGES PGFREED**
  Indicates the number released of 4K fixed virtual pages allocated to system storage pools.

- **PUT JOURNALS**
  Indicates the number of program requests to write statistics to the journal file with the WRITE JOURNAL DML statement.

- **USER MODE CPU TIME**
  Indicates the amount of CPU time (in ten-thousandths seconds) spent in executing user code.

- **DC SERVICE REQUESTS**
  Indicates the number of times user programs requested DC/UCF services (for example, GET STORAGE requests). This value also includes both explicit and implicit requests for database services. For example, OBTAIN record-name is an explicit program request. PUT QUEUE is an implicit request for database services because the DC/UCF system must store the queue record.

- **DB SERVICE REQUESTS**
  Indicates the number of times a user or system program requests database services (for example, OBTAIN record-name).
  For LRF and SQL programs, DB SERVICE REQUESTS should be less than or equal to the number of CALLS TO DBMS. You can use these values to evaluate how efficiently the LRF or SQL path extracts data. For example, a program OBTAIN logical-record command increments DB SERVICE REQUESTS by one, but may greatly increment the CALLS TO DBMS value, especially if an area sweep occurs due to the NULL SELECT clause.

- **PROGRAMS CALLED**
  Indicates the number of programs called (for example, the number of #LOAD PGM requests issued).

- **MAX # RLE’S USED**
  Indicates the highest number of resource link elements (RLEs) used during this session. If this value approximates the threshold established at system generation, increase the threshold.

- **MAX # RCE’S USED**
  Indicates the highest number of resource control elements (RCEs) used during this session. If this value approximates the threshold established at system generation, increase the threshold.

- **MAX # DPE’S USED**
  Indicates the highest number of deadlock prevention elements (DPEs) used during this session. If this value approximates the threshold established at system generation, increase the threshold.

- **STACK HI WATERMARK**
  Indicates the largest amount of the task control element (TCE) stack area used by any task. The stack size is defined at system generation with the STACKSIZE clause of the SYSTEM statement.
- **GET STORAGES**
  Indicates the number of program requests to acquire variable storage dynamically from a DC/UCF storage pool or obtain the address of a previously acquired storage area. The program issues the request with the GET STORAGE DML statement.

- **FREE STORAGES**
  Indicates the number of program requests to free all or part of a DC/UCF storage area. The program issues the request with the FREE STORAGE DML statement.

- **SYSTEM MODE CPU TIME**
  Indicates the amount of CPU time (in ten-thousandths seconds) spent in executing system code.

- **GET SCRATCHES**
  Indicates the number of program requests for scratch records from the DDLDCSCR area. The program issues the request with the GET SCRATCH DML statement.

- **PUT SCRATCHES**
  Indicates the number of program requests to store or replace a scratch record in the DDLDCSCR area. The program issues the request with the PUT SCRATCH DML statement.

- **DELETE SCRATCHES**
  Indicates the number of program requests to delete scratch records from the DDLDCSCR area. The program issues the request with the DELETE SCRATCH DML statement.

- **GET QUEUES**
  Indicates the number of program requests to retrieve a queue record from the DDLDCRUN area and place it in a storage area associated with the issuing program. The program issues the request with the GET QUEUE DML statement.

- **PUT QUEUES**
  Indicates the number of program requests to store a queue record in the DDLDCRUN area. The program issues the request with the PUT QUEUE DML statement.

- **DELETE QUEUES**
  Indicates the number of program requests to delete queue records from the DDLDCRUN area. The program issues the request with the DELETE QUEUE DML statement.

- **GET TIMES**
  Indicates the number of program requests for the system date and time. The program issues the request with the GET TIME DML statement.

- **SET TIMES**
  Indicates the number of SETTIME requests to define an event that is to occur after a specified time interval.

- **SYSTEM MODE CPU**
  Indicates the amount of CPU time (in microseconds) spent in executing system code.

- **USER MODE CPU**
  Indicates the amount of CPU time (in microseconds) spent in executing user code.

- **ZIIP on ZIIP CPU**
  Indicates the amount of CPU time (in microseconds) spent on zIIP.
- **ZIIP on CP CPU**
  Indicates the amount of CPU time (in microseconds) spent on CP, while qualified for zIIP.

- **TOTAL TCB CPU TIME**
  Indicates the amount of CPU time (in microseconds) spent on CP in either system mode or user mode.

- **ENCLAVE CPU**
  Indicates the amount of CPU time (in microseconds) spent on CP or zIIP, while qualified for zIIP.

- **PAGES REQUESTED**
  Indicates the number of pages requested by IDMSDBMS (including pages found in a buffer). A page request does not result in a page read if the page is in the buffer pool.
  Interpretation: The ratio of PAGES REQUESTED/PAGES READ is the buffer utilization ratio. It measures the effectiveness of the buffer-pool size and design of the database (for example, CALC and VIA clustering). The higher the ratio the better. Ratios consistently below 2.0 indicate that processing is random or that the buffer-pool size is too small.
  The buffer utilization ratio may be artificially high for transactions that keep locks, due to the nature of the internal locking mechanism. IDMSDBMS cannot hold a buffer while requesting a lock; therefore, when locks are kept, IDMSDBMS must free and request a page each time a record is requested.

- **PAGES READ**
  Indicates the number of database pages read from disk.

- **PAGES WRITTEN**
  Indicates the number of database pages physically written to disk. A page can be updated several times before it is actually written back to the database.

- **CALLS TO DBMS**
  Indicates the number of calls to the database management system.

  **Note:** Execution of each navigational DML request involves one call; execution of each logical record facility (LRF) and SQL request typically involves multiple calls.

- **RECORDS REQUESTED**
  Indicates the number of database records requested by IDMSDBMS.
  Interpretation: The ratio of RECORDS REQUESTED to PAGES READ is the space management ratio. The space management ratio measures how well space is allocated (for example, VIA options, CALC distribution, and buffering). The higher the ratio the better. Ratios less than 4 or less than the norm indicate that the size of the buffer should be increased and database tuning should be performed.
  The space management ratio may be artificially high for transactions that keep locks, due to the nature of the internal locking mechanism. IDMSDBMS cannot hold a buffer while requesting a lock; therefore, when locks are kept, IDMSDBMS must free and request a page each time a record is requested.

- **RECORDS CURRENT OF RU**
  Indicates the number of records made current of transaction.
  Interpretation: The ratio of RECORDS REQUESTED to RECORDS CURRENT is the effectiveness ratio. The effectiveness ratio measures the amount of work CA IDMS/DB is doing for the programmer.
(that is, how many records the DBMS has to examine to find the one requested). The lower the ratio the better. If the ratio is high, examine set options (for example, sorted order or next pointers only) for appropriateness. If the options are correct, examine the program logic for accurate use of currency.

- **TOTAL LOCKS**
  Indicates the number of all locks acquired and released by all completed transactions. This is NOT a count of locks currently held.

- **PAGES FOUND IN CACHE**
  Indicates the number of requested database pages that have been found in a shared cache or an ESA dataspace.

- **CALCRECS NO OFLOW**
  Indicates the number of CALC records stored on the target page.

- **CALC RECS OFLOW**
  Indicates the number of CALC records not stored on the target page.
  Interpretation: The ratio of CALC records stored on their target page to the total number stored (that is, hits plus overflows) is the **CALC cluster ratio**. The ratio reflects the efficiency of the CALC algorithm.
  The CALC cluster ratio is especially important when the database is loaded or restructured. Ideally, the ratio should be 1, which indicates no overflow. Ratios consistently less than 1 or less than the norm indicate that space utilization is getting high and the database should be tuned.

- **VIA RECS NO OFLOW**
  Indicates the number of VIA and/or DIRECT records stored on the target page.

- **VIA RECS OFLOW**
  Indicates the number of VIA and/or DIRECT records not stored on the target page.
  Interpretation: The ratio of VIA records stored on their target page to the total number of VIA records stored (that is, hits plus overflows) is the **VIA cluster ratio**. The ratio reflects how well VIA records cluster around their owner.
  Ideally, the ratio should be 1, which indicates no overflow. Ratios less than 1 or less than the norm indicate very large data clusters, high utilization of space, or small page size.

- **FRAGMENTS STORED**
  Indicates the number of noncontiguous segments (fragments) stored for variable-length records.

- **RECORDS RELOCATED**
  Indicates the number of records relocated from their home page.

- **PAGES IN PREFETCH BUFF**
  Indicates the number of database pages that have been directly found in a prefetch buffer.

- **SR8 SPLITS**
  Indicates the number of SR8 splits.

- **SR8 SPAWNS**
  Indicates the number of SR8 spawns.
- **ORPHANS ADOPTED**
  Indicates count of index members or SR8s whose up-level pointers were corrected to point to the actual SR8 in which they appear.

- **BTREE SEARCHES**
  Indicates number of Btree index probes.

- **MIN LEVELS SEARCHED**
  Indicates least number of levels descended.

- **MAX LEVELS SEARCHED**
  Indicates highest number of levels descended.

- **SR8 STORES**
  Indicates count of SR8s created.

- **SR8 ERASES**
  Indicates count of SR8s erased.

- **SR7 STORES**
  Indicates count of SR7s created.

- **SR7 ERASES**
  Indicates count of SR7s erased.

- **TOTAL LEVELS SEARCHED**
  Indicates total number of levels descended.

- **SQL COMMANDS**
  Indicates the number of SQL commands executed.

- **SORTS**
  Indicates the number of SQL sorts performed.

- **TUPLES SORTED**
  Indicates the number of rows participating in all sorts.

- **MIN SORT**
  Indicates the least number of rows sorted.

- **MAX SORT**
  Indicates the largest number of rows sorted.

- **TUPLES FETCHED**
  Indicates number of tuples FETCHed.

- **ROWS INSERTED**
  Indicates the number of rows INSERTed.

- **ROWS UPDATED**
  Indicates the number of rows UPDATed.
- **ROWS DELETED**
  Indicates the number of rows DELETEd.

- **AM RECOMPILES**
  Indicates the number of automatic access module recompletions.

For a detailed explanation of DML commands, see the *CA IDMS Navigational DML Administrating section*. For more information on system generation statements, see the *CA IDMS Administrating section*.

**SREPORT 012 - IDMS DC Task Summary**

**Contents**

SREPORT 012, the IDMS DC Task Summary report, summarizes systemwide task statistics, indicating the number of times each task was invoked. Systems administrators can use SREPORT 012 to monitor trends in task usage; for example, watching for peaks in task usage over a set time period.

The following figure shows one page of a sample IDMS DC Task Summary report:

**Sample SREPORT 012:**

<table>
<thead>
<tr>
<th>TASK</th>
<th>TIMES INVOKED</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>0</td>
</tr>
<tr>
<td>SCHEMA</td>
<td>0</td>
</tr>
<tr>
<td>SCHEMAT</td>
<td>0</td>
</tr>
<tr>
<td>SEND</td>
<td>0</td>
</tr>
<tr>
<td>SHOWMAP</td>
<td>0</td>
</tr>
<tr>
<td>SIGNOFF</td>
<td>0</td>
</tr>
<tr>
<td>SIGNON</td>
<td>0</td>
</tr>
<tr>
<td>SSC</td>
<td>0</td>
</tr>
<tr>
<td>SSCT</td>
<td>0</td>
</tr>
<tr>
<td>SUSPEND</td>
<td>0</td>
</tr>
<tr>
<td>SYSGEN</td>
<td>0</td>
</tr>
<tr>
<td>SYSGENT</td>
<td>0</td>
</tr>
<tr>
<td>TCF</td>
<td>0</td>
</tr>
<tr>
<td>TOTAL</td>
<td>4</td>
</tr>
</tbody>
</table>

**Field Descriptions**

A description of the fields in the IDMS DC Task Summary report follows:
SELECTED FROM/TO
Specifies the time period specified on the USE 'SREPORT 000' parameter where SELECTED FROM specifies the beginning date and time and SELECTED TO specifies the ending date and time. The date is in Julian form, yyddd, where yy is the last two digits of the year and ddd is the day. The time is in hh:mm form, where hh is hours based on a 24-hour clock and mm is minutes.

CV SYSTEM START/INTERVAL START/TO
Specifies the actual time period for the report generated. If the USE 'SREPORT 000' parameter does not specify a time range, CV SYSTEM START/INTERVAL START and TO represent the time range for the entire input archive file or input file created by SREPORT 099. When the input file contains cumulative statistics, the CV SYSTEM START date and time will be displayed. When the input file contains interval based statistics, the INTERVAL START date and time will be displayed.

CV JOB NAME
Specifies the starting job name of Central Version which collected the statistics.

CV NUMBER
Specifies the number of Central Version which collected the statistics.

TASK
Specifies the task identifier that is used at run time by a terminal operator or program to invoke the task. A task is the basic unit of work under DC/UCF. It consists of a main program and one or more additional programs. A task is identified to the system by a unique name (such as OLM) that is usually identical to the task code used by the teleprocessing system.

TIMES INVOKED
Specifies the number of times the task was invoked for the session.

SREPORT 013 - IDMS DC Program Summary

Contents
SREPORT 013, the IDMS DC Program Summary report, summarizes systemwide program statistics. System administrators can use SREPORT 013 to monitor program activity relative to available storage.

Available Program Pools
A program can be assigned to any one of the following pools:

- 24-bit program pool
- 24-bit reentrant pool
- 31-bit program pool (Extended addressing only)
- 31-bit reentrant pool (Extended addressing only)

Note: A program can be put into both a 24-bit pool and a 31-bit pool depending on the LOC= parameter used to invoke the task.
The following figure shows one page of a sample IDMS DC Program Summary report:

**Sample SREPORT 013:**

<table>
<thead>
<tr>
<th>PROGRAM NAME</th>
<th>VER</th>
<th>DICTNAME</th>
<th>DICTNODE</th>
<th>TIMES CALLED</th>
<th>TIMES LOADED</th>
<th>RATIO OF CALLED/LOADED</th>
<th>TIMES WAITED</th>
<th>TIMES CHECKED</th>
</tr>
</thead>
<tbody>
<tr>
<td>RHDCWAIT</td>
<td>1</td>
<td></td>
<td></td>
<td>0</td>
<td>1</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RHDCWTL</td>
<td>1</td>
<td></td>
<td></td>
<td>0</td>
<td>1</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RM000121</td>
<td>1</td>
<td>ASFDICT</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RM000122</td>
<td>1</td>
<td>ASFDICT</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RM000123</td>
<td>1</td>
<td>ASFDICT</td>
<td></td>
<td>18</td>
<td>4</td>
<td>4.5000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RM000124</td>
<td>1</td>
<td>ASFDICT</td>
<td></td>
<td>18</td>
<td>4</td>
<td>4.5000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RU000121</td>
<td>1</td>
<td>ASFDICT</td>
<td></td>
<td>3</td>
<td>3</td>
<td>1.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RU000122</td>
<td>1</td>
<td>ASFDICT</td>
<td></td>
<td>3</td>
<td>3</td>
<td>1.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RU000123</td>
<td>1</td>
<td>ASFDICT</td>
<td></td>
<td>3</td>
<td>3</td>
<td>1.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RU000124</td>
<td>1</td>
<td>ASFDICT</td>
<td></td>
<td>20</td>
<td>8</td>
<td>2.5000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSTDNWKA</td>
<td>1</td>
<td></td>
<td></td>
<td>42</td>
<td>1</td>
<td>42.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSTDNWKS</td>
<td>1</td>
<td></td>
<td></td>
<td>2</td>
<td>1</td>
<td>2.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UMBRMAP</td>
<td>1</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UMBRMAPE</td>
<td>1</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XASFMNKS</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TOTALS</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>1,458</strong></td>
<td><strong>181</strong></td>
<td><strong>8.0552</strong></td>
<td><strong>0</strong></td>
<td><strong>0</strong></td>
</tr>
</tbody>
</table>

**Field Descriptions**

A description of the fields in the IDMS DC Program Summary report follows:

- **SELECTED FROM/TO**
  Specifies the time period specified on the USE 'SREPORT 000' parameter where SELECTED FROM specifies the beginning date and time and SELECTED TO specifies the ending date and time. The date is in Julian form, yyddd, where yy is the last two digits of the year and ddd is the day. The time is in hh:mm form, where hh is hours based on a 24-hour clock and mm is minutes.

- **CV SYSTEM START/INTERVAL START/TO**
  Specifies the actual time period for the report generated. If the USE 'SREPORT 000' parameter does not specify a time range, CV SYSTEM START/INTERVAL START and TO represent the time range for the entire input archive file or input file created by SREPORT 099. When the input file contains cumulative statistics, the CV SYSTEM START date and time will be displayed. When the input file contains interval based statistics, the INTERVAL START date and time will be displayed.
- **CV JOB NAME**
  Specifies the starting job name of Central Version which collected the statistics.

- **CV NUMBER**
  Specifies the number of Central Version which collected the statistics.

- **PROGRAM NAME**
  Specifies the name of the program load module.

- **PGM VER**
  Indicates the version number associated with the named program.

- **DICTNAME**
  Names the dictionary in which the program resides.

- **DICTNODE**
  Names the DC/UCF system that controls the dictionary in which the program resides.

- **TIMES CALLED**
  Indicates the number of times the program was called.

- **TIMES LOADED**
  Indicates the number of times the program was loaded from disk to a pool.
  Interpretation: The ratio of TIMES CALLED to TIMES LOADED measures the effectiveness of the program pool size. The higher the ratio the better. A low ratio for a frequently called program indicates that the size of the program pool should be enlarged or that the program should be made resident, reentrant, or reusable.

- **RATIO OF CALLED/LOADED**
  Measures the effectiveness of the program pool size. The higher the ratio the better. A low ratio for a frequently called program indicates that the size of the program pool should be enlarged or that the program should be made resident, reentrant, or reusable.

- **TIMES WAITED**
  Indicates the number of times the program waited to be loaded.

- **TIMES CHECKED**
  Indicates the number of program check errors that occurred.

---

**SREPORT 014 - IDMS DC Queue Summary**

**Contents**

SREPORT 014, the IDMS DC Queue Summary report, summarizes systemwide queue statistics. System administrators can use SREPORT 014 to monitor queue activity.

**Sample SREPORT 014:**

<table>
<thead>
<tr>
<th>REPORT NO. 14</th>
<th>IDMS-DC QUEUE SUMMARY</th>
<th>Rnn.n</th>
<th>mm/dd/yy</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECTED FROM: yyddd hh:mm</td>
<td>TO: yyddd hh:mm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CV SYSTEM START: yyddd hh:mm</td>
<td>TO: yyddd hh:mm</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Field Descriptions

A description of the fields in the IDMS DC Queue Summary report follows:

- **SELECTED FROM/TO**
  Specifies the time period specified on the USE 'SREPORT 000' parameter where SELECTED FROM specifies the beginning date and time and SELECTED TO specifies the ending date and time. The date is in Julian form, yyyyddd, where yy is the last two digits of the year and ddd is the day. The time is in hh:mm form, where hh is hours based on a 24-hour clock and mm is minutes.

- **CV SYSTEM START/INTERVAL START/TO**
  Specifies the actual time period for the report generated. If the USE 'SREPORT 000' parameter does not specify a time range, CV SYSTEM START/INTERVAL START and TO represent the time range for the entire input archive file or input file created by SREPORT 099. When the input file contains cumulative statistics, the CV SYSTEM START date and time will be displayed. When the input file contains interval based statistics, the INTERVAL START date and time will be displayed.

- **CV JOB NAME**
  Specifies the starting job name of Central Version which collected the statistics.

- **CV NUMBER**
  Specifies the number of Central Version which collected the statistics.

- **QUEUE NAME**
  Specifies the queue identifier. A queue is a database work area shared by tasks on all DC/UCF terminals and by batch programs. Queue records can be transferred between tasks or applications or from one terminal to another.

- **TIMES AUTOTASK STARTED**
  Indicates the number of times each queue's associated task was invoked to process queue records. If the value is consistently zero, review the threshold level assigned in the THRESHOLD IS clause of the QUEUE system generation statement; For more information, see the *CA IDMS Administrating section*.

SREPORT 015 - IDMS DC Line Summary

Contents

SREPORT 015, the IDMS DC Line Summary report, summarizes systemwide line statistics. Systems administrators can use SREPORT 015 to monitor the quality of lines in the system.

Sample SREPORT 015:
Field Descriptions

A description of the fields in the IDMS DC Line Summary report follows:

- **SELECTED FROM/TO**
  Specifies the time period specified on the USE 'SREPORT 000' parameter where SELECTED FROM specifies the beginning date and time and SELECTED TO specifies the ending date and time. The date is in Julian form, yyddd, whereyy is the last two digits of the year and ddd is the day. The time is in hh:mm form, where hh is hours based on a 24-hour clock and mm is minutes.

- **CV SYSTEM START/INTERVAL START/TO**
  Specifies the actual time period for the report generated. If the USE 'SREPORT 000' parameter does not specify a time range, CV SYSTEM START/INTERVAL START and TO represent the time range for the entire input archive file or input file created by SREPORT 099. When the input file contains cumulative statistics, the CV SYSTEM START date and time will be displayed. When the input file contains interval based statistics, the INTERVAL START date and time will be displayed.

- **CV JOB NAME**
  Specifies the starting job name of Central Version which collected the statistics.

- **CV NUMBER**
  Specifies the number of Central Version which collected the statistics.

- **LINE NAME**
  Specifies the line identifier. A line is a system component that defines the method of communication for physical terminals that use the same access method.

- **READS**
  Indicates the number of reads performed for each line.

- **WRITES**
  Indicates the number of writes performed for each line.
READ ERRORS
Indicates the number of read errors that occurred for each line. A large value indicates problems in the line.

WRITE ERRORS
Indicates the number of write errors that occurred for each line. A large value indicates problems in the line.

SREPORT 016 - IDMS DC Physical Terminal Summary

Contents
SREPORT 016, the IDMS DC Physical Terminal Summary report, summarizes system wide line statistics.

Sample SREPORT 016:

REPORT NO. 16  IDMS-DC PHYSICAL TERMINAL SUMMARY  Rnn.n  mm/dd/yy  PAGE  2
SELECTED FROM:  yyddd  hh:mm  TO:  yyddd  hh:mm
CV SYSTEM START:  yyddd  hh:mm  TO:  yyddd  hh:mm
CV JOB NAME:  jobname
CV NUMBER:  nnn

<table>
<thead>
<tr>
<th>PHYSICAL TERMINAL</th>
<th>READS</th>
<th>WRITES</th>
<th>READ ERRORS</th>
<th>WRITE ERRORS</th>
<th>RESPONSES</th>
<th>CUMULATIVE RESPONSE TIME</th>
<th>AVERAGE RESPONSE TIME</th>
<th>CUMULATIVE I/O TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>VP91007</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>VP91008</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>VP91009</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>VP91010</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>VP91011</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>VP91012</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>VP91013</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>VP91014</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>VP91015</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>TOTALS</td>
<td>312</td>
<td>238</td>
<td>0</td>
<td>0</td>
<td></td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Field Descriptions

A description of the fields in the IDMS DC Physical Terminal Summary report follows:

SELECTED FROM/TO
Specifies the time period specified on the USE 'SREPORT 000' parameter where SELECTED FROM specifies the beginning date and time and SELECTED TO specifies the ending date and time. The date is in Julian form, yyddd, where yy is the last two digits of the year and add is the day. The time is in hh:mm form, where hh is hours based on a 24-hour clock and mm is minutes.
• **CV SYSTEM START/INTERVAL START/TO**
  Specifies the actual time period for the report generated. If the USE 'SREPORT 000' parameter does not specify a time range, CV SYSTEM START/INTERVAL START and TO represent the time range for the entire input archive file or input file created by SREPORT 099. When the input file contains cumulative statistics, the CV SYSTEM START date and time will be displayed. When the input file contains interval based statistics, the INTERVAL START date and time will be displayed.

• **CV JOB NAME**
  Specifies the starting job name of Central Version which collected the statistics.

• **CV NUMBER**
  Specifies the number of Central Version which collected the statistics.

• **PHYSICAL TERMINAL**
  Specifies the physical terminal identifier. A physical terminal is a physical device such as a CRT (3270-type device), TTY, or printer that exists within a teleprocessing system.

• **READS**
  Indicates the number of reads performed for each physical terminal.

• **WRITES**
  Indicates the number of writes performed for each physical terminal.

• **READ ERRORS**
  Indicates the number of read errors that occurred for each physical terminal.

• **WRITE ERRORS**
  Indicates the number of write errors that occurred for each physical terminal.

• **RESPONSES**
  Indicates the number of physical terminal responses. A response begins when a physical terminal issues a read request and ends when the next read request is issued.

• **CUMULATIVE RESPONSE TIME**
  Indicates the cumulative response time, in hundredths of a second, for the physical terminal. Response time is the total non-I/O time measured from the one read request to the next.

• **AVERAGE RESPONSE TIME**
  Specifies the ratio of CUMULATIVE RESPONSE TIME to RESPONSES. Systems administrators can monitor this value to determine the impact of system configuration changes upon response time.

• **CUMULATIVE I/O TIME**
  Specifies the cumulative I/O time, in hundredths of a second, for the physical terminal.

---

**Task and External Request Unit Service (ERUS) Statistics Reports**

Task statistics and external request unit service (ERUS) statistics record the resource usage of individual tasks and external request units. The following considerations apply:
Task and ERUS statistics, including database statistics, incorporate statistics of system run units (RHDCRUAL/IDMSXTAL) when these run units do work for the task. For example, loading a load module from a load area requires database requests which show up in the task statistics.

Task and ERUS statistics include system-mode statistics; that is, when DC/UCF does work for the task. For example, when a program issues a BIND RUN UNIT, the system obtains variable storage; the storage requests show up in the task statistics.

Uses for Task and ERUS Statistics (see page 232)
Statistics Collected Only Upon Request (see page 232)
Summary of Task Statistics Reports (see page 232)
Field Descriptions (see page 236)

Uses for Task and ERUS Statistics

Task and ERUS statistics are useful for monitoring and tuning individual application programs.

Statistics Collected Only Upon Request

DC/UCF collects task and ERUS statistics only when requested to do so by the user, because the statistics require additional overhead and, when written to the log file, generate a large volume of data.

Summary of Task Statistics Reports

Five statistics reports summarize task and external request unit activity:

| SREPORT 004 | Summarizes task statistics by IP Address. |
| SREPORT 005 | Summarizes task statistics by user identifier. |
| SREPORT 006 | Summarizes task statistics by logical terminal identifier. |
| SREPORT 007 | Summarizes task statistics by task code. |
| SREPORT 008 | Summarizes ERUS statistics by accounting data. |
| SREPORT 009 | Summarizes ERUS statistics by program name. |

Sample SREPORT 004:

REPORT NO. 04 IDMS-DC TASK STATISTICS BY IP ADDRESS Rnn.n mm/dd/yy PAGE 1
SELECTED FROM: yyyy dd hh:mm TO: yyyy dd hh:mm
CV SYSTEM START: yyyy dd hh:mm TO: yyyy dd hh:mm
CV JOB NAME: jobname
CV NUMBER: nnn
IP ADDRESS: 138.42.193.137

3 NUMBER TASK EXECUTIONS
<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number Programs Called</td>
<td>0</td>
<td>Number Programs Loaded</td>
<td>0</td>
</tr>
<tr>
<td>Number Terminal Reads</td>
<td>0</td>
<td>Number Terminal Writes</td>
<td>0</td>
</tr>
<tr>
<td>Number Terminal Errors</td>
<td>111</td>
<td>Number GETSTG Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number GETSCR Requests</td>
<td>0</td>
<td>Number PUTSCR Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number DELSCR Requests</td>
<td>0</td>
<td>Number GETQUE Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number PUTQUE Requests</td>
<td>0</td>
<td>Number DELQUE Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number GETTIME Requests</td>
<td>144</td>
<td>Number SETTIME Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number DB Services Rqsts</td>
<td>0</td>
<td>Number Settime Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number Pages Read</td>
<td>0</td>
<td>Number Pages Read</td>
<td>0</td>
</tr>
<tr>
<td>Task User Mode Time</td>
<td>.0011</td>
<td>Task System Mode Time</td>
<td>.0023</td>
</tr>
<tr>
<td>Task Wait Time</td>
<td>10.2754</td>
<td>Task Wait Time</td>
<td>0.0811</td>
</tr>
<tr>
<td>Number Pages Written</td>
<td>0</td>
<td>Number Pages Requested</td>
<td>0</td>
</tr>
<tr>
<td>Number Calc Recs No OFLOW</td>
<td>0</td>
<td>Number Calc Recs OFLOW</td>
<td>0</td>
</tr>
<tr>
<td>Number Via Recs No OFLOW</td>
<td>0</td>
<td>Number Via Recs OFLOW</td>
<td>0</td>
</tr>
<tr>
<td>Number Records Requested</td>
<td>0</td>
<td>Number Records Current of RU</td>
<td>0</td>
</tr>
<tr>
<td>Number Fragments Stored</td>
<td>0</td>
<td>Number Records Relocated</td>
<td>0</td>
</tr>
<tr>
<td>Number Calls to DBMS</td>
<td>0</td>
<td>Total Locks Acquired</td>
<td>0</td>
</tr>
<tr>
<td>Number Calls to DBMS</td>
<td>0</td>
<td>Number Locks Acquired</td>
<td>0</td>
</tr>
<tr>
<td>Number SRB Splits</td>
<td>0</td>
<td>Number SR8 Stores</td>
<td>0</td>
</tr>
<tr>
<td>Number SRB SPANS</td>
<td>0</td>
<td>Number SR8 ERASES</td>
<td>0</td>
</tr>
<tr>
<td>Number ORPHAN ADOPT</td>
<td>0</td>
<td>Number ORPHAN ADOPT</td>
<td>0</td>
</tr>
<tr>
<td>Number BTREE Searches</td>
<td>0</td>
<td>Number SQL COMMANDS</td>
<td>0</td>
</tr>
<tr>
<td>Number SQL COMMANDS</td>
<td>0</td>
<td>Number SQL SORTS</td>
<td>0</td>
</tr>
<tr>
<td>Number Tuples Fetched</td>
<td>0</td>
<td>Number Tuples Sorted</td>
<td>0</td>
</tr>
<tr>
<td>Number Rows Inserted</td>
<td>0</td>
<td>Number AM RECOMPILES</td>
<td>0</td>
</tr>
<tr>
<td>Number Rows Updated</td>
<td>0</td>
<td>Number AM DECOMPILES</td>
<td>0</td>
</tr>
<tr>
<td>Number Rows Deleted</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task User Mode CPU</td>
<td>.002443</td>
<td>Task ZIIP on CP CPU</td>
<td>.000000</td>
</tr>
<tr>
<td>Task ZIIP on ZIIP CPU</td>
<td>.000000</td>
<td>Task USER Mode CPU</td>
<td>.000140</td>
</tr>
<tr>
<td>Total Task TCB CPU</td>
<td>.003583</td>
<td>Enclave CPU</td>
<td>.000000</td>
</tr>
</tbody>
</table>

Sample SREPORT 005:

```
REPORT NO. 85  IDMS-DC TASK STATISTICS BY USER ID  Rnn.n  mm/dd/yy PAGE 1
SELECTED FROM: yyddd hh:mm  TO: yyddd hh:mm
CV SYSTEM START: yyddd hh:mm  TO: yyddd hh:mm
CV JOB NAME: jobname
CV NUMBER: nnn
USER ID: JEJ

13 NUMBER TASK EXECUTIONS

20 NUMBER PROGRAMS CALLED   1 NUMBER PROGRAMS LOADED
11 NUMBER TERMINAL READS   159 NUMBER TERMINAL WRITES
 0 NUMBER TERMINAL ERRORS   471 NUMBER TERMINAL WRITES
 0 NUMBER GETSCR REQUESTS  0 NUMBER PUTSCR REQUESTS
 0 NUMBER DELSCR REQUESTS  2 NUMBER GETQUE REQUESTS
 0 NUMBER PUTQUE REQUESTS  0 NUMBER DELQUE REQUESTS
151 NUMBER GETTIME REQUESTS  5 NUMBER SETTIME REQUESTS
128 NUMBER DB SERVICES Rqsts  7 NUMBER DB SERVICES Rqsts
 0 NUMBER TASK USER MODE TIME  .0536 TASK SYSTEM MODE CPU
7,573.0212 TASK WAIT TIME
 0 NUMBER PAGES WRITTEN  50 NUMBER PAGES REQUESTED
 0 NUMBER CALC Recs NO OFLOW  0 NUMBER CALC Recs OFLOW
 0 NUMBER VIA Recs NO OFLOW  0 NUMBER VIA Recs OFLOW
85 NUMBER RECORDS Requested  33 NUMBER RECORDS Relocated
160 NUMBER Calls to DBMS  16 TOTAL Locks Acquired
 0 NUMBER SRB Splits  0 NUMBER SRB Stores
 0 NUMBER SRB SPANS  0 NUMBER SRB ERASES
 0 NUMBER ORPHAN ADOPT  0 NUMBER ORPHAN ADOPT
 0 NUMBER BTREE SEARCHES  0 NUMBER BTREE SEARCHES
 0 NUMBER SQL COMMANDS  0 NUMBER SQL COMMANDS
 0 NUMBER Tuples Fetched  0 NUMBER Tuples Sorted
 0 NUMBER Rows Inserted  0 NUMBER AM RECOMPILES
 0 NUMBER Rows Updated  0 NUMBER Rows Deleted
 0.006329 TASK SYSTEM MODE CPU  .000000 TASK ZIIP on CP CPU
 0.000000 TASK ZIIP on ZIIP CPU  .001081 TASK USER Mode CPU
 0.003583 TOTAL TASK TCB CPU  .000000 ENCLAVE CPU
```
### Sample SREPORT 006:

<table>
<thead>
<tr>
<th>Report No.</th>
<th>IDMS-DC Task Statistics by LTERM ID Rnn.n</th>
<th>mm/dd/yy</th>
<th>Page 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected From:</td>
<td>yyddd hh:mm To: yyddd hh:mm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CV System Start:</td>
<td>yyddd hh:mm To: yyddd hh:mm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CV Job Name:</td>
<td>jobname</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CV Number:</td>
<td>nnn</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**LTERM ID:** VL71001

13 NUMBER TASK EXECUTIONS

<table>
<thead>
<tr>
<th>Task</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number Programs Called</td>
<td>1</td>
</tr>
<tr>
<td>Number Terminal Reads</td>
<td>11</td>
</tr>
<tr>
<td>Number Terminal Errors</td>
<td>8</td>
</tr>
<tr>
<td>Number GetScr Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number DelScr Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number PutQue Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number PutScr Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number DelQue Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number Gettime Requests</td>
<td>151</td>
</tr>
<tr>
<td>Number Getime Requests</td>
<td>5</td>
</tr>
<tr>
<td>Number DB Service Rests</td>
<td>128</td>
</tr>
<tr>
<td>Number Pages Read</td>
<td>7</td>
</tr>
</tbody>
</table>

**Task Wait Time**

0.060329 TASK SYSTEM MODE CPU

#### Sample SREPORT 007:

<table>
<thead>
<tr>
<th>Report No.</th>
<th>IDMS-DC Task Statistics by TASK CODE Rnn.n</th>
<th>mm/dd/yy</th>
<th>Page 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected From:</td>
<td>yyddd hh:mm To: yyddd hh:mm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CV System Start:</td>
<td>yyddd hh:mm To: yyddd hh:mm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CV Job Name:</td>
<td>jobname</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CV Number:</td>
<td>nnn</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Task Code:** USGAFIX

1 NUMBER TASK EXECUTIONS

<table>
<thead>
<tr>
<th>Task</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number Programs Called</td>
<td>5</td>
</tr>
<tr>
<td>Number Terminal Reads</td>
<td>0</td>
</tr>
<tr>
<td>Number Terminal Errors</td>
<td>0</td>
</tr>
<tr>
<td>Number GetScr Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number DelScr Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number PutQue Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number PutScr Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number DelQue Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number Gettime Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number Getime Requests</td>
<td>1</td>
</tr>
<tr>
<td>Number DB Service Rests</td>
<td>11</td>
</tr>
<tr>
<td>Number Pages Read</td>
<td>1</td>
</tr>
</tbody>
</table>

**Task Wait Time**

0.08008 TASK ZIIP CPU

#### Sample SREPORT 007:

<table>
<thead>
<tr>
<th>Report No.</th>
<th>IDMS-DC Task Statistics by TASK CODE Rnn.n</th>
<th>mm/dd/yy</th>
<th>Page 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected From:</td>
<td>yyddd hh:mm To: yyddd hh:mm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CV System Start:</td>
<td>yyddd hh:mm To: yyddd hh:mm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CV Job Name:</td>
<td>jobname</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CV Number:</td>
<td>nnn</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Task Code:** USGAFIX

1 NUMBER TASK EXECUTIONS

<table>
<thead>
<tr>
<th>Task</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number Programs Called</td>
<td>5</td>
</tr>
<tr>
<td>Number Terminal Reads</td>
<td>0</td>
</tr>
<tr>
<td>Number Terminal Errors</td>
<td>0</td>
</tr>
<tr>
<td>Number GetScr Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number DelScr Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number PutQue Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number PutScr Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number DelQue Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number Gettime Requests</td>
<td>0</td>
</tr>
<tr>
<td>Number Getime Requests</td>
<td>1</td>
</tr>
<tr>
<td>Number DB Service Rests</td>
<td>11</td>
</tr>
<tr>
<td>Number Pages Read</td>
<td>1</td>
</tr>
</tbody>
</table>
Sample SREPORT 008:

REPORT NO. 08
IDMS-DC ERUS TASK STATISTICS BY ACCTG DATA
Rnn.n
mm/dd/yy PAGE 1
SELECTED FROM: yyyddd hh:mm TO: yyyddd hh:mm
CV SYSTEM START: yyyddd hh:mm TO: yyyddd hh:mm
CV JOB NAME: jobname
CV NUMBER: nnn
ACCOUNTING DATA: KRCJA020..111400030

3
NUMBER TASK EXECUTIONS

6
NUMBER PROGRAMS CALLED
1
NUMBER PROGRAMS LOADED
0
NUMBER TERMINAL READS
0
NUMBER TERMINAL WRITES
0
NUMBER TERMINAL ERRORS
43
NUMBER GETSTG REQUESTS
0
NUMBER PUTSCR REQUESTS
0
NUMBER DELSCR REQUESTS
0
NUMBER GETQUE REQUESTS
0
NUMBER PUTQUE REQUESTS
45
NUMBER GETTIME REQUESTS
1
NUMBER SETTIME REQUESTS
1
NUMBER DB SERVICE REQUESTS
519,761
NUMBER PAGES READ

2.5343
TASK SYSTEM MODE TIME
390.7796
TOTAL TASK TCB TIME

1,166,589
NUMBER PAGES REQUESTED

646,839
NUMBER RECORDS REQUESTED

642,013
NUMBER RECORDS CURRENT OF RU

462,002
NUMBER RECORDS RELOCATED

Sample SREPORT 009:

REPORT NO. 09
IDMS-DC ERUS TASK STATISTICS BY PGM NAME
Rnn.n
mm/dd/yy PAGE 1
SELECTED FROM: yyyddd hh:mm TO: yyyddd hh:mm
CV SYSTEM START: yyyddd hh:mm TO: yyyddd hh:mm
CV JOB NAME: jobname
CV NUMBER: nnn
PROGRAM NAME: ERZPTMSF

1
NUMBER TASK EXECUTIONS

2
NUMBER PROGRAMS CALLED
0
NUMBER PROGRAMS LOADED
0
NUMBER TERMINAL READS
0
NUMBER TERMINAL ERRORS
14
NUMBER GETSTG REQUESTS
0
NUMBER PUTSCR REQUESTS
0
NUMBER DELSCR REQUESTS
0
NUMBER GETQUE REQUESTS
0
NUMBER PUTQUE REQUESTS
15
NUMBER SETTIME REQUESTS
14
NUMBER DB SERVICE REQUESTS
371,254
NUMBER PAGES READ

1.9449
TASK SYSTEM MODE TIME
280.8358
TOTAL TASK TCB TIME

833,263
NUMBER PAGES REQUESTED

462,013
NUMBER RECORDS REQUESTED
### Field Descriptions

A description of the fields common to SREPORTs 004 through 009 follows:

- **SELECTED FROM/TO**
  Specifies the time period specified on the USE 'SREPORT 000' parameter where SELECTED FROM specifies the beginning date and time and SELECTED TO specifies the ending date and time. The date is in Julian form, `yyddd`, where `yy` is the last two digits of the year and `ddd` is the day. The time is in `hh:mm` form, where `hh` is hours based on a 24-hour clock and `mm` is minutes.

- **CV SYSTEM START/INTERVAL START/TO**
  Specifies the actual time period for the report generated. If the USE 'SREPORT 000' parameter does not specify a time range, CV SYSTEM START/INTERVAL START and TO represent the time range for the entire input archive file or input file created by SREPORT 099. When the input file contains cumulative statistics, the CV SYSTEM START date and time will be displayed. When the input file contains interval based statistics, the INTERVAL START date and time will be displayed.

- **CV JOB NAME**
  Specifies the starting job name of Central Version which collected the statistics.

- **CV NUMBER**
  Specifies the number of Central Version which collected the statistics.

- **IP ADDRESS**
  IPV4 or IPV6 (dotted string format) of the originating IP Address for TCP/IP transactions.

- **NUMBER TASK EXECUTIONS**
  Indicates the total number of tasks executed within the actual time frame of the report. This label appears on SREPORTs 005, 006, 007, 008, and 009.

- **NUMBER OF TRANSACTIONS**
  Indicates the total number of transactions executed within the actual time frame of the report. This label appears on SREPORTs 010, 011, and 021.

- **NUMBER PROGRAMS CALLED**
  Indicates the number of programs called.

- **NUMBER PROGRAMS LOADED**
  Indicates the number of programs loaded from disk (either a load area or a load/core-image library).
- **NUMBER TERMINAL READS**
  Indicates the number of terminal reads performed.

- **NUMBER TERMINAL WRITES**
  Indicates the number of terminal writes performed.

- **NUMBER TERMINAL ERRORS**
  Indicates the number of terminal I/O errors.

- **NUMBER GETSTG REQUESTS**
  Indicates the number of requests to acquire storage dynamically from storage pool.

- **NUMBER GETSCR REQUESTS**
  Indicates the number of requests to retrieve scratch records from the DDLDCSCR area.

- **NUMBER PUTSCR REQUESTS**
  Indicates the number of requests to place scratch records in the DDLDCSCR area.

- **NUMBER DELSCR REQUESTS**
  Indicates the number of requests to delete scratch records from the DDLDCSCR area.

- **NUMBER GETQUE REQUESTS**
  Indicates the number of requests to get queue records from the DDLDCRUN area.

- **NUMBER PUTQUE REQUESTS**
  Indicates the number of requests to store or replace queue records in the DDLDCRUN area.

- **NUMBER DELQUE REQUESTS**
  Indicates the number of requests to delete queue records from the DDLDCRUN area.

- **NUMBER GETTIME REQUESTS**
  Indicates the number of requests for the date and time of day.

- **NUMBER SETTIME REQUESTS**
  Indicates the number of requests to define an event that is to occur after a specified time interval.

- **NUMBER DB SERVICE RQSTS**
  Indicates the number of times requests for database services (for example, OBTAIN record-name) were issued by the task.
  For LRF and SQL programs, this value should be less than or equal to the number of CALLS TO DBMS. You can use these values to evaluate how efficiently the LRF or SQL path extracts data. For example, a program OBTAIN logical-record command increments NUMBER DB SERVICE RQSTS by one, but may greatly increment the CALLS TO DBMS value, especially if an area sweep occurs due to the NULL SELECT clause.

- **NUMBER PAGES READ**
  Indicates the number of database pages read from disk.

- **TASK USER MODE TIME**
  Indicates the amount of CPU time (in ten-thousandths seconds) spent executing user code. DC/UCF collects user-mode time statistics if enabled in the STATISTICS TASK clause of the SYSTEM system generation statement.
- **TASK SYSTEM MODE TIME**
  Indicates the amount of CPU time (in ten-thousandths seconds) spent performing DC/UCF services for the task. DC/UCF collects system-mode time statistics if enabled in the STATISTICS TASK clause of the SYSTEM system generation statement.

- **TASK WAIT TIME**
  Indicates the amount of time spent on I/O requests and waiting for other system resources. The value is the difference between wall-clock and CPU time for the task.

- **NUMBER PAGES WRITTEN**
  Indicates the number of database pages physically written to disk for the transaction. A page can be updated several times before it is actually written back to the database.

- **NUMBER PAGES REQUESTED**
  Indicates the number of database pages requested by IDMSDBMS (including pages found in a buffer). A page request does not result in a page read if the page is in the buffer pool.
  Interpretation: The ratio of NUMBER PAGES REQUESTED to NUMBER PAGES READ is the buffer utilization ratio. The buffer utilization ratio measures the effectiveness of the buffer-pool size and design of the database (for example, CALC and VIA clustering). The higher the ratio the better.
  Ratios consistently below 2.0 indicate that processing is random or that the buffer-pool size is too small.
  The buffer utilization ratio may be artificially high for transactions that keep locks, due to the nature of the internal locking mechanism. IDMSDBMS cannot hold a buffer while requesting a lock; therefore, when locks are kept, IDMSDBMS must free and request a page each time a record is requested.

- **NUMBER CALC RECS NO OFLOW**
  Indicates the number of CALC records stored on the target page.

- **NUMBER CALC RECS OOFLOW**
  Indicates the number of CALC records not stored on the target page.
  Interpretation: The ratio of NO OVERFLOW records to the total number of CALC records stored is the CALC cluster ratio. Ideally, the ratio should be 1, which indicates no overflow. Ratios less than 1 or less than the norm indicate space utilization is getting high and the database should be tuned.

- **NUMBER VIA RECS NO OFLOW**
  Indicates the number of VIA and/or DIRECT records stored on the target page.

- **NUMBER VIA RECS OOFLOW**
  Indicates the number of VIA and or DIRECT records not stored on the target page.
  Interpretation: The ratio of NO OVERFLOW records to the total number of VIA records stored is the VIA cluster ratio. Ideally, the ratio should be 1, which indicates no overflow. A value less than 1 or less than the norm indicates very large data clusters, high utilization of space, or small page size.

- **NUMBER RECORDS REQUESTED**
  Indicates the number of records requested by the DBMS.
  Interpretation: The ratio of NUMBER RECORDS REQUESTED to NUMBER PAGES REQUESTED is the space management ratio. The space management ratio measures how well space is allocated (for example, VIA options, CALC distribution, and buffering). The higher the ratio the better. Ratios less than 4 or less than the norm indicate the size of the buffer should be increased and database tuning should be performed.
The space management ratio may be artificially high for transactions that keep locks, due to the nature of the internal locking mechanism. IDMSDBMS cannot hold a buffer while requesting a lock; therefore, when locks are kept, IDMSDBMS must free and request a page each time a record is requested.

- **NUMBER RECORDS CURRENT OF RU**
  Indicates the number of records made current of the transaction. Interpretation: The ratio of NUMBER RECORDS REQUESTED to NUMBER RECORDS CURRENT OR RU is the *effectiveness ratio*. The effectiveness ratio measures how much work the CA IDMS/DB has to do to find the requested record. The lower the ratio the better. If the ratio is high, examine set options (for example, sorted order or next pointers only) for appropriateness. If the options are correct, examine the program logic for accurate use of currency.

- **NUMBER FRAGMENTS STORED**
  Indicates the number of noncontiguous segments stored for variable-length records.

- **NUMBER RECORDS RELOCATED**
  Indicates the number of records relocated from the home page.

- **NUMBER CALLS TO DBMS**
  Indicates the number of calls to the database management system.

  **Note:** Execution of each navigational DML request involves one call; execution of each logical record facility (LRF) or SQL request typically involves multiple calls.

- **NUMBER SR8 SPLITS**
  Indicates the number of SR8 splits.

- **NUMBER SR8 SPAWNS**
  Indicates the number of SR8 spawns.

- **NUMBER ORPHAN ADOPT**
  Indicates count of Index members or SR8s whose up-level pointers were corrected to point to the actual SR8 in which they appear.

- **NUMBER BTREE SEARCHES**
  Indicates number of Btree index probes.

- **NUMBER SQL COMMANDS**
  Indicates the number of SQL commands executed.

- **NUMBER TUPLES FETCHED**
  Indicates number of tuples FETCHed.

- **NUMBER ROWS INSERTED**
  Indicates the number of rows INSERTed.

- **NUMBER ROWS UPDATED**
  Indicates the number of rows UPDATed.
- **NUMBER ROWS DELETED**
  Indicates the number of rows DELETEd.

- **NUMBER TOTAL LOCKS**
  Indicates the number of all locks acquired and released by all completed transactions. This is NOT a count of locks currently held.

- **NUMBER SR8 STORES**
  Indicates count of SR8s created.

- **NUMBER SR8 ERASES**
  Indicates count of SR8s erased.

- **NUMBER SR7 STORES**
  Indicates count of SR7s created.

- **NUMBER SR7 ERASES**
  Indicates count of SR7s erased.

- **NUMBER SORTS**
  Indicates the number of SQL sorts performed.

- **NUMBER TUPLES SORTED**
  Indicates the number of rows participating in all sorts.

- **NUMBER AM RECOMPILES**
  Indicates the number of automatic access module recompilations.

- **TASK SYSTEM MODE CPU**
  Indicates the amount of CPU time (in microseconds) spent in executing system code.

- **TASK USER MODE CPU**
  Indicates the amount of CPU time (in microseconds) spent in executing user code.

- **TASK ZIIP on ZIIP CPU**
  Indicates the amount of CPU time (in microseconds) spent on zIIP.

- **TASK ZIIP on CP CPU**
  Indicates the amount of CPU time (in microseconds) spent on CP, while qualified for zIIP.

- **TASK TOTAL TCB CPU TIME**
  Indicates the amount of CPU time (in microseconds) spent on CP in either system mode or user mode.

- **ENCLAVE CPU**
  Indicates the amount of CPU time (in microseconds) spent on CP or zIIP, while qualified for zIIP.
Transaction Statistics Reports

Transaction statistics record resource usage by transaction (that is, across tasks). At run time, IDMS DC accumulates transaction statistics when requested to do so by a user program.

The program initiates collection with a BIND TRANSACTION STATISTICS statement and terminates collection with an END TRANSACTION STATISTICS statement. The END TRANSACTION STATISTICS and ACCEPT TRANSACTION STATISTICS statements write statistics to the DC/UCF log file.

- Summary of Transaction Statistics Reports (see page 241)
- Field Descriptions (see page 243)

Summary of Transaction Statistics Reports

Three statistics reports summarize transaction activity:

<table>
<thead>
<tr>
<th>SREPORT 010</th>
<th>Summarizes transaction statistics by user identifier.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SREPORT 011</td>
<td>Summarizes transaction statistics by logical terminal identifier.</td>
</tr>
<tr>
<td>SREPORT 021</td>
<td>Summarizes transaction statistics by dialog.</td>
</tr>
<tr>
<td>SREPORT 022</td>
<td>Summarizes transaction statistics by IP Address</td>
</tr>
</tbody>
</table>

Sample SREPORT 010:

```
REPORT NO. 10                      IDMS-DC TRANSACTION STATISTICS BY USER ID Rnn.n
SELECTED FROM:  yyddd hh:mm       TO:  yyddd hh:mm
CV SYSTEM START:  yyddd hh:mm     TO:  yyddd hh:mm
CV JOB NAME:  jobname             CV NUMBER:  nnn
USER ID:  CMH
NUMBER PROGRAMS CALLED  484       NUMBER PROGRAMS LOADED  20
NUMBER TERMINAL READS  22         NUMBER TERMINAL WRITES  22
NUMBER TERMINAL ERRORS 0          NUMBER GETSBG REQUESTS  1,200
NUMBER GETSCR REQUESTS  61         NUMBER PUTSCR REQUESTS  71
NUMBER DELSCR REQUESTS  19         NUMBER GETQUE REQUESTS  48
NUMBER PUTQUE REQUESTS  33         NUMBER DELQUE REQUESTS  11
NUMBER GETTIME REQUESTS 313        NUMBER SETTIME REQUESTS  0
NUMBER DB SERVICE REQUESTS 42      NUMBER PAGES READ  180
TASK USER MODE TIME  .8382 TASK SYSTEM MODE TIME  2.7998
TASK WAIT TIME  18.2473
NUMBER PAGES WRITTEN  15          NUMBER PAGES REQUESTED  2,556
NUMBER VIA RECS NO OFLOW  0         NUMBER VIA RECS OFLOW  0
NUMBER CALC RECS NO OFLOW  11       NUMBER RECORDS CURRENT OF RU  920
NUMBER RECS NO OFLOW  33          NUMBER RECORDS RELOCATED  0
NUMBER RECORDS REQUESTED  2,737    NUMBER RECORDS CURRENT OF RU  7,466
NUMBER FRAGMENTS STORED  0          NUMBER LOCKS ACQUIRED  56
NUMBER SRB Splits  56            NUMBER SRB STORES  44
NUMBER SRB SPAWS  4              NUMBER SRB ERASES  44
NUMBER ORPHAN ADOPT  18           NUMBER SRB STORES  2
NUMBER BTREE SEARCHES  381        NUMBER SRB ERASES  2
NUMBER SQL COMMANDS  27           NUMBER SQL SORTS  1
NUMBER TUPLES FETCHED  785        NUMBER TUPLES SORTED  728
NUMBER ROWS INSERTED  0            NUMBER AM RECOMPILES  0
NUMBER ROWS UPDATED  0             NUMBER SRB STORES  0
NUMBER ROWS DELETED  0
```
Sample SREPORT 011:

REPORT NO. 11  IDMS-DC TRANSACTION STATISTICS BY LTERM ID  Rnn.n  mm/dd/yy PAGE 13

SELECTED FROM:  yyddd hh:mm  TO:  yyddd hh:mm
CV SYSTEM START:  yyddd hh:mm  T0:  yyddd hh:mm
CV JOB NAME:  jobname
CV NUMBER:  nnn

LTERM ID:  LT12022

NUMBER PROGRAMS CALLED  484  NUMBER PROGRAMS LOADED  20
NUMBER TERMINAL READS  22  NUMBER TERMINAL WRITES  22
NUMBER TERMINAL ERRORS  0  NUMBER GETSBG REQUESTS  1,280
NUMBER GETSCR REQUESTS  61  NUMBER PUTSCR REQUESTS  71
NUMBER DELSCR REQUESTS  19  NUMBER GETQUE REQUESTS  48
NUMBER PUTQUE REQUESTS  33  NUMBER DELQUE REQUESTS  11
NUMBER GETTIME REQUESTS  313  NUMBER SETTIME REQUESTS  0
NUMBER DB SERVICE ROSTS  42  NUMBER PAGES READ  180
TASK USER MODE TIME  .8382  TASK SYSTEM MODE TIME  2.7998
TASK WAIT TIME  18.2473
NUMBER PAGES WRITTEN  15  NUMBER PAGES REQUESTED  2,556
NUMBER CALC RECS NO OFLOW  0  NUMBER CALC RECS OFLOW  0
NUMBER VIA RECS NO OFLOW  33  NUMBER VIA RECS OFLOW  0
NUMBER RECORDS REQUESTED  2,737  NUMBER RECORDS CURRENT OF RU  920
NUMBER FRAGMENTS STORED  0  NUMBER RECORDS RELOCATED  0
NUMBER CALLS TO DBMS  581  TOTAL LOCKS ACQUIRED  7,466
NUMBER SR9 Splits  56  NUMBER SR9 STORES  44
NUMBER SR8 Splits  4  NUMBER SR8 ERASES  44
NUMBER ORPHAN ADOPIT  18  NUMBER SR8 STORES  44
NUMBER BTREE SEARCHES  381  NUMBER SR7 ERASES  2
NUMBER SQL COMMANDS  27  NUMBER SQL SORTS  1
NUMBER TUPLES FETCHED  728  NUMBER TUPLES SORTED  728
NUMBER ROWS INSERTED  0  NUMBER AM RECOMPILES  0
NUMBER ROWS UPDATED  0  NUMBER ROWS DELETED  0

TOTAL TASK TCB CPU  17.487940  TASK ZIIP ON CP CPU  .000000
TASK ZIP ON ZIIP CPU  .000800  TASK USER MODE CPU  .000000
TOTAL TASK TCB CPU  17.487940  ENCLAVE CPU  .000000

Sample SREPORT 021:

REPORT NO. 21  IDMS-DC TRANSACTION STATISTICS BY DIALOG  Rnn.n  mm/dd/yy PAGE 7

SELECTED FROM:  yyddd hh:mm  TO:  yyddd hh:mm
CV SYSTEM START:  yyddd hh:mm  T0:  yyddd hh:mm
CV JOB NAME:  jobname

DIALOG:  ASRDFD5

NUMBER PROGRAMS CALLED  484  NUMBER PROGRAMS LOADED  20
NUMBER TERMINAL READS  22  NUMBER TERMINAL WRITES  22
NUMBER TERMINAL ERRORS  0  NUMBER GETSBG REQUESTS  1,280
NUMBER GETSCR REQUESTS  61  NUMBER PUTSCR REQUESTS  71
NUMBER DELSCR REQUESTS  19  NUMBER GETQUE REQUESTS  48
NUMBER PUTQUE REQUESTS  33  NUMBER DELQUE REQUESTS  11
NUMBER GETTIME REQUESTS  313  NUMBER SETTIME REQUESTS  0
NUMBER DB SERVICE ROSTS  42  NUMBER PAGES READ  180
TASK USER MODE TIME  .8382  TASK SYSTEM MODE TIME  2.7998
TASK WAIT TIME  18.2473
NUMBER PAGES WRITTEN  15  NUMBER PAGES REQUESTED  2,556
NUMBER CALC RECS NO OFLOW  0  NUMBER CALC RECS OFLOW  0
NUMBER VIA RECS NO OFLOW  33  NUMBER VIA RECS OFLOW  0
NUMBER RECORDS REQUESTED  2,737  NUMBER RECORDS CURRENT OF RU  920
NUMBER FRAGMENTS STORED  0  NUMBER RECORDS RELOCATED  0
NUMBER CALLS TO DBMS  581  TOTAL LOCKS ACQUIRED  7,466
NUMBER SR9 Splits  56  NUMBER SR9 STORES  44
NUMBER SR8 Splits  18  NUMBER SR8 STORES  44

TOTAL TASK TCB CPU  17.487940  TASK ZIIP ON CP CPU  .000000
TASK ZIP ON ZIIP CPU  .000800  TASK USER MODE CPU  .000000
TOTAL TASK TCB CPU  17.487940  ENCLAVE CPU  .000000
### Field Descriptions

SREPORTs 010, 011, 021 and 022 have fields in common with task and ERUS reports; for a description of each field, see [Task and External Request Unit Service (ERUS) Statistics Reports](https://docops.ca.com/display/IDMS/Task+and+External+Request+Unit+Service+(ERUS)+Statistics+Reports+v19_Next).
CA ADS Dialog Statistics

Contents

- Field Descriptions (see page 246)

Overview

If CA ADS dialog and transaction statistics are enabled, statistics are collected for all or selected dialogs that execute during an application program. DC/UCF collects the following types of dialog statistics each time a dialog issues a control command:

- Statistics for explicitly coded control commands issued by the dialog
- Statistics for implicitly coded control commands issued by the ADS runtime system on behalf of the dialog
- Dialog execution statistics
- Statistics for record buffer block (RBB) usage

Statistics are written to the system log when the number of statistics accumulations equals the checkpoint interval established at system generation and when the application terminates.

⚠️ **Note:** For more information on collecting these statistics, see the CA ADS Reference section and Administering section.

⚠️ **Note:** Statistics for dialogs executed in batch mode can be collected in a separate log file associated with the batch dialog. For more information, see the CA ADS Batch Using section.

Summary of ADS Statistics Reports

Three statistics reports summarize dialog runtime activity:

<table>
<thead>
<tr>
<th>SREPORT 018</th>
<th>Summarizes CA ADS dialog statistics by user.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SREPORT 019</td>
<td>Summarizes CA ADS dialog statistics by dialog and version number.</td>
</tr>
<tr>
<td>SREPORT 020</td>
<td>Summarizes CA ADS dialog statistics by logical terminal.</td>
</tr>
</tbody>
</table>

Sample SREPORT 018:
Sample SREPORT 019:

Report No. 019  
ADS Statistics By Dialog And Version Number: Rnn.n  
mm/dd/yy  
Page 8

DIAL NAME: ADSOAFNC  
VERSION NUMBER: 1

Date: yyyyyy Time: hh:mm  
User ID: SMT  
Date Bind: yyyyyy Time Bind: hh:mm  
LTERM ID: LT12011

Display Command: 21 Display Continu: 21 Invoices: 3 Link To Dialogs: 18  
Links To Program: 18 Returns: 0 Return Continue: 0 Transfers: 18  
Leave Ads: 1 Leave Application: 0 Aborts: 0 Impl Displays: 0  
Impl Invoe: 0 Impl Link Dlgs: 0 Impl Link Poms: 0 Impl Returns: 0  
Impl Ret Cont: 0 Impl Transfers: 1 Impl Leave Ads: 0 Impl Leave Poms: 0  
Expl Get Scrs: 4 Expl Put Scrs: 3 Expl Del Scrs: 26 Write Print Reqs: 0  
Put New Details: 0 Put Cur Details: 0 Get Details: 0 Size Of Fdb: 23,808  
Size Of Vob: 836 Highest Link Lev: 1 Lowest Link Lev: 1 Rbb Put To Scr: 0  
Rbb Stg Hi Mark: 3,176 Rbb Free Hi: 908 Rbb Stg Low Mk: 3,176 Rbb Free Low: 908  
Most Rbb Acq: 304 Least Rbb Acq: 304 Hicount Rbb Use: 1 Locount Rbb Use: 1

**** DIALOG TOTAL ****

Display Command: 21 Display Continu: 21 Invoices: 3 Link To Dialogs: 18  
Links To Program: 18 Returns: 0 Return Continue: 0 Transfers: 18  
Leave Ads: 0 Leave Application: 0 Aborts: 0 Impl Displays: 0  
Impl Invoe: 0 Impl Link Dlgs: 0 Impl Link Poms: 0 Impl Returns: 0  
Impl Ret Cont: 0 Impl Transfers: 1 Impl Leave Ads: 0 Impl Leave Poms: 0  
Expl Get Scrs: 4 Expl Put Scrs: 3 Expl Del Scrs: 26 Write Print Reqs: 0  
Put New Details: 0 Put Cur Details: 0 Get Details: 0 Record Count: 2
### Field Descriptions

A description of the fields common to SREPORTs 018 through 020 follows:

- **DATE**
  
  Specifies the date the statistics were written to the system log; the date is in Julian form, `yyddd`, where `yy` is the last two digits of the year and `ddd` is the Julian day.

- **TIME**
  
  Specifies the time the statistics were written to the system log.

- **DATE BIND**
  
  Specifies the date the TSB BIND command was issued; the date is in Julian form, `yyddd`, where `yy` is the last two digits of the year and `ddd` is the Julian day.
• **TIME BIND**  
  Specifies the time the TSB BIND command was issued.

• **DISPLAY COMMAND**  
  Specifies the number of explicit DISPLAY commands issued by the dialog.

• **DISPLAY CONTINU**  
  Specifies the number of explicit DISPLAY CONTINUE commands issued by the dialog.

• **INVOKES**  
  Specifies the number of explicit INVOKE commands issued by the dialog.

• **LINKS TO DIALOGS**  
  Specifies the number of explicit LINK TO DIALOG commands issued by the dialog.

• **LINKS TO PROGRAM**  
  Specifies the number of explicit LINK TO PROGRAM commands issued by the dialog.

• **RETURNS**  
  Specifies the number of explicit RETURN commands issued by the dialog.

• **RETURN CONTINUE**  
  Specifies the number of explicit RETURN CONTINUE commands issued by the dialog.

• **TRANSFERS**  
  Specifies the number of explicit TRANSFER commands issued by the dialog.

• **LEAVE ADS**  
  Specifies the number of explicit LEAVE ADS commands issued by the dialog.

• **LEAVE APPLICATN**  
  Indicates number of explicit LEAVE APPLICATION commands issued by the dialog.

• **ABORTS**  
  Indicates the number of explicit ABORT commands issued by the dialog.

• **IMPL DISPLAYS**  
  Indicates the number of implicitly generated DISPLAY commands issued during the application.

• **IMPL INVOKE**  
  Specifies the number of implicitly generated INVOKE commands issued during the application.

• **IMPL LINK DLGS**  
  Indicates the number of implicitly generated LINK TO DIALOG commands issued during the application.

• **IMPL LINK PGMS**  
  Indicates the number of implicitly generated LINK TO PROGRAM commands issued during the application.

• **IMPL RETURNS**  
  Indicates the number of implicitly generated RETURN commands issued during the application.
• **IMPL RET CONT**
  Indicates the number of implicitly generated RETURN CONTINUE commands issued during the application.

• **IMPL TRANSFERS**
  Indicates the number of implicitly generated TRANSFER commands issued during the application.

• **IMPL LEAVE ADS**
  Indicates the number of implicitly generated LEAVE ADS commands issued during the application.

• **IMPL LEAVE PGMS**
  Indicates the number of implicitly generated LEAVE APPLICATION commands issued during the application.

• **IMPL ABORTS**
  Indicates the number of implicitly generated ABORT commands issued during the application.

• **PREMAP PROCESS**
  Indicates the number of dialog premap process executions.

• **RESPONSE PROCES**
  Indicates the number of dialog response process executions.

• **STAT ACCUM CALL**
  Indicates the number of calls to IDMS DC to accumulate dialog transaction statistics.

• **EXPL GET SCRS**
  Specifies the number of explicit dialog requests to retrieve scratch records from the DDLDCSCR area.

• **EXPL PUT SCRS**
  Specifies the number of explicit dialog requests to place scratch records in the DDLDCSCR area.

• **EXPL DEL SCRS**
  Specifies the number of explicit dialog requests to delete scratch records from the DDLDCSCR area.

• **WRTE PRINT REQS**
  Indicates the number of dialog WRITE PRINTER commands.

• **PUT NEW DETAILS**
  Indicates the number of dialog PUT NEW DETAIL commands.

• **PUT CUR DETAILS**
  Indicates the number of dialog PUT CURRENT DETAIL commands.

• **GET DETAILS**
  Indicates the number of dialog GET DETAIL commands.

• **SIZE OF FDB**
  Specifies the size of the fixed dialog block in bytes. The FDB is the dialog load module generated by the dialog generator.
• SIZE OF VDB
  Specifies the size of the variable dialog block in bytes. The VDB is created dynamically for
  the issuing dialog at runtime and contains runtime variable information on the dialog.

• HIGHEST LNK LEV
  Indicates the highest level within an application thread at which the dialog was executed. The
  value for the highest link level is zero.

• LOWEST LNK LEVL
  Indicates the lowest level within an application thread at which the dialog was executed. The
  higher the value the lower the link level.

• RBB PUT TO SCR
  Indicates the number of record buffer blocks placed as scratch records in the DDLDCSCR area.

• RBB STG HI MARK
  Indicates the most record buffer block storage allocated for all dialogs.

• RBB FREE HI
  Indicates the amount of free record buffer block space when the most storage is allocated for all
  dialogs.

• RBB STG LOW MK
  Indicates the least record buffer block storage allocated for all dialogs.

• RBB FREE LOW
  Indicates the amount of free space in the record buffer block when the least storage is used.

• MOST RBB ACQ
  Indicates the most space acquired in the record buffer block for the dialog.

• LEAST RBB ACQ
  Indicates the least space acquired in the record buffer block for the dialog.

• HICOUNT RBB USE
  Indicates the highest number of record buffer blocks used.

• LOCOUNT RBB USE
  Indicates the lowest number of record buffer blocks used.

• RECORD COUNT
  (Total summary only) Indicates the total number of dialog statistics records written.

Note: For a detailed explanation of the CA ADS commands, see the CA ADS Reference section.
Histogram Report

Histograms show statistical data for events (for example, program loads into the program pool) in terms of frequency of occurrence within predefined value ranges (for example, number of program loads smaller than 250 bytes, between 250 to 500 bytes, and so on).

Note: For a detailed discussion of histogram categories, classes, default values, and the #HSTDEF macro that overrides default values, see Administrating section.

When Statistics Are Written to Log

Like system statistics, histograms are written to the DC/UCF log file at the following times:

- At normal system shutdown
- At the statistics interval established at system generation by the STATISTICS parameter of the SYSTEM statement. The statistics interval can be varied at run time with the DCMT VARY STATISTICS command.
- Upon explicit request by means of a DCMT WRITE STATISTICS command.

Summary of Histogram Record Subtypes

SREPORT 001, IDMS Statistics-Histogram Report, produces a histogram for each of 25 histogram record subtypes. The following Histogram Records table lists each histogram record subtype by identifier and descriptor.

<table>
<thead>
<tr>
<th>Subtype Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>PUT JOURNAL request size</td>
</tr>
<tr>
<td>02</td>
<td>Program size in bytes</td>
</tr>
<tr>
<td>03</td>
<td>Queue record sizes</td>
</tr>
<tr>
<td>04</td>
<td>Scratch record sizes</td>
</tr>
<tr>
<td>05</td>
<td>User storage size</td>
</tr>
<tr>
<td>06</td>
<td>Total size of GET STORAGE requests</td>
</tr>
<tr>
<td>07</td>
<td>Programs called by task</td>
</tr>
<tr>
<td>08</td>
<td>Programs loaded by task</td>
</tr>
<tr>
<td>09</td>
<td>Terminal reads by task</td>
</tr>
<tr>
<td>10</td>
<td>Terminal writes by task</td>
</tr>
<tr>
<td>11</td>
<td>Terminal errors by task</td>
</tr>
<tr>
<td>12</td>
<td>Number of GET STORAGE requests by task</td>
</tr>
<tr>
<td>13</td>
<td>Number of GET SCRATCH requests by task</td>
</tr>
<tr>
<td>Subtype Identifier</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>14</td>
<td>Number of PUT SCRATCH requests by task</td>
</tr>
<tr>
<td>15</td>
<td>Number of DELETE SCRATCH requests by task</td>
</tr>
<tr>
<td>16</td>
<td>Number of GET QUEUE requests by task</td>
</tr>
<tr>
<td>17</td>
<td>Number of PUT QUEUE requests by task</td>
</tr>
<tr>
<td>18</td>
<td>Number of DELETE QUEUE requests by task</td>
</tr>
<tr>
<td>19</td>
<td>Number of GET TIME requests by task</td>
</tr>
<tr>
<td>20</td>
<td>Number of SET TIME requests by task</td>
</tr>
<tr>
<td>21</td>
<td>Number of database calls by task</td>
</tr>
<tr>
<td>22</td>
<td>Number of high stacks by task</td>
</tr>
<tr>
<td>23</td>
<td>Total time (in ten-thousandths seconds) spent in user mode by task</td>
</tr>
<tr>
<td>24</td>
<td>Total time (in ten-thousandths seconds) spent in system mode by task</td>
</tr>
<tr>
<td>25</td>
<td>Total wait time by task</td>
</tr>
</tbody>
</table>

The following report illustrates a page of a sample histogram report; it contains a histogram of the number of programs loaded by task (subtype 8).

Sample SREPORT 001:

```
REPORT NO. 01          IDMS STATISTICS - HISTOGRAM REPORT  Rnn.n          mm/dd/yy PAGE 8

SELECTED FROM:  yyddd hh:mm       TO:  yyddd hh:mm
ACTUAL:        yyddd hh:mm       TO:  yyddd hh:mm

SUB-TYPE      Gw-DESCRIPTION BINS LOW END INCREMENT LOW VALUE HIGH VALUE     DATE       TIME

HISTOGRAM FOR SUB-TYPE 08 PROGRAMS LOADED BY TASK 10 1 2 292 0   yyddd hh:mm

BELOW        1  292
1 T0         2  38
3 T0         4  7
5 T0         6  3
7 T0         8  0
9 T0         10  1
11 T0        12  0
13 T0        14  0
15 T0        16  1
17 T0        18  0
19 T0        20  0
21 AND ABOVE... 0

TOTAL:  342
```
Field Descriptions

A description of the fields for the 25 histograms produced by SREPORT 001 follows:

- **SELECTED FROM/TO**
  Specifies the time period specified on the USE 'SREPORT 000' parameter where SELECTED FROM specifies the beginning date and time and SELECTED TO specifies the ending date and time. The date is in Julian form, \textit{yyddd}, where \textit{yy} is the last two digits of the year and \textit{ddd} is the day. The time is in \textit{hh:mm} form, where \textit{hh} is hours based on a 24-hour clock and \textit{mm} is minutes.

- **ACTUAL/TO**
  Specifies the actual time period for the report generated. If the USE 'SREPORT 000' parameter does not specify a time range, ACTUAL and TO represent the time range for the entire input archive file or input file created by SREPORT 099.

- **SUB-TYPE**
  Specifies the identifier of the histogram record subtype.

- **GW-DESCRIPTION**
  Specifies the descriptor for the histogram record subtype.

- **BINS**
  Indicates the number of bins in the histogram; a bin has finite limits. The default is 10.

- **LOW END**
  Indicates the low end of the range of values of the histogram.

- **INCREMENT**
  Specifies the increment added to range values for successive bins.

- **LOW VALUE**
  Indicates the number of occurrences below the value specified for LOW END.

- **HIGH VALUE**
  Indicates the number of occurrences greater than the high end of the histogram's range of values.

- **STAT RECORD DATE**
  Specifies the date, in Julian form, the histogram record was written to the DC/UCF log file.

- **STAT RECORD TIME**
  Specifies the time, in \textit{hh:mm} form, the histogram was written to the DC/UCF log file.

- **TOTAL**
  Indicates the total of the bin occurrences.

Record Summary Statistics Report

Contents

- Contents of SREPORT 017 (see page 254)
Overview

DC/UCF logs 35 types of statistics records to the system log file. Particular statistic reports require specific record types. For example, statistics record 30 provides the statistics summarized in SREPORT 013, IDMS DC Program Summary.

Summary of Statistics Records

The following table lists each type of statistics record. The record ID numbers correspond to numbers shown in column 1 of SREPORT 017, Summary of Records Read:

<table>
<thead>
<tr>
<th>Record Number</th>
<th>Statistics Record Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>System startup marker record</td>
</tr>
<tr>
<td>01/01</td>
<td>Histogram record of PUT JOURNAL DML requests</td>
</tr>
<tr>
<td>01/02</td>
<td>Histogram record of program size in pages</td>
</tr>
<tr>
<td>01/03</td>
<td>Histogram record of queue record size</td>
</tr>
<tr>
<td>01/04</td>
<td>Histogram record of scratch record size</td>
</tr>
<tr>
<td>01/05</td>
<td>Histogram record of user storage size</td>
</tr>
<tr>
<td>01/06</td>
<td>Histogram record of total requested storage</td>
</tr>
<tr>
<td>01/07</td>
<td>Histogram record of programs called by task</td>
</tr>
<tr>
<td>01/08</td>
<td>Histogram record of programs loaded by task</td>
</tr>
<tr>
<td>01/09</td>
<td>Histogram record of terminal reads by task</td>
</tr>
<tr>
<td>01/0A</td>
<td>Histogram record of terminal writes by task</td>
</tr>
<tr>
<td>01/0B</td>
<td>Histogram record of terminal errors by task</td>
</tr>
<tr>
<td>01/0C</td>
<td>Histogram record of storage requests by task</td>
</tr>
<tr>
<td>01/0D</td>
<td>Histogram record of GET SCRATCH requests by task</td>
</tr>
<tr>
<td>01/0E</td>
<td>Histogram record of PUT SCRATCH requests by task</td>
</tr>
<tr>
<td>01/0F</td>
<td>Histogram record of DELETE SCRATCH requests by task</td>
</tr>
<tr>
<td>01/10</td>
<td>Histogram record of GET QUEUE requests by task</td>
</tr>
<tr>
<td>01/11</td>
<td>Histogram record of PUT QUEUE requests by task</td>
</tr>
<tr>
<td>01/12</td>
<td>Histogram record of DELETE QUEUE requests by task</td>
</tr>
<tr>
<td>01/13</td>
<td>Histogram record of GET TIME requests by task</td>
</tr>
<tr>
<td>01/14</td>
<td>Histogram record of SET TIME requests by task</td>
</tr>
<tr>
<td>01/15</td>
<td>Histogram record of database calls by task</td>
</tr>
<tr>
<td>01/16</td>
<td>Histogram record of high stacks by task</td>
</tr>
<tr>
<td>01/17</td>
<td>Histogram record of user time by task</td>
</tr>
<tr>
<td>01/18</td>
<td>Histogram record of system time by task</td>
</tr>
<tr>
<td>01/19</td>
<td>Histogram record of wait time by task</td>
</tr>
<tr>
<td>01/1A</td>
<td>Histogram record of high RCE count</td>
</tr>
</tbody>
</table>
Contents of SREPORT 017

SREPORT 017, the Summary of Records Read report, itemizes the number of each type of statistics record written to the log file within a defined period of time. Systems administrators can use SREPORT 017 to determine which statistics reports can be run based upon available statistics records.

Sample SREPORT 017:

<table>
<thead>
<tr>
<th>Record Number</th>
<th>Statistics Record Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/1B</td>
<td>Histogram record of high RLE count</td>
</tr>
<tr>
<td>01/1C</td>
<td>Histogram record of high DPE count</td>
</tr>
<tr>
<td>07/01</td>
<td>Histogram record of response time by line</td>
</tr>
<tr>
<td>01</td>
<td>System-wide statistics record</td>
</tr>
<tr>
<td>02</td>
<td>Task statistics record</td>
</tr>
<tr>
<td>03</td>
<td>Transaction statistics record</td>
</tr>
<tr>
<td>04</td>
<td>Task code statistics record</td>
</tr>
<tr>
<td>05</td>
<td>Program statistics record</td>
</tr>
<tr>
<td>06</td>
<td>Queue statistics record</td>
</tr>
<tr>
<td>07</td>
<td>Line statistics record</td>
</tr>
<tr>
<td>08</td>
<td>Physical terminal statistics record</td>
</tr>
<tr>
<td>09</td>
<td>CA ADS dialog statistics record</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RECORD NO. 17</th>
<th>SUMMARY OF RECORDS READ Rnn.n mm/dd/yy PAGE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECTED FROM:</td>
<td>yddd hh:mm TO: yddd hh:mm</td>
</tr>
<tr>
<td>START UP</td>
<td>00</td>
</tr>
<tr>
<td>PUTJRL REQ SIZE</td>
<td>01/01</td>
</tr>
<tr>
<td>LOADED PGM SIZE</td>
<td>01/02</td>
</tr>
<tr>
<td>QUEUE RECORD SIZE</td>
<td>01/03</td>
</tr>
<tr>
<td>SCRATCH REC SIZE</td>
<td>01/04</td>
</tr>
<tr>
<td>USER STG SIZE</td>
<td>01/05</td>
</tr>
<tr>
<td>TOTAL GETSTG SIZE</td>
<td>01/06</td>
</tr>
<tr>
<td>PMS CALLED</td>
<td>01/07</td>
</tr>
<tr>
<td>PMS LOADED</td>
<td>01/08</td>
</tr>
<tr>
<td>TERMINAL READS</td>
<td>01/09</td>
</tr>
<tr>
<td>TERMINAL WRITES</td>
<td>01/10</td>
</tr>
<tr>
<td>TERMINAL ERRORS</td>
<td>01/11</td>
</tr>
<tr>
<td>GET STORAGES</td>
<td>01/12</td>
</tr>
<tr>
<td>GET SCRATCHES</td>
<td>01/13</td>
</tr>
<tr>
<td>GET TIMES</td>
<td>01/14</td>
</tr>
<tr>
<td>DB CALLS</td>
<td>01/15</td>
</tr>
<tr>
<td>STACK HIWATER</td>
<td>01/16</td>
</tr>
<tr>
<td>USER-MODE TIME</td>
<td>01/17</td>
</tr>
<tr>
<td>SYSTEM-MODE TIME</td>
<td>01/18</td>
</tr>
</tbody>
</table>
Field Descriptions

A description of the fields in the Summary of Records Read report follows:

- **SELECTED FROM/TO**
  Gives the time period specified on the USE 'SREPORT 000' parameter where SELECTED FROM represents the beginning date and time and SELECTED TO represents the ending date and time. The date is in Julian form, yyddd, where yy is the last two digits of the year and dddd is the day. The time is in hh:mm form, where hh is hours based on a 24-hour clock and mm is minutes.

- **ACTUAL/TO**
  Specifies the actual time period for the report generated. If the USE 'SREPORT 000' parameter does not specify a time range, ACTUAL and TO represent the time range for the entire input archive file or input file created by SREPORT 099.

- **COUNTS FOR STATISTICS RECORDS**
  Specifies the type of statistic record. The table above lists each type of statistics record, as defined in DSECT #STLDS; for more information, see the CA IDMS DSECT Reference section.

- **BELOW DATE/TIME RANGE**
  Indicates the number of records of each type logged to the log file before the date and time specified in the SELECTED FROM field.

- **BEFORE TRIGGER STARTUP RECORDS**
  Indicates the number of records of each type logged to the log file after the date and time specified in the SELECTED FROM field, but before the system startup marker record (type 00) corresponding to the session indicator specified on the USE 'SREPORT 000' parameter.

- **PASSED TO STATISTICS ROUTINES**
  Indicates the number of records processed by statistics routines.

- **DATES/TIMES OF 1ST AND LAST RECS OF EACH TYPE PASSED TO THE ROUTINES**
  Specifies the date and time of the first record of each type passed to a statistics routine and the date and time of the last record passed to a statistics routine. If no records were passed to a routine, the field is blank.
DC/UCF System Reports -- CREPORTS

CA IDMS/DC and CA IDMS UCF (DC/UCF) system reports describe the dictionary entities used to define the characteristics and components of an executable DC/UCF system. The DC/UCF system reports document information maintained in the DDLDML, DDLDCMSG, and DDLDCLOD areas of the dictionary. The records being reported on are defined to the dictionary through various CA IDMS system software components, such as the Data Dictionary Definition Language (DDDL) compiler and the system generation compiler.

For details about the structure of these records and how the records are defined to the dictionary, see the CA IDMS Dictionary Structure Reference section.

- Uses for DC/UCF System Reports (see page 256)
- Summary of CREPORTs (see page 257)
- Producing DC/UCF System Reports (see page 259)
- CA ADS Parameter Reports (CREPORTs 040 and 045) (see page 261)
- Defined Devices Report (CREPORT 029) (see page 264)
- Defined Messages Report (CREPORT 028) (see page 265)
- Destination Reports (CREPORTs 007 and 024) (see page 266)
- Load Area Report (CREPORT 050) (see page 267)
- Mapping Reports (CREPORTs 030 through 035) (see page 268)
- Nodes and Resource Table Reports (CREPORTS 043 and 044) (see page 271)
- Module Text to Card Utility (CREPORT 051) (see page 272)
- Module Text to File Utility (CREPORT 052) (see page 273)
- Network Description Reports (CREPORTs 001-003, 014-018) (see page 274)
- CA OLQ Reports (CREPORTs 041 and 046) (see page 280)
- Program Description Reports (CREPORTs 004 and 019) (see page 282)
- Queue Description Reports (CREPORTs 006, 022, and 023) (see page 285)
- SQL CACHE Reports (CREPORTs 047 and 048) (see page 286)
- Symbol Table Report (CREPORT 053) (see page 287)
- System Options Reports (CREPORTs 011 and 025) (see page 288)
- Task Description Reports (CREPORTs 005, 020, and 021) (see page 297)
- Builder Codes (see page 301)

Uses for DC/UCF System Reports

DC/UCF system reports can be used to:
Provide descriptions of executable systems and of systems that have been defined but not generated.
DC/UCF system object reports describe executable systems (that is, systems for which GENERATE has been issued in the system generation compile); these reports access dictionary object records. DC/UCF system source reports describe systems that have been defined to the dictionary but not yet generated; these reports access dictionary source records.

Note: For more information on source and object records, see the CA IDMS Administrating section.

Because the dictionary structure includes source and object records, you can modify the system definitions without affecting the runtime definitions. The system source reports can be used to review the proposed modifications before they are implemented. Once you are satisfied with the new configuration, you can update the runtime definitions by issuing a system generation GENERATE command for any system that you want to update.

- Review the screen/data field relationships for mapping operations.
- Monitor the contents of the load area.
- Monitor the messages defined to the DDLDCMSG area of the dictionary.

Summary of CREPORTs

The DC/UCF system reports are presented in alphabetical order, with the following exceptions:

- The Physical Terminal by Line and Logical Terminal by Physical Terminal reports (CREPORTs 016 and 018) are discussed under Network Description Reports (CREPORTs 001-003, 014-018) (see page 274).

- The Listing of Map reports (CREPORTs 032 through 035) are discussed under Mapping Reports (CREPORTs 030 through 035) (see page 268).

The following table lists the CREPORTs in the order of presentation in this section. For a list of CREPORTs by report number, see the CREPORT Listing table in Appendix D.

<table>
<thead>
<tr>
<th>CREPORT Number</th>
<th>CREPORT Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>040</td>
<td>ADS/OnLine Parameters Report (Object)</td>
</tr>
<tr>
<td>045</td>
<td>ADS/OnLine Parameters Report (Source)</td>
</tr>
<tr>
<td>029</td>
<td>Defined Devices</td>
</tr>
<tr>
<td>028</td>
<td>Defined Messages(1)</td>
</tr>
<tr>
<td>007</td>
<td>Destination Report (Object)</td>
</tr>
<tr>
<td>024</td>
<td>Destination Report (Source)</td>
</tr>
<tr>
<td>050</td>
<td>Load Area Report(1)</td>
</tr>
<tr>
<td>CREPORT Number</td>
<td>CREPORT Name</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>030</td>
<td>Map Record Indices</td>
</tr>
<tr>
<td>031</td>
<td>Map Field Indices</td>
</tr>
<tr>
<td>032</td>
<td>Listing of Maps by Panel</td>
</tr>
<tr>
<td>033</td>
<td>Listing of Maps</td>
</tr>
<tr>
<td>034</td>
<td>Listing of Maps by Record Name</td>
</tr>
<tr>
<td>035</td>
<td>Listing of Maps by Element Name</td>
</tr>
<tr>
<td>043</td>
<td>Listing of Nodes</td>
</tr>
<tr>
<td>044</td>
<td>Listing of Defined Resources</td>
</tr>
<tr>
<td>051</td>
<td>Module Text to Card Utility</td>
</tr>
<tr>
<td>052</td>
<td>Module Text to File Utility</td>
</tr>
<tr>
<td>001</td>
<td>Network Description by Line (Object)</td>
</tr>
<tr>
<td>014</td>
<td>Network Description by Line (Source)</td>
</tr>
<tr>
<td>002</td>
<td>Network Description by Physical Terminal (Object)</td>
</tr>
<tr>
<td>015</td>
<td>Network Description by Physical Terminal (Source)</td>
</tr>
<tr>
<td>003</td>
<td>Network Description by Logical Terminal (Object)</td>
</tr>
<tr>
<td>017</td>
<td>Network Description by Logical Terminal (Source)</td>
</tr>
<tr>
<td>016</td>
<td>Physical Terminals within Line (Source)</td>
</tr>
<tr>
<td>018</td>
<td>Logical Terminal by Physical Terminal (Source)</td>
</tr>
<tr>
<td>041</td>
<td>OLQ Report (Object)</td>
</tr>
<tr>
<td>046</td>
<td>OLQ Report (Source)</td>
</tr>
<tr>
<td>004</td>
<td>Program Description Report (Object)</td>
</tr>
<tr>
<td>019</td>
<td>Program Description Report (Source)</td>
</tr>
<tr>
<td>006</td>
<td>Queue Description Report (Object)</td>
</tr>
<tr>
<td>022</td>
<td>Queue Description Report (Source)</td>
</tr>
<tr>
<td>023</td>
<td>Queue Description within Task Report (Source)</td>
</tr>
<tr>
<td>047</td>
<td>SQL CACHE Report (Source)</td>
</tr>
<tr>
<td>048</td>
<td>SQL CACHE Report (Object)</td>
</tr>
<tr>
<td>053</td>
<td>Symbol Table Report</td>
</tr>
<tr>
<td>011</td>
<td>System Options Report (Object)</td>
</tr>
<tr>
<td>025</td>
<td>System Options Report (Source)</td>
</tr>
<tr>
<td>005</td>
<td>Task Description Report (Object)</td>
</tr>
<tr>
<td>020</td>
<td>Task Description Report (Source)</td>
</tr>
<tr>
<td>021</td>
<td>Task Description within Program Report (Source)</td>
</tr>
</tbody>
</table>

**Note:** Reports 28 and 50 must be run alone.
Produce DC/UCF System Reports

Contents

- Syntax (see page 259)
- Parameters (see page 259)
- Examples (see page 260)

A DC/UCF system report is produced by submitting a batch job that includes the standard JCL for the CA Culprit for CA IDMS report writer and report-specific control statements. JCL for z/OS, z/VSE, and z/VM operating systems is shown in Appendixes A through D, respectively.

Syntax

```
CREPORT= creport-number

DICTIONARY=dictionary-name
DBNAME=database-name
DICTIONODE=dictionary-node-name
DBNODE=database-node-name

PARAM= LIST
NOLIST
EJECT

KEY key-field-name 'key-field-value'

SELECT BUFFER record-name level-number IN PATH path-id

WHEN boolean-expression
```

Parameters

General syntax rules for the CA IDMS reports are described in detail in Section 1. Syntax rules specific to the CREPORTs are discussed as follows:

- **CREPORT=creport-number**
  Identifies the DC/UCF system report being requested. CREPORT must begin in column 1; all other parameters begin in column 2.
  Creport-number is a 3-digit number that identifies the report module. Leading zeros can be omitted.
  With the exception of CREPORTs 028 and 050, which must be run alone, multiple reports can be requested in the same job run.
Examples

Example 1

These control statements can be used to request CREPORTS 001, 010, and 041. Optionally, you can omit leading zeros for the report modules and/or use a blank or equal sign to separate each report. The report modules used to run the reports are in the CULPDICTIONARY; data for the reports is taken from the DOCUDICT dictionary.

```
DATABASE DICTNAME=CULPDICTIONARY DBNAME=DOCUDICT
CREPORT=001,010,041
```

The same reports can be requested by specifying each report on a separate line:

```
CREPORT=1
CREPORT=10
CREPORT=41
```

Example 2

These control statements can be used to produce a report on the task occurrence BYE. The report modules used to produce the report are in the default dictionary; data for the report is taken from the DOCUNWK dictionary.

```
DATABASE DBNAME=DOCUNWK
CREPORT=005
SEL BUFFER WHEN TASK-NAME-023 EQ 'BYE'
```

Example 3

These control statements can be used to produce a report on all message occurrences whose id begins with DC301. The report modules used to produce the report are in the CULPDICTIONARY; data for the report is taken from the default dictionary.

```
DATABASE DICTNAME=CULPDICTIONARY
CREPORT=005
SEL MSG-KEY-116 WHEN MSG-KEYRED-116 EQ 'DC301'
```

Example 4

These control statements can be used to request the three network object reports. A parameter listing is requested for CREPORT 001; parameter listings are not requested for CREPORTs 002 and 003.

```
DATABASE DICTNAME=CULPDICTIONARY DBNAME=TESTDICTIONARY
PARAM=LIST
CREPORT=1
PARAM=NOLIST
CREPORT=2,3
```
CA ADS Parameter Reports (CREPORTs 040 and 045)

The CA ADS parameter reports provide information on the CA ADS runtime environment. The CA ADS definitions for each system are represented in the dictionary by the CVGDEFS-142 record, which is a logical extension of the SYS-041 record.

CREPORT 040 describes the CA ADS environment for executable systems; CREPORT 045 describes the CA ADS environment for systems that have been defined but not generated. The fields and format of the two reports are the same.

Sample CREPORT 040:

```
REPORT NO. 40 CA IDMS/DC ADS ONLINE REPORT REL nn.n mm/dd/yy PAGE 1
CREPORT 040 LISTING OF ADS ONLINE PARAMETERS
OBJECT REPORT

SYSTEM VERSION AUTO DIALOG PRIMARY SECONDARY TCF MAXIMUM MENU PRIMARY SECONDARY BUILDER ACTION ACTIV
VERSION 99 TASK CODE ADS TASK CODE ADS2 TASK CODE ADS2T LINKS IS POOL POOL CODE CODE LOG
AUTO DIALOG ADS ADS2 ADS2T USER 10 4084 4084 R
DIAGNOSTIC NEWPAGE RESOURCES AUTOSTATUS COBOL MOVE DIALOG STATISTICS REC RECORD STORAGE
SCREEN MAPOUT ARE ENABLED OPT/MAND ENABLED OPT/MAND ON/OFF SEL/ALL INTERVAL COMPRESSION MODE
YES NO FIXED YES OPTIONAL NO OPTIONAL OFF ON/OFF ALL SEL ALL SEL/ALL ALL SEL/ALL SENSOR ALL SELECT
- FAST MODE THRESHOLD / INTERVAL OFF 10 ASSO-STAT-DEF-REC VERSION 1

SYSTEM VERSION AUTO DIALOG PRIMARY SECONDARY TCF MAXIMUM MENU PRIMARY SECONDARY BUILDER ACTION ACTIV
VERSION 105 TASK CODE ADS TASK CODE ADS2 TASK CODE ADS2T LINKS IS POOL POOL CODE CODE LOG
AUTO DIALOG ADS ADS2 ADS2T USER 10 4084 33000 R
DIAGNOSTIC NEWPAGE RESOURCES AUTOSTATUS COBOL MOVE DIALOG STATISTICS REC RECORD STORAGE
SCREEN MAPOUT ARE ENABLED OPT/MAND ENABLED OPT/MAND ON/OFF SEL/ALL INTERVAL COMPRESSION MODE
YES NO FIXED YES OPTIONAL NO OPTIONAL OFF ON/OFF ALL SEL ALL SEL/ALL ALL SEL/ALL SENSOR ALL SELECT
- FAST MODE THRESHOLD / INTERVAL OFF 10 ASSO-STAT-DEF-REC VERSION 1

Field Descriptions

A description of the fields in the sample report follows:

- **SYSTEM VERSION**
  Identifies the system associated with these CA ADS parameters.

- **AUTO DIALOG**
  Identifies the mainline dialog that is executed immediately at run time (the mainline dialog menu screen is bypassed).

- **PRIMARY TASK CODE**
  Identifies the task code entered by the user to initiate the CA ADS runtime system. This task code must invoke the program ADSORUN1; the default is ADS.
- **SECONDARY TASK CODE**
  Identifies the task code that invokes the program ADSOMAIN; the default is ADS2TASK.

- **TCF TASK CODE**
  Identifies the task code that invokes the program ADSOMAIN while running under the transfer control facility (TCF). The default is ADS2T.

- **MAXIMUM LINKS**
  Identifies the maximum number of dialog levels that can be defined by each respective CA ADS application thread. The default is 10.

- **MENU IS**
  Identifies which dialog names will be displayed on the CA ADS menu. USER (default) indicates that only those mainline dialog names for which the current user is authorized will be displayed; ALL indicates that all mainline dialog names known to the DC/UCF system will be displayed.

- **PRIMARY POOL**
  Indicates the size of the primary record buffer. The default is 4000.

- **SECONDARY POOL**
  Indicates the size of the secondary pool to be allocated from DC storage when the primary pool is full. The default is 2000.

- **BUILDER CODE**
  Identifies the builder code for the record that defines the CA ADS parameters. For a description of builder codes, see Builder Codes table at the end of this section.

- **ACTION CODE**
  Indicates whether the record that defines the CA ADS parameters has been updated (U) or deleted (D). Blank indicates that the CA ADS parameters have not been updated since the last time the system was generated.

- **ACTIV LOG**
  Indicates whether the activity log for the database commands in a dialog is on (YES) or off (NO).

- **DIAGNOSTIC SCREEN**
  Indicates whether CA ADS bypasses display of the Dialog Abort Information screen when the runtime system abends a dialog. YES indicates that the diagnostic screen will not be displayed; NO indicates that the screen will be displayed.

- **NEWPAGE MAPOUT**
  Indicates how a mapout is performed when a dialog's map is already displayed as the result of a previous mapout. NO indicates that CA ADS will transmit only the map's data fields and message field; YES indicates that CA ADS will always perform a new page mapout.

- **RESOURCES ARE**
  Specifies whether information from the CA ADS Terminal Block (OTB) and the variable dialog block (VDB) in the storage pool will be written to the scratch area (DDLDCSCR) across a pseudo-converse.
  FIXED indicates that the storage is not relocatable; the buffer pools remain in the storage pool provided that the fast mode threshold has not been exceeded. RELOCATABLE indicates that the storage is relocatable; the buffer pools are written to scratch across a pseudo-converse and the storage is freed.
- **AUTOSTATUS ENABLED**
  Indicates whether (YES) or not (NO) the AUTOSTATUS setting for dialog generation sessions is enabled.

- **AUTOSTATUS OPT/MAND**
  Indicates whether (OPTIONAL) or not (MANDATORY) the application developer is allowed to override the default AUTOSTATUS setting during dialog generation.

- **COBOL MOVE ENABLED**
  Indicates how CA ADS moves the result of an arithmetic or assignment command into the target field. YES instructs CA ADS to use COBOL rules; NO instructs CA ADS to use CA ADS rules.

- **COBOL MOVE OPT/MAND**
  Indicates whether (OPTIONAL) or not (MANDATORY) the application developer is allowed to change the COBOL MOVE setting on a dialog-by-dialog basis.

- **DIALOG STATISTICS ON/OFF**
  Indicates whether (YES) or not (NO) dialog statistics are collected for CA ADS dialogs.

- **DIALOG STATISTICS SEL/ALL**
  Indicates whether statistics are collected for dialogs on a dialog-by-dialog basis (SELECTED) or for all dialogs (ALL).

- **DIALOG STATISTICS INTERVAL**
  Specifies the interval at which dialog statistics are written to the log file after being accumulated the indicated number of times. 0 indicates that statistics are written to the log file after an application terminates.

- **RECORD COMPRESSION**
  Indicates whether (YES) or not (NO) record buffer blocks (RBBs) are compressed across a pseudo-converse when they are retained in the storage pool.

- **STORAGE MODE**
  Indicates how storage for record buffer blocks (RBBs) is to be allocated. SGENSIZE uses the buffer sizes specified in the PRIMARY POOL and SECONDARY POOL parameters of the ADSO system generation statement; CALCULATED uses the calculated size of the RBBs for an application or dialog when allocating storage.

- **FAST MODE THRESHOLD / INTERVAL**
  Indicates whether (ON) or not (OFF) the CA ADS runtime system writes record buffer blocks (RBBs) and statistics control blocks to scratch across a pseudo-converse. INTERVAL specifies the size of the fast mode threshold.

- **STATUS DEFINITION RECORD**
  Specifies the name and version of the status definition record and whether (OPTIONAL) or not (MANDATORY) the application developer is allowed to override the default status definition record specification during dialog generation.
Defined Devices Report (CREPORT 029)

The Defined Devices report provides information on line and physical terminal device types supported by DC/UCF. CREPORT 029 obtains information from the DCDEVICES-127 dictionary record, which is stored at installation. DC/UCF compilers use DCDEVICES-127 record occurrences when handling device-dependent syntax.

Sample CREPORT 029:

<table>
<thead>
<tr>
<th>DEFINED DEVICE</th>
<th>LOWER PARSE MODULE</th>
<th>LINE/Terminal</th>
<th>TERMINAL CODE</th>
<th>DEVICE CODE</th>
<th>Access Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR33</td>
<td>RHDCP06E</td>
<td>TERM</td>
<td>19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASYNC</td>
<td>RHDCLOSE</td>
<td>LINE</td>
<td>6</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>BSC1</td>
<td>RHDCLOBB</td>
<td>LINE</td>
<td>B</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>BSC2</td>
<td>RHDCLOCB</td>
<td>LINE</td>
<td>C</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>BSC3</td>
<td>RHDCLODB</td>
<td>LINE</td>
<td>D</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>BULK</td>
<td>RHDCPOL5</td>
<td>TERM</td>
<td>47</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCI</td>
<td>RHDCLOLS</td>
<td>LINE</td>
<td>L</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>CONSOLE</td>
<td>RHDCLO4W</td>
<td>LINE</td>
<td>4</td>
<td>W</td>
<td></td>
</tr>
</tbody>
</table>

Field Descriptions

A description of the fields in the sample report follows:

- **Defined Device**
  Identifies the device type being described.

- **Lower Parse Module**
  Identifies the name of the syntax tree used to handle the device.

- **Line/Terminal**
  Indicates whether the device being described is a line or a physical terminal.

- **Terminal Code**
  Identifies the terminal type code of the physical terminal. For a list of the possible values, see the **CA IDMS DSECT Reference section**.

- **Device Code** and **Access Method**
  Identify the line type code and the line/terminal access method code for each line.

- **Comments**
  Displays any comments associated with the occurrence. This field is not shown in the sample report.
Defined Messages Report (CREPORT 028)

The Defined Messages report provides information on messages that have been defined for the DC/UCF system. Messages are represented in the DDLDCMSG area of the dictionary by the MESSAGE-116 record.

Sample CREPORT 028:

<table>
<thead>
<tr>
<th>REPORT NO. 28</th>
<th>CA IDMS/DC SYSTEM GENERATION REPORT</th>
<th>REL nn.n</th>
<th>nn/dd/yy</th>
<th>PAGE 29</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREPORT 028</td>
<td>CA IDMS/DC SYSTEM GENERATION REPORT</td>
<td>DEFINED MESSAGES</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MESSAGE ID: AB030026  BUILDER CODE: D

SEVERITY: 6  DESTINATIONS:  DESTINATION ID:

MESSAGE TEXT  LINE NUMBER 1:

DATA TYPE NOT RECOGNIZED

COMMENT NUMBER 100: MODULE(S) = ________

COMMENT NUMBER 200:

COMMENT NUMBER 300: ADDITIONAL INFORMATION ...

COMMENT NUMBER 400:

COMMENT NUMBER 500:

MESSAGE ID: AB030057  BUILDER CODE: D

SEVERITY: 6  DESTINATIONS:  DESTINATION ID:

MESSAGE TEXT  LINE NUMBER 1:

UNDETERMINED SYNTAX ERROR IN INSTRUCTION

COMMENT NUMBER 100:

COMMENT NUMBER 100: MODULE(S) = ________

COMMENT NUMBER 200:

Field Descriptions

A description of the fields in the sample report follows:

- **MESSAGE ID**
  Specifies the identifier assigned to the message.

- **BUILDER CODE**
  Identifies the builder code for the record that defines the message. For a description of builder codes, see Builder Codes table at the end of this section.

- **SEVERITY**
  Identifies the severity level assigned to the message.

- **DESTINATIONS**
  Identifies the destinations to which the message line is routed.

- **DESTINATION ID**
  Identifies the terminal to which the message line is routed when the destination identifier flag is set.
MESSAGE TEXT and LINE NUMBER
Identify the text of the message. When multiple lines are defined for a message, each line is listed separately.

COMMENT NUMBER
Identifies comments associated with the message.

Destination Reports (CREPORTs 007 and 024)

Destination reports provide information on destination occurrences that have been defined to the dictionary. CREPORT 007 provides destination information for executable systems; these destinations are represented in the dictionary by the DESTLST-027 record. CREPORT 024 provides information for systems that have been defined but not generated; these destinations are represented in the dictionary by the DEST-028 record. The fields and format of the two reports are the same.

Sample CREPORT 007:

<table>
<thead>
<tr>
<th>DESTINATION</th>
<th>BUILDER</th>
<th>ACTION CODE</th>
<th>VERSION</th>
<th>DISABLED</th>
<th>MEMBER TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>USWSWDPL</td>
<td>R</td>
<td>1</td>
<td>YES</td>
<td>TERMINAL</td>
<td></td>
</tr>
</tbody>
</table>

Field Descriptions

A description of the fields in the sample report follows:

SYSTEM NAME and VERSION
Identify the name and version number of the system associated with the destination being described.

DESTINATION
Identifies the destination being described.

BUILDER
Identifies the builder code for the record that defines the destination. For a description of builder codes, see Builder Codes table at the end of this section.

ACTION CODE
Indicates whether the destination occurrence has been updated (U) or deleted (D). A blank indicates that the destination has not been changed since that last time the system was generated.

VERSION
Indicates the version number of the destination occurrence.
Load Area Report (CREPORT 050)

The Load Area report provides information on load modules. Load modules are stored in the DDLDCLOD area of the dictionary and are represented in the dictionary by the LOADHDR-156 record.

Sample CREPORT 050:

<table>
<thead>
<tr>
<th>LOAD AREA REPORT</th>
<th>CA IDMS/DC LOAD AREA REPORT</th>
<th>REL nn.n</th>
<th>mm/dd/yy</th>
<th>PAGE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUM RLD ENTRIES</td>
<td>ENTRY PT ADDRESS</td>
<td>MODULE</td>
<td>COMPILE DATE</td>
<td>COMPILE TIME</td>
</tr>
<tr>
<td>MODULE NAME</td>
<td>VERSION</td>
<td>LENGTH</td>
<td>DATE</td>
<td>TIME</td>
</tr>
<tr>
<td>$ACFgtAT</td>
<td>1</td>
<td>118</td>
<td>mm/dd/yy</td>
<td>hhmmss</td>
</tr>
<tr>
<td>IDMSCSTB</td>
<td>99</td>
<td>563</td>
<td>mm/dd/yy</td>
<td>hhmmss</td>
</tr>
<tr>
<td>IDMSCSTB</td>
<td>105</td>
<td>930</td>
<td>mm/dd/yy</td>
<td>hhmmss</td>
</tr>
<tr>
<td>IDMSCSTB</td>
<td>777</td>
<td>563</td>
<td>mm/dd/yy</td>
<td>hhmmss</td>
</tr>
<tr>
<td>SQACVSS1</td>
<td>1</td>
<td>1,640</td>
<td>mm/dd/yy</td>
<td>hhmmss</td>
</tr>
<tr>
<td>SQACVSS2</td>
<td>1</td>
<td>1,176</td>
<td>mm/dd/yy</td>
<td>hhmmss</td>
</tr>
</tbody>
</table>

Field Descriptions

A description of the fields in the sample report follows:

- **MODULE NAME and VERSION**
  Identify the name and version number of the load module being described.

- **NUM RLD ENTRIES**
  Identifies the number of entries in the relocation dictionary (RLD) for the load module.

- **ENTRY PT ADDRESS**
  Identifies the entry point address of the load module.

- **MODULE LENGTH**
  Identifies the length, in bytes, of the object text.

- **COMPILE DATE**
  Identifies the date the load module was created.

- **COMPILE TIME**
  Identifies the time the load module was created.
- **DELETE FLAG**
  Indicates whether the load module has been logically deleted. The flag is set ON when a new module is generated for the load module in use or when a deletion is requested. The module is not eligible for replacement when the flag is OFF.

- **MODULE TYPE**
  Identifies the type of load module: access module, subschema, map, CA ADS dialog, edit/code table, or mainline dialog.

## Mapping Reports (CREPORTs 030 through 035)

Mapping reports describe the interrelationships among maps, panels, map fields, and panel fields that have been defined for DC/UCF systems. Maps are represented in the dictionary by the MAP-098 record; map fields are represented by the MAPFLD-124 record. The following table summarizes each report:

<table>
<thead>
<tr>
<th>Mapping Report ID</th>
<th>Report Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREPORT 030</td>
<td>Map Record Indices</td>
<td>Lists map record elements sorted on map name</td>
</tr>
<tr>
<td>CREPORT 031</td>
<td>Map Field Indices</td>
<td>Lists map panel-fields sorted on map name</td>
</tr>
<tr>
<td>CREPORT 032</td>
<td>Maps by Panel</td>
<td>Lists map occurrences sorted on panel name</td>
</tr>
<tr>
<td>CREPORT 033</td>
<td>Maps in alphabetic order</td>
<td>Lists all map occurrences in alphabetic order</td>
</tr>
<tr>
<td>CREPORT 034</td>
<td>Maps by Record Name</td>
<td>Lists map occurrences by map record name</td>
</tr>
<tr>
<td>CREPORT 035</td>
<td>Maps by Element Name</td>
<td>Lists map occurrences sorted on map element name</td>
</tr>
</tbody>
</table>

### Sample CREPORT 030:

```
REPORT NO. 30 CA IDMS/DC MAPPING REPORT REL nn.n mm/dd/yy PAGE 1
CREPORT 030 MAP NAME: ADMI01M
MAP VERSION: 1
MAP RECORD INDICES

OLMPF-0003 CURSOR
ALARM NO
UNLOCK YES
RESET YES
MAP DATE mm/dd/yy
MAP TIME hhmmss
FIELD COUNT 4
RECORD COUNT 1

RECORD NAME INDEX
EMPLOYEE 1
```

### Sample CREPORT 031:

```
REPORT NO. 31 CA IDMS/DC MAPPING REPORT REL nn.n mm/dd/yy PAGE 1
CREPORT 031 MAP NAME: ADMI01M
MAP VERSION: 1
MAP FIELD INDICES

OLMPF-0003 CURSOR
ALARM NO
UNLOCK YES
RESET YES
MAP DATE mm/dd/yy
MAP TIME hhmmss
FIELD COUNT 4
RECORD COUNT 1

FIELD NAME INDEX
EMP-ID-0415 2
EMP-NAME-0415 3
```

### Sample CREPORT 032: 
Sample CREPORT 033:

<table>
<thead>
<tr>
<th>MAP NAME</th>
<th>MAP VERSION</th>
<th>MAP BUILDER</th>
<th>PANEL NAME</th>
<th>PANEL VERSION</th>
<th>MAP DATE</th>
<th>MAP TIME</th>
<th>FIELD COUNT</th>
<th>RECORD COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCD01M</td>
<td>1</td>
<td>G</td>
<td>ABCD01M-OLMPANEL</td>
<td></td>
<td>mm/dd/yy</td>
<td>hhmmss</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Sample CREPORT 034:

<table>
<thead>
<tr>
<th>MAP NAME</th>
<th>MAP VERSION</th>
<th>PANEL NAME</th>
<th>PANEL VERSION</th>
<th>MAP DATE</th>
<th>MAP TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD21M007</td>
<td>1</td>
<td>AD21M007-OLMPANEL</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AD21M008</td>
<td>1</td>
<td>AD21M008-OLMPANEL</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AD21M010</td>
<td>1</td>
<td>AD21M010-OLMPANEL</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sample CREPORT 035:

<table>
<thead>
<tr>
<th>MAP NAME</th>
<th>MAP VERSION</th>
<th>PANEL NAME</th>
<th>PANEL VERSION</th>
<th>MAP DATE</th>
<th>MAP TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>A310M4</td>
<td>1</td>
<td>A310M4-OLMPANEL</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Descriptions

Because many of the field names on the map reports are the same from report to report, a description of all the fields is presented once in alphabetical order.

- **ALARM**
  Indicates whether the terminal’s audible alarm will sound automatically when the map is written out to the screen (CREPORTs 030 and 031 only).
- **BUILDER**
  Identifies the builder code for the record that defines the map (CREPORT 032 only). For a description of builder codes, see the Builder Codes table at the end of this section.

- **CURSOR**
  Identifies the name of the panel field in which the cursor appears after a mapout operation (CREPORTs 030 and 031 only).

- **ELEMENT NAME**
  Identifies the name of a data field (record element) whose map associations are being described (CREPORT 035 only).

- **FIELD COUNT**
  Indicates the number of nonliteral fields in the map (CREPORTs 030, 031, and 032 only).

- **FIELD NAME**
  Identifies the panel fields that appear in the map (CREPORT 031 only).

- **INDEX**
  Indicates the order in which the record or record fields are used by the map (CREPORTs 030 and 031 only).

- **MAP BUILDER**
  Identifies the builder code for the record that defines the map occurrence (CREPORT 033 only). For a description of builder codes, see the Builder Codes table at the end of this section.

- **MAP DATE**
  Identifies the date the map occurrence was defined to the dictionary (CREPORTs 030, 031, 032, and 033 only).

- **MAP NAME and MAP VERSION**
  Identify the map occurrence being described.

- **MAP TIME**
  Identifies the time the map was last compiled with critical changes (CREPORTs 030, 031, 032, and 033 only).

- **PANEL NAME and PANEL VERSION**
  Identify the panel associated with the map being described (CREPORTs 032, 033, 034, and 035 only).

- **PANEL BUILDER**
  Identifies the builder code for the record that defines the panel occurrence (CREPORT 032 only). For a description of builder codes, see the Builder Codes table at the end of this section.

- **RECORD COUNT**
  Indicates the number of records used by the map.

- **RECORD NAME**
  Identifies the record used by the map (CREPORTs 030, 031, and 032 only).
**RESET**
Indicates whether all modified data tags are reset when the map is mapped out (YES) or remain unchanged (NO) (CREPORTs 030 and 032 only).

**UNLOCK**
Indicates whether the keyboard will be locked (NO) or unlocked (YES) when the map is mapped out (CREPORTs 030 and 032 only).

**Nodes and Resource Table Reports (CREPORTS 043 and 044)**

The Nodes and Resource Table reports provide information on all the resources that have been defined to the systems, including the location where the resource resides and the type of communication method used to access these nodes. The following table below summarizes each report:

<table>
<thead>
<tr>
<th>CREPORT 043</th>
<th>Listing of Nodes</th>
<th>Lists all the nodes sorted on access type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREPORT 044</td>
<td>Listing of Defined Resources</td>
<td>Lists all the resources sorted on database or destination name</td>
</tr>
</tbody>
</table>

**Sample CREPORT 043:**

```
REPORT NO. 43
CREPORT 043

CA IDMS/DC NODE NAME REPORT
REL nn.n
mm/dd/yy PAGE 1
LISTING OF NODES

SYSTEM VERSION NUMBER | NODE NAME | ACCESS TYPE | DEFAULT NODE | CV NUMBER | SVC NUMBER | BUILDER CODE | ACTION CODE
-----------------------|-----------|-------------|--------------|-----------|------------|--------------|-------------
71                     | A66VT12   | CCI         |             |           |            | R            |             
71                     | SYSTEM22  | CCI         |             |           |            | R            |             
71                     | SYSTEM92  | CCI         |             |           |            | R            |             
71                     | DBDCGR    | GROUP       | NULL        |           |            | R            |             
71                     | DBG001    | GROUP       | NULL        |           |            | R            |             
71                     | DBG002    | GROUP       | LOCAL       |           |            | R            |             
71                     | DBG003    | GROUP       | SYSTEM72    |           |            | R            |             
71                     | IDMSGR    | GROUP       | NULL        |           |            | R            |             
71                     | SYSTEM71  | LOCAL       |             |           |            | R            |             
71                     | CVN001    | SVC         | 101 173     |           |            | R            |             
71                     | CVN002    | SVC         | 102 102     |           |            | R            |             
71                     | SYSTEM72  | TCP/IP      |             |           |            | R            |             
71                     | SYSTEM73  | TCP/IP      |             |           |            | R            |             
71                     | SYSTEM74  | VTAM        |             |           |            | R            |             
71                     | TECHDC99  | VTAM        |             |           |            | R            |             
```

**Sample CREPORT 044:**

```
REPORT NO. 44
CREPORT 044

CA IDMS/DC RESOURCE TABLE REPORT
REL nn.n
mm/dd/yy PAGE 1
LISTING OF DEFINED RESOURCES

SYSTEM VERSION NUMBER | NODE NAME | ACTION CODE | DATABASE NAME | DESTINATION | NODE NAME
-----------------------|-----------|-------------|---------------|-------------|-----------
71                     | DBNAM1    | R           |               | LOCAL       |           
```
Field Descriptions

A description of the fields in the sample reports follows:

- **SYSTEM VERSION NUMBER**
  Identifies the version number of the system associated with the nodes or resources.

- **NODE NAME**
  Identifies the name of the node.

- **ACCESS TYPE**
  Specifies the communication method used to access the corresponding node.

- **DEFAULT NODE**
  For access type GROUP only. Identifies the default node to use if access to the requested group fails.

- **CV NUMBER**
  For access type SVC only. Identifies the number of the central version.

- **SVC NUMBER**
  For access type SVC only. Identifies the number of the SVC through which the system will send packets to the corresponding node.

- **BUILDER CODE**
  Identifies the builder code for the record that defines the node or resource.
  For a description of building codes, see Table 4-1 at the end of this section.

- **ACTION CODE**
  Identifies whether the node or resource occurrence has been updated (U) or deleted (D). A blank indicates that it has not been changed since the last time the system was generated.

- **DATABASE NAME**
  Identifies the name of the database that is included in the resource name table.

- **DESTINATION**
  Identifies a nodename to which requests for data can be sent.

Module Text to Card Utility (CREPORT 051)

The Module Text to Card utility lets you punch the module source code of a specified module to cards. The CREPORT and KEY parameters used to punch module CREPORT 011 to card are:

- CREPORT=051
- KEY MOD-NAME-067 ‘CREPORT 011’
How to Run CREPORT 051

You must run CREPORT 051 alone. To run CREPORT 051, include the following specification in the JCL:

- **For z/OS systems:**
  
  ```
  //SYSPCH DD SYSOUT=B,DCB=BLKSIZE=80
  ```

- **For z/VSE systems:**
  
  ```
  // ASSGN SYSPCH,X'ccc'
  ```

  - **ccc**
    
    device assignment (channel and unit) for punched output

- **For z/VM and z/VM systems:**
  
  ```
  FILEDEF SYSPCH DISK syspch output a
  ```

  - **syspch output a**
    
    file identifier of the card-image output file

How to Request Card-Image Listing

No printed output is produced by this report other than an accounting of the number of records written. To request a card-image listing of the module instead of punched cards, include the following JCL specification:

- **For z/OS systems:**
  
  ```
  //SYSPCH DD SYSOUT=A
  ```

- **For z/VSE systems:**
  
  ```
  // ASSGN SYSPCH,X'ppp'
  ```

  - **ppp**
    
    printer device assignment

- **For z/VM and z/VM systems:**
  
  ```
  FILEDEF SYSPCH PRINTER
  ```

Module Text to File Utility (CREPORT 052)

The module text to output file utility (CREPORT 052) lets you output module source code to a disk file. The CREPORT and KEY parameters used to output module CREPORT 011 to file are shown below:

- **CREPORT=052**
  
  ```
  KEY MOD-NAME-067 'CREPORT 011'
  ```

How to Run CREPORT 052

CREPORT 052 must be run alone. To run CREPORT 052, add the following specification to the JCL:

- **For z/OS files:**
Network Description Reports (CREPORTs 001-003, 014-018)

Contents

- Contents (see page 274)
- Sample Reports (see page 275)
- Field Descriptions (see page 277)

Contents

Network reports (CREPORTs 001, 002, 003, 014, 015, 016, 017, and 018) describe the lines, physical terminals, and logical terminals associated with a DC/UCF system. CREPORTs 001, 002, and 003 describe the networks that have been defined for executable systems; CREPORTs 014 through 018 describe networks for systems that have been defined but not generated.
Summary of Network Description Reports

<table>
<thead>
<tr>
<th>CREPORT Module</th>
<th>CREPORT Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>Network Description by Line (Object Report)</td>
</tr>
<tr>
<td>002</td>
<td>Network Description by Physical Terminal (Object Report)</td>
</tr>
<tr>
<td>003</td>
<td>Network Description by Logical Terminal (Object Report)</td>
</tr>
<tr>
<td>014</td>
<td>Network Description by Line (Source Report)</td>
</tr>
<tr>
<td>015</td>
<td>Network Description by Physical Terminal (Source Report)</td>
</tr>
<tr>
<td>016</td>
<td>Physical Terminals within Line (Source Report)</td>
</tr>
<tr>
<td>017</td>
<td>Network Description by Logical Terminal (Source Report)</td>
</tr>
<tr>
<td>018</td>
<td>Logical Terminal by Physical Terminal (Source Report)</td>
</tr>
</tbody>
</table>

Sample Reports

The following figures show sample output for CREPORTs 001, 002, 016, and 018. Sample output for CREPORTs 014, 015, and 017 is not shown; the fields in these three reports are the same as those in CREPORTs 001, 002, and 003, respectively.

Sample CREPORT 001:

Sample CREPORT 002:
### Sample CREPORT 016:

<table>
<thead>
<tr>
<th>PTERM NAME</th>
<th>BUILDER</th>
<th>ACTION</th>
<th>VERSION</th>
<th>TYPE</th>
<th>MAX-ERR</th>
<th>LINE-LEN</th>
<th>PAGE-LEN</th>
<th>MODEL</th>
<th>DISABLED</th>
<th>ASCII</th>
<th>LNDEL</th>
<th>CHRDEL</th>
<th>CANCEL</th>
<th>P-CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCFTB1</td>
<td>G</td>
<td></td>
<td>1</td>
<td>2F</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>00</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>UCFTB2</td>
<td>G</td>
<td></td>
<td>1</td>
<td>2F</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>00</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>UCFTB3</td>
<td>G</td>
<td></td>
<td>1</td>
<td>2F</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>00</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>UCFTB4</td>
<td>G</td>
<td></td>
<td>1</td>
<td>2F</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>00</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>UCFTB5</td>
<td>G</td>
<td></td>
<td>1</td>
<td>2F</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>00</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
</tbody>
</table>

### Sample CREPORT 018:

<table>
<thead>
<tr>
<th>PTERM NAME</th>
<th>LTERM NAME</th>
<th>BUILDER</th>
<th>ACTION</th>
<th>VERSION</th>
<th>PRIORITY</th>
<th>CASE</th>
<th>DISABLED</th>
<th>PRINTER</th>
<th>TASK CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPERATOR</td>
<td>CONSOLE</td>
<td>G</td>
<td></td>
<td>1</td>
<td>240</td>
<td>UPPER</td>
<td>NO</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>UCFTB1</td>
<td>UCFTB1</td>
<td>G</td>
<td></td>
<td>1</td>
<td></td>
<td>UPPER</td>
<td>NO</td>
<td>NO</td>
<td></td>
</tr>
</tbody>
</table>
Field Descriptions

Because many of the field names in the network reports are the same from report to report, all the fields in the sample reports are described once, in alphabetical order:

- **ACTION**
  Indicates whether the entity occurrence being described has been updated (U) or deleted (D) or remain unchanged (blank) since the most recent generate.

- **APPLICATION ID**
  Defines the VTAM application id used by the system at run time to sign the line on to VTAM (CREPORTs 001 and 014 only).

- **APPLICATION PW**
  Identifies the optional password used by the system to sign on to VTAM (CREPORTs 001 and 014 only).

- **ASCII**
  Indicates whether the physical terminal supports the ASCII character set (CREPORTs 001, 002, 014, 015, and 016 only).

- **BACKLOG**
  Identifies the maximum length for the queue of pending connections TCP/IP allows before disallowing connection requests (CREPORTs 001 and 014 only).

- **BUFFSIZE**
  Identifies the line I/O page buffer size, in bytes, for the line (CREPORTs 001 and 014 only).

- **BUILDER**
  Identifies the builder code for the record that defines the entity occurrence. For a description of builder codes, see Builder Codes table at the end of this section.

- **CANCEL**
  Identifies the control character used as the attention key by the physical terminal device type (CREPORTs 001, 002, 014, 015, and 016 only).

- **CASE**
  Identifies the character set the logical terminal uses on input (CREPORTs 001, 003, 014, 017, and 018 only).

- **CHRDEL**
  Identifies the control character that is used to delete characters (CREPORTs 001, 002, 014, 015, and 016 only).
- **DDNAMES**
  Identifies the ddname and/or file id for the line, as specified in the system startup JCL (CREPORTs 001 and 014 only).

- **DEST**
  Specifies the line destination.

- **DEV TYPE or TYPE**
  (Lines only) Identifies the generic linetype for the terminals that are included in the line group (CREPORTs 001, 002, 014, 015, and 016 only).

- **HOST NAME/ADDRESS**
  Identifies the name or the IP address of the host (CREPORTs 001 and 014 only).

- **IDLE**
  Identifies the time interval a non-permanent DDS connection stays in an idle state after the corresponding DDS request has finished. (CREPORTs 001 and 014 only).

- **LINE-LEN**
  Identifies the maximum line length in characters for the physical terminal (CREPORTs 001, 002, 014, 015, and 016 only).

- **DISABLED/DISABL**
  Indicates whether the line group, physical terminal, or logical terminal is disabled when the system is started.

- **LNDEL**
  Identifies the control character (flush character) that is used to delete lines (CREPORTs 001, 002, 014, 015, and 016 only).

- **MAX ERR**
  Indicates the number of retries allowed after a terminal I/O error before the teleprocessing monitor disables the physical terminal (CREPORTs 001, 002, 014, 015, and 016 only).

- **MAX CON**
  For a DDSTCPIP type PTERM, identifies the maximum number of active connections allowed from the local system. For a LISTENER type PTERM, identifies the maximum number of active BULK PTERM that can be started from that listener (CREPORTs 001 and 014 only).

- **MODE**
  Indicates whether the mode of the task attached by the listener. Valid values are SYSTEM or USER. (CREPORTs 001 and 014 only)

- **MODEL**
  Identifies the model number of the physical terminal (CREPORTs 001, 002, 014, 015, and 016 only).

- **NAME**
  Identifies the entity being described (can be preceded by SYSTEM, LINE, LTERM, or PTERM).
- **PAGE LEN**
  Identifies the maximum page size, in text lines, for the physical terminal (CREPORTs 001, 002, 014, 015, and 016 only).

- **PARM**
  Identifies a string that is passed to the task attached by generic listening (CREPORTs 001 and 014 only).

- **PERM CON**
  Identifies the number of permanent DDS connections that can exist between the host and the target system (CREPORTs 001 and 014 only).

- **PLUG-IN**
  Identifies the name of the plug-in module that implements support for specific TCP/IP stack implementations (CREPORTs 001 and 014 only).

- **PORT**
  Indicates the number of the listener port (CREPORTs 001 and 014 only).

- **PORT-RANGE**
  Identifies the range of port numbers that are used to BIND the local sockets explicitly. Each time a new DDS connection is established, the first free port from the range is selected and associated (bound) with the corresponding socket. If no free port is found, the request is aborted. (CREPORTs 001 and 014 only).

- **PRINTER/PRINT**
  (IDMS DC only) Indicates whether the logical terminal is a 3280-type printer (CREPORTs 001, 003, 014, 017, and 018 only).

- **PRINTER CLASS(ES)/PRT CLS**
  (IDMS DC only) Defines the default printer class for the physical terminal.

- **PRIORITY/PRIOR**
  Identifies the dispatching priority for requests to IDMS DC from the logical terminal (CREPORTs 001, 003, 014, 017, and 018 only).

- **REPEAT COUNT/RPT CT**
  Indicates the number of times to clone the physical and eventual associated logical terminal when a central version is started.

- **REQUEST PARAM LIST**
  Indicates the number of write-only input/output requests that can be handled at one time by the local VTAM line group (CREPORTs 001 and 014 only).

- **STACK NAME**
  Identifies the job name of the TCP/IP stack (CREPORTs 001 and 014 only).

- **TARGET NAME/ADDRESS**
  Identifies the name or the IP address of the target host (CREPORTs 001 and 014 only).
• **TASK CODE**  
Identifies the name of the task code to start when a connection request arrives at a listener PTERM (CREPORTs 001 and 014 only).

• **TGT PORT**  
Identifies the number of the target port (CREPORTs 001 and 014 only).

• **TYPE**  
(Physical terminals only) Identifies the device type of the physical terminal. See DEV TYPE above for a description of line type.

• **VERSION**  
Identifies the version number of the entity occurrence being described.

---

**CA OLQ Reports (CREPORTs 041 and 046)**

CA OLQ reports provide information on CA OLQ runtime parameters that have been defined for each system. CA OLQ parameters are represented in the dictionary by the CVGDEFS-142 record, which is a logical extension of the SYS-041 record.

CREPORT 041 describes CA OLQ parameters for executable systems; CREPORT 046 describes CA OLQ parameters for systems that have been defined but not generated. The fields and format of the two reports are the same.

**Sample CREPORT 041:**

<table>
<thead>
<tr>
<th>SYSTEM</th>
<th>OLQ</th>
<th>VERSION</th>
<th>TRANS</th>
<th>ID</th>
<th>NAME</th>
<th>CODE</th>
<th>CODE</th>
<th>LINE SIZE</th>
<th>PAGE SIZE</th>
<th>COUNT</th>
<th>PAGE SIZE</th>
<th>PAGE SIZE</th>
<th>LINE SIZE</th>
<th>RETENTION</th>
<th>RETENTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>99</td>
<td>OLQ</td>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>105</td>
<td>OLQ</td>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field Descriptions**

A description of the fields in the sample report follows:

• **SYSTEM VERSION**  
Identifies the system whose OLQ runtime definitions are being described.
- **OLQ TRANS ID**
  Identifies the task code used to invoke the CA OLQ runtime system.

- **PFKEY MOD NAME**
  Identifies the module that contains the default control key assignments for CA OLQ.

- **BUILDER CODE**
  Identifies the builder code for the record that defines the CA OLQ parameters. For a description of builder codes, see Builder Codes table at the end of this section.

- **ACTION CODE**
  Indicates whether the record that defines the CA OLQ parameters has been updated (U) or deleted (D) or remains unchanged (blank) since the most recent generate.

- **PRINT LINE SIZE**
  Identifies the line length, in characters, for CA OLQ report output on TTY-type terminals.

- **PRINT PAGE SIZE**
  Identifies the page length, in lines, for CA OLQ report output on TTY-type terminals.

- **INTERRUPT COUNT**
  Indicates the maximum number of records that will be read by CA OLQ before requesting further terminal input.

- **INT STORAGE PAGE SIZE**
  Indicates the size, in bytes, of the CA OLQ internal storage pages used to store control data across a pseudo-converse.

- **REPORT FILE PAGE SIZE**
  Indicates the size, in bytes, for CA OLQ report file pages written to the DDLDCRUN area.

- **INPUT LINE SIZE**
  Identifies the number of lines on the screen that are available for input.

- **REPORT RETENTION**
  Identifies the default report retention time, in days.

- **MAXIMUM RETENTION**
  Identifies the maximum report retention time, in days.

- **REPORT DICT**
  Identifies the name of the dictionary in which catalog information on CA OLQ saved reports is stored.

- **SQL ACCESS**
  Indicates how SQL statements used to access a CA IDMS/DB database will be processed. OLQSQL indicates CA OLQ will process the statements; IDMSSQL indicates CA IDMS/DB will process the statements.

- **BATCH CLASS**
  (z/OS only) Identifies the print class used by CA OLQ when submitting batch jobs.
Program Description Reports (CREPORTs 004 and 019)

Program description reports provide information on programs that have been included in a DC/UCF system. System-supplied programs, subschemas, database procedures, maps, edit and code tables, CA ADS dialogs, and user programs written in COBOL, PL/I, and Assembler are represented in the dictionary as program occurrences.

CREPORT 004 describes programs associated with executable systems; these programs are represented in the dictionary by the PROGLST-049 record. CREPORT 019 describes programs associated with systems that have been defined but not generated; these programs are represented by the PROG-051 record. The fields and format of the two reports are the same.

Sample CREPORT 004:

<table>
<thead>
<tr>
<th>REPORT NO. 04</th>
<th>CA IDMS/DC SYSTEM GENERATION REPORT</th>
<th>REL nn.n</th>
<th>mm/dd/yy</th>
<th>PAGE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREPORT 084</td>
<td>SYSTEM NAME: DCSYSTEM</td>
<td>SYSTEM VERSION: 110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROGRAM</td>
<td>TYPE</td>
<td>BUILDER ACTION</td>
<td>VERSION</td>
<td></td>
</tr>
<tr>
<td>$ACF@GEN</td>
<td>TABLE</td>
<td>R 1</td>
<td>ISA-SIZE</td>
<td>ERROR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>THRESHOLD</td>
<td>THRESHOLD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LOAD FROM</td>
<td>LANGUAGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ASSEMBLER</td>
<td>SAVE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AREA</td>
<td>RESIDENT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CONCURRENT</td>
<td>REUSABLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>REENTRANT</td>
<td>OVERLAYABLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DISABLED</td>
<td>PROTECT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NEW COPY</td>
<td>NO</td>
</tr>
</tbody>
</table>

16-Jan-2018 282/898
Field Descriptions

A description of the fields in the sample report follows:

- **SYSTEM NAME and VERSION**
  Identify the name and version of the system associated with the program occurrences being described.

- **PROGRAM**
  Identifies the program being described.

- **TYPE**
  Identifies the program occurrence as a PROGRAM, MAP, SUBSCHEMA, TABLE, DIALOG, ACCESS MODULE, or MAP HELP.

- **BUILDER**
  Identifies the builder code for the record that defines the program occurrence. For a description of builder codes, see Builder Codes table at the end of this section.

- **ACTION**
  Indicates whether the program occurrence has been updated (U) or deleted (D) or remains unchanged (blank) since the most recent generate.

- **VERSION**
  Identifies the version number associated with the program occurrence.

- **ISA-SIZE**
  (IDMS DC only) Indicates the amount of storage, in bytes, that is allocated for the program's initial storage area (ISA) (applies to Assembler and PL/I programs only).

- **ERROR THRESHOLD**
  (IDMS DC only) Indicates the number of program check errors that can occur before the program is disabled by the system.

- **DUMP THRESHOLD**
  (IDMS DC only) Indicates the maximum number of dumps to be taken for program check errors that occur in the program.

- **LOAD FROM**
  Indicates whether the program resides in a load library (LOADLIB) or in the load area of the dictionary (DICTIONARY).
- **LANGUAGE**
  Identifies the source language of the program.

- **SAVE AREA**
  Indicates whether a save area is acquired automatically before each execution of the program.

- **RESIDENT**
  Indicates whether the program is made resident when the DC/UCF system is started (YES) or whether the program is a nonresident program that is loaded into the storage pool on request (NO).

- **CONCURRENT**
  Indicates whether the program can process more than one request concurrently.

- **REUSABLE**
  Indicates whether the program is reusable.

- **REENTRANT**
  Indicates whether the program is fully reentrant, quasi-reentrant, or nonreentrant.

- **OVERLAYABLE**
  (IDMS DC only) Indicates whether the program can be overlaid in the program pool.

- **DISABLED**
  Indicates whether the program is disabled when the DC/UCF system is started.

- **PROTECT**
  Indicates whether the DC/UCF storage protection feature is in effect for the program.

- **NEW COPY**
  Indicates whether the new copy facility is enabled.

- **MAINLINE DIALOG**
  Indicates whether (YES) or not (NO) the program is a CA ADS mainline dialog.

- **EXCLUDE FROM MENU**
  For a CA ADS dialog, indicates whether (YES) or not (NO) the dialog will appear on the CA ADS menu screen.

- **DIALOG STATS ON**
  For a CA ADS dialog, indicates whether (YES) or not (NO) statistics are collected.

- **DYNAMIC**
  Indicates whether (YES) or not (NO) users are allowed to define additional versions of the program at run time either by means of the DCMT VARY DYNAMIC PROGRAM command or, if the program is eligible for automatic definition, through definition of null PDEs on the SYSTEM system generation statement.

- **MPMODE**
  Identifies the multiprocessing mode (MPMODE) for the program. SYSTEM directs DC/UCF to assign a mode to the program at execution time. ANY specifies an MPMODE of ANY.
• MULTIPLE ENCLAVE
Indicates whether this program is eligible to use the same high level Language Environment process/enclave when multiple programs are executed in the same task.

Queue Description Reports (CREPORTs 006, 022, and 023)

Queue description reports provide information on queue occurrences associated with a DC/UCF system. CREPORT 006 describes queues associated with executable systems; these queues are represented in the dictionary by the QUEUELST-029 record. CREPORTs 022 and 023 describe queues associated with systems that have been defined but not generated; these queues are represented by the QUEUE-030 record. The fields and format of the three reports are almost identical.

Sample CREPORT 006:

<table>
<thead>
<tr>
<th>QUEUE</th>
<th>PROGRAM</th>
<th>TASK</th>
<th>BUILDER</th>
<th>ACTION</th>
<th>VERSION</th>
<th>THRESHOLD VALUE</th>
<th>UPPER LIMIT</th>
<th>DISABLED</th>
<th>RETENTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLQQNOTE</td>
<td>OLQSNOTE</td>
<td>OLQTNOTE</td>
<td>R</td>
<td>U</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>NO</td>
<td>1</td>
</tr>
</tbody>
</table>

Field Descriptions

A description of the fields in the sample report follows:

• SYSTEM NAME and VERSION
Identifies the name and version number of the system associated with the queues being described.

• QUEUE
Identifies the queue being described.

• PROGRAM
Identifies the name of the initial program invoked by the task associated with the queue.

• TASK
Identifies the task code for the task invoked when the number of entries in the queue reaches the limit defined with the THRESHOLD VALUE parameter (below).

• BUILDER
Identifies the builder code for the record that defines the queue occurrence. For a description of builder codes, see the Dictionary Builder Codes table at the end of this section.
CA IDMS - 19.0

- **ACTION**
  Indicates whether the queue occurrence has been updated (U) or deleted (D) or remains unchanged (blank) since the most recent generate.

- **VERSION**
  Indicates the version number of the queue being described.

- **THRESHOLD VALUE**
  Indicates the number of entries that must be in the queue before the system invokes the task associated with the queue.

- **UPPER LIMIT**
  Indicates the maximum number of records that can be directed to the queue.

- **DISABLED**
  Indicates whether the queue is disabled at system startup.

- **QUEUE RETENTION**
  Indicates the queue retention period in days.

SQL CACHE Reports (CREPORTs 047 and 048)

SQL CACHE reports provide information on SQL CACHE parameters that have been defined for each system. SQL CACHE parameters are represented in the dictionary by the CVGDEFS-142 record, which is a logical extension of the SYS-041 record.

CREPORT 047 describes SQL CACHE parameters for systems that have been defined but not generated; CREPORT 048 describes SQL CACHE parameters for executable systems. The fields and the formats of the two reports are the same.

**Sample CREPORT 048:**

```
REPORT NO. 48          CA IDMS SQL CACHE REPORT   REL nn.n
CREPORT 048            LISTING OF SQL CACHE PARAMETERS
                        OBJECT REPORT

SYSTEM VERSION  99     DEFAULT CACHING IS OFF
                      STATEMENTS

SYSTEM VERSION  110    DEFAULT CACHING IS ON
                      EXCEPT CONNECT TO APPLDICT
                      EXCEPT CONNECT TO TSTDICT
                      700 STATEMENTS
```

Field Descriptions

A description of the fields in the sample report follows:

- **SYSTEM VERSION**
  Identifies the system whose SQL CACHE runtime definitions are being described.
DEFAULT CACHING
Specifies whether caching of dynamic SQL statements is enabled by default.

n STATEMENTS
Specifies the maximum number (n) of SQL statements that can be placed in SQL CACHE.

EXCEPT CONNECT TO
Identifies the name of a dictionary/catalog to which a user of the CV can connect. The connect-names form an exception list to the default caching specification.

Symbol Table Report (CREPORT 053)
The symbol table report lists information on symbol table load modules stored in the system dictionary DDLDCLOD area.

Sample CREPORT 053:

<table>
<thead>
<tr>
<th>REPORT NO. 53</th>
<th>CA IDMS/DC LOAD AREA REPORT</th>
<th>SYMBOL TABLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREPORT 053</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MODULE NAME</th>
<th>VERSION</th>
<th>NUM RLD ENTRIES</th>
<th>ENTRY PT ADDRESS</th>
<th>MODULE LENGTH</th>
<th>COMPILE DATE</th>
<th>COMPILE TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMI01D</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>5,036</td>
<td>mm/dd/yy</td>
<td>hhmmss</td>
</tr>
<tr>
<td>ADDOLFIX</td>
<td>1</td>
<td>8</td>
<td>0</td>
<td>6,088</td>
<td>mm/dd/yy</td>
<td>hhmmss</td>
</tr>
<tr>
<td>ADOL01D</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>7,976</td>
<td>mm/dd/yy</td>
<td>hhmmss</td>
</tr>
<tr>
<td>ADRP01D</td>
<td>1</td>
<td>14</td>
<td>0</td>
<td>2,768</td>
<td>mm/dd/yy</td>
<td>hhmmss</td>
</tr>
<tr>
<td>ANSI01D</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>6,144</td>
<td>mm/dd/yy</td>
<td>hhmmss</td>
</tr>
<tr>
<td>BIFX01D3</td>
<td>1</td>
<td>21</td>
<td>0</td>
<td>3,408</td>
<td>mm/dd/yy</td>
<td>hhmmss</td>
</tr>
<tr>
<td>BIFX02D</td>
<td>1</td>
<td>21</td>
<td>0</td>
<td>4,600</td>
<td>mm/dd/yy</td>
<td>hhmmss</td>
</tr>
<tr>
<td>CLSTAX1</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>3,424</td>
<td>mm/dd/yy</td>
<td>hhmmss</td>
</tr>
<tr>
<td>CLSTOX1</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>3,424</td>
<td>mm/dd/yy</td>
<td>hhmmss</td>
</tr>
<tr>
<td>CLSTRHDX</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>3,424</td>
<td>mm/dd/yy</td>
<td>hhmmss</td>
</tr>
<tr>
<td>CLSTRHDX</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>3,424</td>
<td>mm/dd/yy</td>
<td>hhmmss</td>
</tr>
</tbody>
</table>

Field Descriptions
A description of the fields in the sample report follows:

- **MODULE NAME**
  Name of the symbol table load module.

- **VERSION**
  Version number of the symbol table load module.

- **NUM RLD ENTRIES**
  Number of entries in the relocation dictionary (RLD) for the symbol table load module.

- **ENTRY PT ADDRESS**
  Entry point address of the symbol table load module.

- **MODULE LENGTH**
  Length, in bytes, of the object text for the symbol table load module.
- **COMPILE DATE**
  Date the symbol table load module was compiled (mm/dd/yy).

- **COMPILE TIME**
  Time the symbol table load module was compiled (hhmmss).

---

**System Options Reports (CREPORTs 011 and 025)**

System options reports provide detailed information on DC/UCF systems that have been defined to the dictionary. CREPORT 011 describes the options defined for executable systems; these systems are represented in the dictionary by the SYSMO-170 record. CREPORT 025 describes options defined for systems that have been defined but not generated; these systems are represented by the SYS-041 record. The fields and format of the two reports are the same.

Sample CREPORT 011:

```plaintext
REPORT NO. 11                  CA IDMS/DC SYSTEM GENERATION REPORT REL nn.n mm/dd/yy PAGE 1
CREPORT 011                  CA IDMS/DC SYSTEM OPTIONS - SYSGEN
SYSTEM NAME                  VERSION BUILDER GENERATION ID SYSTEM ID OPERATING SYSTEM DESCRIPTION
DCSYSTEM                   110 R TECHD110 TECHD110 05/MVS
DATE CREATED: mm/dd/yy    PREPARED BY: MET DATE LAST USED: mm/dd/yy REVISED BY: JOS
DC PARAMETERS:
STORAGE ALLOCATION:       DMA PROGRAM STORAGE STORAGE REENTRANT XA PROGRAM XA STORAGE XA REENTRANT RELOCATABLE
SIZE POOL(K) CUSHION(K) POOL(K) POOL(K) POOL(K) POOL(K) POOL(K) POOL(K) THRESHOLD(%) 0 1
32000 600 100 700 400 1000 50000 9000 100
RUNTIME ENVIRONMENT:      MAXIMUM - - - - - - - - SYSTEM RUN UNITS - - - - - - - - STORAGE ALTERNATE TICKER
TASKS MSGDICT SCRATCH/QUEUE SIGNON/DEST LOADER SECURITY SYSTEM/DEST PROTECTION PROT KEY INTERVAL
113 4 4 4 4 4 4 4 4 4 4 4 YES 0 1
AUTOMATIC - - - - - UNDEFINED LOADABLE - - - - - PRINTED PRINTER - - - - - RESOURCE TIMEOUT - -
NEW COPY SUBSCHEMAS MAPS TABLES DIALOGS ACCESS MODULES REP RET CHECKPOINT INTERVAL PROGRAM VERSION
YES YES YES YES YES YES YES YES 7 OFF OFF RHDCBYE 1
OLD ADSO PF KEYS OLM KEYS PAGE REL LOADLIST SCRATCH XA MULTIPLE ENCLAVE SHARING
YES YES YES YES YES NO YES NO YES
INTERNAL STG LOCK CALL DBIO EXTERNAL STG LOCK CALL DBIO
LIMITS: LIMIT LIMIT LIMIT LIMIT LIMIT LIMIT LIMIT LIMIT
OFF 0 0 0 0 0 0 0 0 0 0 0 0
OPERATING SYSTEM INTERFACE:
LOG TO Logging Information - Logging Information SVC
DATABASE FILE 1 COUNT 1 FILE 2 COUNT 2 DEVICE NUMBER
YES CMSLOGA 0 0 DOS ONLY 175
3270 PRINT KEY WTD DESCRIPTION CODES WTD ROUTE CODES
PF12 NONE 01,02,11
ERROR HANDLING:
INACTIVE RUNAWAY SYSTEM SYSTEM SYSTEM USER USERUSER ABBR DEADLOCK
INTERVAL INTERVAL DUMP TRACE ENTRIES TRACE ENTRIES SNAP INTERVAL
NONE 10 YES ON 9999 OFF 0 YES 1
SYSTEM INTERNALS:
ABEND DPE ECR RCE RLE STACKSIZE PRIMARY SECONDARY
STORAGE(WDS) COUNT LIST(WDS) COUNT COUNT (WDS) NULL POES NULL POES
1000 3000 226 4000 32000 2800 64 64
STATISTICS OPTIONS:
TIME INTERVAL TIME BY TASK BY TRAN BY LINE BY USER
0 0 YES YES NO YES COLLECT
DB PARAMETERS:
JOURNAL MAXIMUM AREA AREA RETRIEVAL UPDATE SCRATCH/QUEUE JOURNAL JOURNAL
RETRIEVAL SYSLocks RETRY LOCKING LOCKING JOURNAL BEFORE ONLY FRAGMENT TRANSACTION
NOJOUR 150000 1 FOREVER NOLOCK NOLOCK YES OFF 0
ON COMMIT WRITE COMT ON ROLLBACK RETAIN ID
```

---

16-Jan-2018 288/898
Field Descriptions

A description of the fields in the sample report follows:

- **SYSTEM NAME and VERSION**
  Identifies the DC/UCF system being described.

- **BUILDER**
  Identifies the builder code for the record that defines the system occurrence. For a description of builder codes, see the Dictionary Builder Codes table at the end of this section.

- **GENERATION ID**
  Identifies the unique identifier for the system, as defined in the system options table at startup.

- **SYSTEM ID**
  Indicates the name (nodename) by which the DC/UCF system is known to other nodes in the DC/UCF communications network.

- **OPERATING SYSTEM**
  Identifies the name of the host operating system under which the DC/UCF system runs.

- **DESCRIPTION**
  Identifies the system description defined to the dictionary.

- **DATE CREATED**
  Identifies the date the system occurrence was added to the dictionary.

- **PREPARED BY**
  Identifies the user who added the system occurrence to the dictionary.

- **DATE LAST USED**
  Identifies the date the system occurrence was last accessed.

- **REVISED BY**
  Identifies the user who last modified the system occurrence.

- **STORAGE ALLOCATION**
  Describes the storage allocation parameters:

  - **CWA SIZE**
    Identifies size, in kilobytes, of the Common Work Area (CWA).

  - **PROGRAM POOL(K)**
    Identifies the amount of storage, in kilobytes, that is available for loading nonresident programs, subschemas, maps, database procedures.
- **STORAGE CUSHION(K)**
  Identifies the amount of storage, in kilobytes, that is available in the storage pool for currently executing tasks.

- **STORAGE POOL(K)**
  Identifies the amount of storage, in kilobytes, that is provided for subschema work areas, COBOL working storage sections, user variable storage, packet-data-movement buffers, and SPF work areas (SPF users only).

- **REENTRANT POOL(K)**
  Indicates the size, in kilobytes, of an optional secondary program pool reserved for reentrant programs and tables (for example, subschemas, database procedures, and IDMS DC maps).

- **XA PROGRAM POOL**
  For systems supporting 31-bit addressing, identifies the size, in kilobytes, of the 31-bit program pool.

- **XA STORAGE POOL**
  For systems supporting 31-bit addressing, identifies the size, in kilobytes, of storage pool number 255 (the 31-bit storage pool).

- **XA REENTRANT POOL**
  For systems supporting 31-bit addressing, indicates the size, in kilobytes, of the 31-bit reentrant pool.

- **RELOCATABLE THRESHOLD (%)**
  Indicates that the system should write relocatable storage to the scratch area across a pseudo-converse when the amount of space used in the storage pool reaches the indicated percentage.

- **RUNTIME ENVIRONMENT**
  Describes the runtime parameters:

  - **MAXIMUM TASKS**
    Indicates the maximum number of user tasks that can be active concurrently. This number does not include external request units or the IDMS-DC system tasks. For users executing under the central version, the value includes tasks invoked by online IDD, the online subschema compiler, and CA OLQ.

  - **SYSTEM RUN UNITS**
    Indicates the number of system run units initiated at startup to service:

    - Message dictionary requests (MSGDICT)
    - Queue requests (SCRATCH/QUEUE)
    - Signon requests (SIGNON/DEST)
    - Dictionary load requests (LOADER)
    - Security requests on system-level resources (SECURITY)
    - Destination requests (SYSTEM/DEST)

  - **STORAGE PROTECTION**
    (IDMS DC only) Indicates whether the storage protection feature is enabled for the system.

  - **ALTERNATE PROT KEY**
    (IDMS DC only) Indicates the number of the alternate storage protect key used by the IDMS DC system storage protection feature.
- **TICKER INTERVAL**
  Indicates how frequently, in wall-clock seconds, the system checks for the occurrence of timer-related events.

- **AUTOMATIC NEW COPY**
  Indicates the action taken if, in loading a program from the load area, the system finds that the program has been deleted. YES indicates that the system attempts to load the program from the system load library. NO indicates that the system does not attempt to load the program from the system load library until a DCMT VARY PROGRAM NEW COPY or an IDMS CV operator VARY SUBSCHEMA NEW COPY is issued.

- **UNDEFINED LOADABLE**
  Indicates whether SUBSCHEMAS, MAPS, TABLES, DIALOGS, and ACCESS MODULES can be automatically defined to the dictionary at run time.

- **PRINTED REP RET**
  (IDMS DC only) Identifies the amount of time, in days, that the DC system retains a report in the scratch/queue area.

- **PRINTER CHECKPOINT**
  (IDMS DC only) Identifies the page count (printer checkpoints) for all active reports. A value of 0 or OFF indicates that interrupted reports are reprinted from the beginning. Any other value indicates that printing is resumed at the last checkpoint.

- **RESOURCE TIMEOUT INTERVAL**
  (IDMS DC only) Identifies the amount of time, in wall-clock seconds, that the IDMS DC system permits a terminal to be inactive before freeing the terminal resources.

- **RESOURCE TIMEOUT PROGRAM/VERSION**
  (IDMS DC only) Identifies the name and version number of the program invoked by IDMS DC to handle the resources of an inactive terminal that has exceeded the timeout interval.

- **OLQ**
  Indicates whether the system includes CA OLQ.

- **ADSO**
  Indicates whether the system includes CA ADS.

- **PF KEYS**
  Indicates whether the system includes at least one keys table.

- **OLM KEYS**
  Indicates whether a key table is defined for OLM.

- **PAGE REL**
  Indicates whether the system invokes operating system services when one or more virtual pages are no longer required and the contents of those pages need not be saved.

- **LOADLIST**
  Identifies the default load list to be used by the system when searching for programs.

- **XA SCRATCH**
  Indicates whether or not the scratch area (DDLDCSCR) uses a 31-bit storage pool.
• **MULTIPLE ENCLAVE**
  Indicates whether the programs in this system are eligible to use the same high level Language Environment process/enclave when multiple programs are executed in the same task.

• **TRAN SHARING**
  Specifies whether to activate the Transaction Sharing option for all tasks.

• **INTERNAL LIMITS**
  Indicates whether DC/UCF controls limits on all tasks defined to the system during system generation or at run time. ENABLED indicates that limits are enforced; DISABLED indicates that limits are not enforced except by means of a DCMT VARY LIMITS command. OFF indicates that limits are not enforced.

  - **STG LIMIT**
    Indicates the storage limit, in kilobytes, that a task can hold at one time.

  - **LOCK LIMIT**
    Indicates the limit of record locks that a task can set.

  - **CALL LIMIT**
    Indicates the limit of system service calls (for example, #GETSTG, #LOAD) a task can issue.

  - **DBIO LIMIT**
    Indicates the limit of database I/O operations (for example, reads and writes) that are performed for a task.

• **EXTERNAL LIMITS**
  Indicates whether DC/UCF controls limits on all tasks associated with external request units (that is, ERUS tasks). ENABLED indicates that limits are enforced; DISABLED indicates that limits are not enforced except by means of a DCMT VARY LIMITS command. OFF indicates that limits are not enforced.

  - **STG LIMIT**
    Indicates the storage limit, in kilobytes, that a task can hold at one time.

  - **LOCK LIMIT**
    Indicates the limit of record locks that a task can set.

  - **CALL LIMIT**
    Indicates the limit of system service calls (for example, #GETSTG, #LOAD) a task can issue.

  - **DBIO LIMIT**
    Indicates the limit of database I/O operations (for example, reads and writes) that are performed for a task.

• **OPERATING SYSTEM INTERFACE**
  Describes the parameters that define how the DC/UCF system and the operating system interact.

  - **LOG TO DATABASE**
    Indicates whether log records are written to the DDLDCLOG area of the dictionary.
- **LOGGING INFORMATION**
  Identifies the system log file and the maximum number of records that can be written to FILE 1, the primary file, and FILE 2, the alternate log file. If COUNT 1 is 0, a maximum number of records was not defined. If COUNT 2 is -1, an alternate log file was not defined. If COUNT 1 and COUNT 2 are both 0, log records are written to the DDLDLCLOG area of the dictionary.

- **LOG DEVICE**
  *(z/VSE only)* Identifies the device type for the log file.

- **SVC NUMBER**
  Identifies the number of the SVC used for communication between IDMS CV and external request units.

- **3270 PRINT KEY**
  Identifies the PF key used to print screen contents.

- **WTO DESCRIPTION CODES**
  *(z/OS only)* Identifies the z/OS operator message codes, which are used to identify the values supplied to the DESC parameter for write-to operator (WTO) macros issued by the system.

- **WTO ROUTE CODES**
  *(z/OS only)* Identifies the z/OS operator message routing codes, which are used to identify values for the ROUTCDE parameter for write-to-operator (WTO) macros issued by the system.

- **ERROR HANDLING**
  Describes error handling procedures.

- **INACTIVE INTERVAL**
  *(IDMS DC only)* Indicates the time, in wall-clock seconds, that the system allows an internal task to wait for a resource before abnormally terminating the task.

- **RUNAWAY INTERVAL**
  Indicates the maximum time, in wall-clock seconds, that the system allows a task or transaction to execute between interval waits before abnormally terminating the task or transaction.

- **SYSTEM DUMP**
  Indicates whether the DC/UCF system takes a memory dump for all system abend codes.

- **SYSTEM TRACE**
  *(IDMS DC only)* Indicates whether the IDMS DC system trace facility is enabled to trace system events during program development and debugging.

- **TRACE ENTRIES**
  *(IDMS DC only)* Indicates the number of entries allocated to the trace table buffer.

- **USER TRACE**
  *(IDMS DC only)* Indicates whether the user trace facility is enabled to trace program requests for IDMS DC system services.

- **USERTRACE ENTRIES**
  Indicates the number of entries allocated to the user trace buffer.
ABRU SNAP
Indicates whether a snap dump is written to the log when an external request unit terminates abnormally.

DEADLOCK INTERVAL
Indicates the amount of time, in wall-clock seconds, that elapses before the system searches for deadlocked tasks.

SYSTEM INTERNALS
Describe the internal characteristics of the system.

ABEND STORAGE(WDS)
(IDMS DC only) Indicates the amount of storage, in fullwords, available to the IDMS DC system for processing abends.

DPE COUNT
(IDMS DC only) Indicates the number of deadlock prevention elements (DPEs) allocated to the IDMS DC system at startup.

ECB LIST(WDS)
(IDMS DC only) Indicates the size, in fullwords, of the storage allocated for the Event Control Block list.

RCE COUNT
(IDMS DC only) Indicates the number of resource control elements (RCEs) allocated to the IDMS DC system at startup.

RLE COUNT
Indicates the number of resource link elements (RLEs) allocated to the DC/UCF system at startup.

STACKSIZE (WDS)
(IDMS DC only) Indicates the size, in fullwords, of the work storage stack within the task control element (TCE) of the IDMS DC system.

PRIMARY NULL PDES
Indicates the number of null program definition elements (PDEs) allocated at system startup for the automatic definition of programs not defined in PROGRAM statements. The default 0 indicates that automatic definition is disallowed.

SECONDARY NULL PDES
Indicates the number of additional PDEs that can be allocated from the storage pool when the primary null PDEs have been used.

STATISTICS OPTIONS
Describe how system statistics are logged.

TIME INTERVAL
Indicates how frequently (in seconds) histograms and system statistics are collected.

BY TASK
Indicates whether the DC/UCF system collects CPU-time statistics for each task.
- **BY TRAN**
  Indicates whether the DC/UCF system collects statistics on a transaction-by-transaction basis for all tasks.

- **BY LINE**
  Specifies whether the DC/UCF system collects by-line histograms.

- **BY USER**
  Indicates whether the DC/UCF system collects separate by-task CPU time statistics for system-mode time and user-mode time.

- **COLLECT/WRITE**
  Indicates whether the system is collecting by-task histograms or task statistics. COLLECT (the default) indicates collection of by-task histograms; WRITE indicates collection of task statistics.

- **DB PARAMETERS**
  Describe database access, journaling, and locking parameters.

- **JOURNAL RETRIEVAL**
  Indicates whether the system is writing BGIN and ENDJ checkpoints to the journal file for retrieval transactions.

- **MAXIMUM SYSLOCKS**
  Indicates the maximum number of record that the system is to maintain for all run units at a given time.

- **AREA THRESHOLD**
  Indicates the point at which, during ready processing, the system will begin to accumulate area locks for a database transaction. OFF directs the system not to accumulate area locks until the system can acquire all areas needed by a database transaction.

- **AREA RETRY**
  Indicates the number of times the system will continue trying to gain access to all areas without accumulating area locks. FOREVER directs the system to keep trying until it acquires all areas or until operating system resource and time limits are exceeded.

- **RETRIEVAL LOCKING**
  Indicates whether the system is to maintain locks automatically for records in areas accessed in shared retrieval mode.

- **UPDATE LOCKING**
  Indicates whether the system is to maintain locks automatically for records in areas accessed in protected update mode.

- **SCRATCH/QUEUE JOURNAL BEFORE ONLY**
  Indicates whether the system performs partial journaling. NO indicates that full journaling is taking place (that is, both before and after images are being written to the journal file).

- **JOURNAL FRAGMENT**
  Indicates the maximum number of journal blocks to write to the journal file before the system writes a dummy segment (DSEG) record to the journal file. OFF indicates that the journal fragment interval is off.
• **JOURNAL TRANSACTION**
  Indicates the number of active transactions that must be running in a DC/UCF system to defer the writing of a journal block.

• **ON COMMIT**
  Specifies options that control commit behavior for all tasks in the system. Valid values are:
  - **WRITE COMT**
    Writes a COMT journal record.
  - **WRITE ENDJ**
    Writes an ENDJ journal record.
  - **NEW ID**
    Assigns a new local transaction ID.
  - **RETain ID**
    Retains the existing local transaction ID.

• **ON ROLLBACK**
  Specifies options that control rollback behavior for tasks in this system. These options apply only to rollback operations in which the transaction is continued. Valid values are:
  - **RETAIN ID**
    Retains the existing local transaction ID on a rollback.
  - **NEW ID**
    Assigns a new local transaction ID on a rollback.

• **CV PARAMETERS**
  Describe abend handling and time parameters for the central version.

• **CHKUSER TASKS**
  (z/OS only) Indicates the number of tasks started by the system at run time to detect abnormally terminated z/OS batch transactions. The value also represents the maximum number of batch transactions that can access the database concurrently, overriding the MAXIMUM ERUS parameter.

• **CVNUMBER**
  Indicates the number of the DC/UCF system to the CA IDMS SVC.

• **EXTERNAL WAIT**
  Indicates the time, in wall-clock seconds, that the system waits for an external request unit to issue a database request before abnormally terminating the transaction.

• **INTERNAL WAIT**
  Indicates the time, in wall-clock seconds, that the system permits an external request unit to wait for a database or system resource before abnormally terminating the transaction.

• **MAXIMUM ERUS**
  Indicates the maximum number of external request units (ERUs) that can be active concurrently.
- **RUPRTY**: Indicates the default execution priority for all transactions.

- **SYSCTL PARAMETERS**: Describes parameters for the system control file used by programs executing outside the system region/partition.

- **DDNAME**: (z/OS only) Identifies the ddname of the system control (SYSCTL) file used by the system.

- **DBNAME**: Identifies the dictionary or database to which the system will route requests from programs using the SYSCTL file.

- **ALWAYS/DEFAULT**: Indicates which database/dictionary is accessed by programs at run time. ALWAYS indicates that programs always use the database named in the SYSCTL file regardless of IDMSOPTI or program specifications. DEFAULT indicates that programs use the database/dictionary named in the SYSCTL file only if a dictionary/database name is not specified in the IDMSOPTI module or in the program.

- **NODE NAME**: Identifies the name of a DC/UCF system defined to DC/UCF communications network to which the system will route requests from programs using the SYSCTL file.

- **ALWAYS/DEFAULT**: Indicates which node is accessed by programs at run time. ALWAYS indicates that programs use the node named in the SYSCTL file, regardless of the IDMSOPTI or program specifications. DEFAULT indicates that programs use the node named in the SYSCTL file only if a node is not specified in the IDMSOPTI module or in the program.

**Task Description Reports (CREPORTs 005, 020, and 021)**

Task description reports provide information on tasks that have been associated with a DC/UCF system. CREPORT 005 describes tasks associated with executable systems; these tasks are represented in the dictionary by the TASKLST-023 record. CREPORTs 020 and 021 describe tasks associated with systems that have been defined but not generated; these tasks are represented by the TASK-025 record.

The fields and format of the three task reports are almost identical. In CREPORTs 005 and 020, task/program relationships are listed in alphabetical order by task name. In CREPORT 020 task/program relationships are listed in alphabetical order by program name.

**Sample CREPORT 005:**

---

<table>
<thead>
<tr>
<th>REPORT NO.</th>
<th>CA IDMS/DC SYSTEM GENERATION REPORT</th>
<th>REL.</th>
<th>mm/dd/yy</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>05</td>
<td></td>
<td>nn.n</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>CREPORT 005</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SYSTEM NAME**: DCSYSTEM

**SYSTEM VERSION**: 110

**TASK DESCRIPTION**

---

297/898
Field Descriptions

A description of the fields in the sample report follows:

- **SYSTEM NAME and VERSION**
  
  Identify the name and version of the system associated with the tasks being described.

- **TASK and TASK VERS**
  
  Identify the name and version number of the task being described.

- **PROGRAM and PROG VERS**
  
  Identify the name and version number of the initial program invoked by the system for the task being described.

- **BUILDER**
  
  Identifies the builder code for the record that defines the task occurrence. For a description of builder codes, see the Builder Codes table at the end of this section.

- **ACTION**
  
  Indicates whether the task occurrence has been updated (U) or deleted (D) or remains unchanged (blank) since the most recent generate of the system.

- **TCF TASKCODE**
  
  Identifies the code that invokes the transfer control facility (TCF) control program (RHDCUMBR) under which this task will run.

- **TCF VERS**
  
  Identifies the version of the TCF task code.

- **PRIORITY**
  
  Identifies the dispatching priority for the task.
INPUT
Indicates whether the terminal input buffer associated with the task contains data in addition to the task code at run time.

MAP
Indicates whether a mapout operation is performed automatically when the task is invoked.

INTERNAL
Indicates whether the task can be invoked internally and/or externally. YES indicates that the task can only be invoked internally; NO indicates that the task can be invoked internally or externally.

DISABLED
Indicates whether the task is disabled when the system is started.

SAVE SCREEN
Indicates whether screen contents associated with the task are saved before an immediate-write data stream is written to the terminal.

STG LOC
Indicates whether programs that run under the task can reside anywhere in the DC/UCF region (ANY) or whether programs must reside below 16 megabytes (BELLOW).

PROTOCOL
Indicates the response protocol to be used by the task when communicating with terminals associated with a VTAMLIN type line. Possible values are DEFRESP and EXPRESP.

RESOURCE TIMEOUT PROGRAM/ VERS
Identifies the name and version of the resource timeout program, which the system invokes to handle the resources owned by an inactive terminal following the expiration of the resource timeout interval.

RESOURCE TIMEOUT INTERVAL
Specifies the amount of time the system is to permit a terminal to be inactive before invoking the terminal resource program. SYSTEM directs the system to use the INTERVAL IS value specified on the RESOURCE TIMEOUT parameter of the SYSTEM system generation statement.

AREA ACQUIRE RETRY
Identifies the limit on the number of times the system will continue trying to gain access to all areas without accumulating area locks. FOREVER directs the system to keep trying until it acquires all areas or until operating system resource and time limits are exceeded. SYSTEM directs the system to use the value specified on the AREA ACQUISITION THRESHOLD parameter of the SYSTEM system generation statement.

AREA ACQUIRE THRESHOLD
Indicates the number of times, during ready processing, that the system will wait on an area lock before it starts to accumulate area locks for a transaction. OFF directs the system not to accumulate area locks. SYSTEM directs the system to use the values specified in the AREA ACQUISITION THRESHOLD parameter of the SYSTEM system generation statement.

MAX TASKS
Indicates the limit of maximum concurrent threads for a task. OFF indicates the system does not limit the number of concurrent threads.
- **INACTIVE INTERVAL**
  Indicates the amount of time the system is to permit an internal task to wait for a resource before abnormally terminating the task. SYSTEM directs the system to use the value specified on the INACTIVE INTERVAL parameter of the SYSTEM system generation statement.

- **RESOURCE LIMIT CALLS**
  Indicates the limit of system service calls (for example, OBTAIN CALC), that can be issued by an online task. SYSTEM directs the system to use the limit specified in the CALL LIMIT FOR ONLINE TASKS parameter on the SYSTEM system generation statement.

- **RESOURCE LIMIT DBIO**
  Indicates the limit of database I/O operations (reads and writes) that can be issued by an online task. SYSTEM directs the system to use the limit specified in the DBIO LIMIT FOR ONLINE TASKS parameter on the SYSTEM system generation statement.

- **RESOURCE LIMIT LOCKS**
  Indicates the limit of record locks allocated to an online task during the life of the task. SYSTEM directs the system to use the limit specified in the LOCK LIMIT FOR ONLINE TASKS parameter on the SYSTEM system generation statement.

- **RESOURCE LIMIT STORAGE**
  Indicates the limit of storage, in kilobytes, that an online task can hold at one time. SYSTEM directs the system to use the limit specified in the STORAGE LIMIT FOR ONLINE TASKS parameter on the SYSTEM system generation statement.

- **PRODUCT CODE**
  Indicates a generic name for a product and related task codes.

- **PRINT KEY**
  Indicates the print-screen key assignment. SYS directs the system to use the print key assignment specified on the PRINT KEY parameter of the SYSTEM system generation statement.

- **ON COMMIT**
  Specifies options that control commit behavior for the task. Valid values are:
  - **WRITE COMT**
    Writes a COMT journal record.
  - **WRITE ENDJ**
    Writes an ENDJ journal record.
  - **NEW ID**
    Assigns a new local transaction ID.
  - **RETain ID**
    Retains the existing local transaction ID.

- **ON ROLLBACK**
  Specifies options that control rollback behavior for tasks in this system. These options apply only to rollback operations in which the transaction is continued. Valid values are:
  - **RETAIN ID**
    Retains the existing local transaction ID on a rollback.
- **NEW ID**
  Assigns a new local transaction ID on a rollback.

- **TRAN SHARING**
  Specifies whether to activate the Transaction Sharing option for the task or whether the option for the task is based on the SYSTEM statement (SYSTEM).

### Builder Codes

The following Dictionary Builder Codes table presents the builder codes that are referenced in the DREPORTs and CREPORTs.

<table>
<thead>
<tr>
<th>Builder Code</th>
<th>Input Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>DDDL compiler</td>
</tr>
<tr>
<td>S</td>
<td>Schema compiler</td>
</tr>
<tr>
<td>C</td>
<td>DC/UCF mapping compilers</td>
</tr>
<tr>
<td>V</td>
<td>Subschema compiler</td>
</tr>
<tr>
<td>G</td>
<td>DC/UCF system generation compiler, before GENERATE command</td>
</tr>
<tr>
<td>R</td>
<td>DC/UCF system generation compiler, after GENERATE command</td>
</tr>
<tr>
<td>M</td>
<td>DML processors</td>
</tr>
<tr>
<td>A</td>
<td>ADS/ONLINE dialog generator</td>
</tr>
<tr>
<td>X</td>
<td>IDMSDIRL utility</td>
</tr>
</tbody>
</table>

**Note:** The builder code identifies the component that defined the entity occurrence to the dictionary or the component that last updated the entity occurrence.
IDMS Dictionary Migrator Reports

This section describes and illustrates the nine reports produced by CA IDMS Dictionary Migrator. Each report tracks a different activity performed by CA IDMS Dictionary Migrator. You receive reports with detail and summary results of an extraction, comparisons of the extractions to the object dictionary, parameter input and dictionary verifications, a summary of syntax production difficulties, and a listing of syntax statements produced by CA IDMS Dictionary Migrator.

- CA IDMS Dictionary Migrator Reports (see page 303)
  - Parameter Verification Report (see page 304)
  - Source Dictionary Verification Report (see page 304)
- Extract Summary and Extract Detail Reports (see page 305)
  - Extract Summary Report (see page 306)
  - Extract Detail Report (see page 307)
- Comparison Reports (see page 308)
  - Entity Discrepancy Report (see page 308)
  - Entity Cross-Reference Report (see page 309)
  - Syntax Production Report (see page 310)
  - Syntax Files Display Report (see page 310)
  - Catalog Navigation Report (see page 311)

When reviewing the reports produced by CA IDMS Dictionary Migrator, you should remember the attributes of a successful migration:

- Everything needed to recreate an entity in another environment is migrated or accessible.
- Introduction of the migrated entities will not adversely affect anything already in the target dictionary.

The CA IDMS Dictionary Migrator reports are descriptions of what is produced after a migration. Each report tracks a different activity. Reports contain descriptions of errors, problems, or anything adversely affected in the target dictionary as well. The following is a table of the CA IDMS Dictionary Migrator reports and their summaries:

<table>
<thead>
<tr>
<th>Report</th>
<th>Report Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Verification Report</td>
<td>Displays the parameter statements entered for an execution of CA IDMS Dictionary Migrator after the syntax of the statements has been checked.</td>
</tr>
<tr>
<td>Source Dictionary Verification Report</td>
<td>Provides information on the source dictionary after it has been checked for entities.</td>
</tr>
<tr>
<td>Extract Summary Report</td>
<td>Provides information on the total number of transactions that will take place for the upload to the target dictionary.</td>
</tr>
<tr>
<td>Extract Detail Report</td>
<td>Provides information on each entity that will be extracted from the source dictionary and migrated to the target dictionary.</td>
</tr>
</tbody>
</table>
### CA IDMS Dictionary Migrator Reports

CA IDMS Dictionary Migrator produces nine comprehensive reports:

- Parameter Verification Report
- Source Dictionary Verification Report
- Extract Summary Report
- Extract Detail Report
- Entity Discrepancy Report
- Entity Cross-Reference Report
- Syntax Production Report
- Syntax Files Display Report
- Catalog Navigation Report

From the information in the reports you can preview the migration to check parameter statements, the total number of transactions that will occur during the migration, and the detail of each transaction. You can also use the reports to compare the extracted entities to the object dictionary and review any discrepancies. If any difficulties were encountered in the production of syntax, you can use the reports to identify corrective measures. The reports give you complete control of the migration.

The following pages describe the functions and show samples of CA IDMS Dictionary Migrator reports.
Parameter Verification Report

The Parameter Verification Report displays the parameter statements entered for an execution of CA IDMS Dictionary Migrator after the syntax of the statements has been checked. The statement, as you entered it, is displayed in the right column. If there are errors, they will be marked. Informative, warning, and error messages appear in the left column of the report.

The following report shows a sample Parameter Verification Report.

The Parameter Verification Report contains:

- **release number** -- the version of CA IDMS Dictionary Migrator that was installed, where nn.nn represents the release number and the subrelease number.
- **message code** -- the eight-character code of an informative or error message.
- **message text** -- information that explains the message code.
- **parameter statement** -- display of parameter statement as you have entered it.

Source Dictionary Verification Report

The Source Dictionary Verification Report provides information on the source dictionary after it has been checked for entities. If the entities you requested for an extraction do not appear in the source dictionary, this report shows informative messages about those entities. This report also provides information on referenced entities within processes or maps.

For example, The following report shows a sample Source Dictionary Verification Report that provides messages about components related to other components that are being migrated. The indicated subschema does not have a corresponding load modules in the load area of the dictionary.

The Source Dictionary Verification Report contains:

- **entity type** -- the type of entity referenced within a process or map.
- **entity name** -- the name of the entity referenced within a process or map.
- **version number** -- the version number of the entity referenced within a process or map.

- **message code** -- the eight-character code of an informative or error message.

- **message text** -- information that explains the message code. Detailed information on the message--reasons and appropriate actions--is provided in the section "Messages".

The Source Verification report will contain messages related to item 1, finding all of the necessary pieces. If components of an entity cannot be found, or if they appear discrepant with the entity, a message will be produced. The Source Verification Report is formatted as an exception report, only the deficiencies are listed.

Some of the likeliest conditions are:

1. A message is not found. See the section "Parameters" for conditions where you might not want to migrate messages. Take into account that by not following the suggestions listed might result in a large number of warning messages produced.

2. A source component has been changed since it was last compiled. This can occur anytime a change is made without a subsequent regeneration. However, it may indicate that a noteworthy change has been made, but not yet compiled.

3. An entity is not found. Depending on the message, this may be an entity named in the parameter's EXTRACT statement or some component of another entity. In either case, it indicates that something that is needed for the migration was not found and will not be migrated, which may cause the migration to be incomplete.

---

### Extract Summary and Extract Detail Reports

These reports provide information on the extent of what will be migrated. The Extract Detail Report lists every entity which is related to the entity(s) named in the Extract statement(s). This is known as the "Extract Set." For each entity, its status for the migration listed:

- New (additions)
- Changed (modified)
- Dropped (not migrated)

An entity is dropped when it is presumed to be the same in the source dictionary.

The Extract Summary Report gives totals by entity type and utility.
The Extract Detail Report provides information on each entity that will be extracted from the source dictionary and entities for which delete syntax is created.

**Extract Summary Report**

The Extract Summary Report provides information on the total number of transactions that will take place for the upload to the CA IDMS utilities and compilers. You can use this report before an actual migration to plan for dictionary space.

The following report shows a sample Extract Summary Report. The report contains:

- **CA IDMS utility or compiler name**
- **transaction type** -- the type of transaction that occurred for the utility or compiler.
- **transaction total** -- the number of transactions for each utility or compiler.
- **transaction type total** -- the number of transactions for each transaction type.
- **change summary** -- indicates the number of add, change (modify), and drop (not migrated) transactions.

<table>
<thead>
<tr>
<th>CA TOOLS RELEASE</th>
<th>CA IDMS/Dictionary Migrator EXTRACT SUMMARY REPORT DATE mm/dd/yy</th>
<th>TIME hh:mm:ss</th>
<th>PAGE nnnn</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTRACTION CATEGORIES</td>
<td></td>
<td>TOTAL</td>
<td>CHANGE</td>
</tr>
<tr>
<td>IDMSDDDL TRANSACTIONS</td>
<td>522</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MESSAGE DELETES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DICTIONARY LOAD DELETES</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DIALOG DELETES</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PROGRAM DELETES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SCHEMA DELETES</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>TABLE DELETES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>QFILE DELETES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PROCESS DELETES</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MODULE DELETES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PGROUP DELETES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RECORD DELETES</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>REPORT DELETES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TRANSACTION DELETES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FILE DELETES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ELEMENT DELETES</td>
<td>228</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSTEM DELETES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ATTRIBUTE DELETES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>USER DELETES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CLASS DELETES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CLASS UPDATES</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>USER UPDATES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ATTRIBUTE UPDATES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSTEM UPDATES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ELEMENT UPDATES</td>
<td>228</td>
<td>228</td>
<td>0</td>
</tr>
<tr>
<td>MODULE UPDATES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PROCESS UPDATES</td>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>QFILE UPDATES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TABLE UPDATES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FILE UPDATES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RECORD UPDATES</td>
<td>20</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>REPORT UPDATES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TRANSACTION UPDATES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PROGRAM UPDATES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DIALOG UPDATES</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>APPLICATION UPDATES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DLOAD MODULE UPDATES</td>
<td>7</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>MESSAGE UPDATES</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>IDMSCHM TRANSACTIONS</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Extract Detail Report

The Extract Detail Report provides information on each entity that will be extracted from the source dictionary and entities for which delete syntax is created. The report displays the entity's type, name, version number, source dictionary, and dds node.

You can use this report to review every component of a system. The review is very helpful when you have large systems in development and need to know all of their components before a migration. You can review the Extract Detail Report to determine which components should be moved from testing into production.

You can also limit the information on this report by using the NOPRINT parameter. See the section "Parameters" for complete details.

The following report shows a sample Extract Detail Report. The report contains:

- **entity type** -- the type of entity that will be extracted.
- **entity name** -- the name of the entity that will be extracted.
- **version number** -- the version from which the entity will be extracted.
- **dictionary name** -- the name of the source dictionary from which the entity will be extracted.
- **change-status** -- indicates which entities will be added, changed (modified), or dropped (not migrated).

<table>
<thead>
<tr>
<th>ENTITY-TYPE</th>
<th>ENTITY-NAME</th>
<th>VERSION</th>
<th>CHANGEONLY-STATUS</th>
<th>FROM DICTIONARY/NODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOAD-DELETE</td>
<td>AAMAP1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD-DELETE</td>
<td>AAX</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD-DELETE</td>
<td>EMPSSLR</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD-DELETE</td>
<td>EMPSS01</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD-DELETE</td>
<td>QALR845S</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD-DELETE</td>
<td>QALR075S</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD-DELETE</td>
<td>QALR175S</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIALOG-DELETE</td>
<td>AAX</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIALOG-DELETE</td>
<td>EMPSSLR</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUBSCHEMA-DELETE</td>
<td>EMPSS01</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUBSCHEMA-DELETE</td>
<td>QALR845S</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUBSCHEMA-DELETE</td>
<td>QALR075S</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUBSCHEMA-DELETE</td>
<td>QALR175S</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCHEMA-DELETE</td>
<td>EMPSCHM</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAP-DELETE</td>
<td>AAMAP1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROCESS-DELETE</td>
<td>AAX-PREMAP-1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROCESS-DELETE</td>
<td>AAX-RESPONSE-1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Comparison Reports

CA IDMS Dictionary Migrator produces two reports--Entity Discrepancy Report and Entity Cross-Reference Report--that are the results of comparing source dictionary extractions to the object dictionary. The reports show where an extracted entity is used in the object dictionary.

When CA IDMS Dictionary Migrator extracts an entity from the source dictionary, it checks the object dictionary for that entity. Once the entity is located, it is checked for use by other entities in the object dictionary. If these entities appear in the list of extracted entities, the results of the search and comparison are provided on the Entity Cross-Reference Report. If the entity is used in the object dictionary by entities that do not appear in the list of extracted entities, their inconsistencies and the impact of migrating that entity are provided on the Entity Discrepancy Report.

Entity Discrepancy Report

The Extract Discrepancy Report addresses the second part of a successful migration, avoiding detrimental impact on other entities in the target dictionary. The messages in this report show the name of the migrated entity and the name of another entity that may be affected if the migration is done.

The sample reports shown in The following two reports illustrate how comparisons and inconsistencies are reported by CA IDMS Dictionary Migrator.

The Entity Discrepancy Report contains:

- **entity type** -- the type of entity that will be extracted.

- **entity name** -- the name of the entity that will be extracted.

- **version number** -- the version from which the entity will be extracted.
• **error message** -- the eight-character code and message text. Detailed information on the message (reasons and appropriate actions) is provided in the section "Messages."

### Entity Cross-Reference Report

The Entity Cross-Reference Report displays the extracted entities that are currently used in the object dictionary. The relationships of the extracted entities are reported in the sample Entity Cross-Reference Report in the following report.

If an extracted entity does not appear in either the Entity Cross-Reference Report or the Entity Discrepancy Report, the entity is not currently used by entities in the object dictionary.

The report can be omitted using the NOXREF parameter, or you can limit the amount of information on the Entity Cross-Reference Report by using the NOPRINT parameter. See the section "Parameters" for complete details.

The Entity Cross-Reference Report contains:

- **extract entity** -- the type of entity extracted from the source dictionary
- **entity name** -- the name of the entity in the source and object dictionaries
- **version** -- the version number of the extracted entity
- **related entity** -- the type of entity that uses the extracted entity in the object and source dictionaries
- **related entity name** -- the name of the entity that uses the extracted entity in the object and in the source dictionaries
- **version** -- the version of the related entity

<table>
<thead>
<tr>
<th>nn.nn</th>
<th>ENTITY CROSS-REFERENCE REPORT</th>
<th>mm/dd/yy</th>
<th>hh:mm:ss</th>
<th>nnnn</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTITY CROSS-REFERENCE REPORT IN OBJECT DICTIONARY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EXTRACT-ENTITY ELEMENT</th>
<th>ENTITY-NAME</th>
<th>VERSION</th>
<th>RELATED-ENTITY</th>
<th>RELATED-ENTITY-NAME</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELEMENT</td>
<td>AT-SCR-BEGIN-REPLAY</td>
<td>1</td>
<td>RELATES TO ENTITY</td>
<td>AT-WORK-RECORD-BEGIN</td>
<td>1</td>
</tr>
<tr>
<td>ELEMENT</td>
<td>AT-SCR-DIALOG-NAME</td>
<td>1</td>
<td>RELATES TO ELEMENT</td>
<td>AT-WORK-RECORD-BEGIN</td>
<td>1</td>
</tr>
<tr>
<td>ELEMENT</td>
<td>AT-SCR-DIALOG-NAME-ALT</td>
<td>1</td>
<td>RELATES TO RECORD</td>
<td>AT-LINK-RECORD</td>
<td>1</td>
</tr>
<tr>
<td>ELEMENT</td>
<td>AT-SCR-DIALOG-VER</td>
<td>1</td>
<td>RELATES TO RECORD</td>
<td>AT-LINK-RECORD</td>
<td>1</td>
</tr>
<tr>
<td>ELEMENT</td>
<td>AT-SCR-ERROR-LIT</td>
<td>1</td>
<td>RELATES TO ELEMENT</td>
<td>AT-SCR-HOLD</td>
<td>1</td>
</tr>
</tbody>
</table>
Syntax Production Report

The Syntax Production Report displays messages reflecting difficulties that have occurred in the creation of syntax. The messages will alert you to conditions which result in discontinuation of the migration process, and also indicate where the unsatisfactory conditions originate.

If messages do not appear in the Syntax Production Report, then syntax was successfully created for all entities that were extracted.

The following report shows a sample Syntax Production Report. The report contains:

- **message code** -- the eight-character code of an error message.

- **message text** -- information that explains the message code. Detailed information on the message--reasons and appropriate actions--is provided in the section "Messages".

Syntax Files Display Report

The Syntax Files Display Report shows the contents of each syntax file that will be used by the CA IDMS utilities and compilers or the Computer Associates utility to populate the object dictionary. This report is a hard copy of the syntax files produced by CA IDMS Dictionary Migrator. The Syntax Files Display Report is useful for verifying and documenting migrated entities.

The following report shows pages from a sample Syntax Files Display Report. The report contains:
Catalog Navigation Report

The Catalog Navigation Report provides information on each SQL table, view, schema, or table procedure that will be extracted from the source catalog and for which SQL CREATE syntax statements are generated. This report also highlights any error conditions encountered during the SQL catalog extraction process.

For more information on the message(reasons and appropriate actions), see the section "Messages."

The report contains:

- **message-code** -- the eight-character code of an informative or error message.

- **message text** -- information that explains the message. Detailed information on the message.
| USMS010I EXTRACT CONSTRAINT - SCHEMA: DEMO, CONSTRAINT: DEPT_BUDGET |
| USMS002I EXTRACT TABLE - SCHEMA: DEMO, TABLE: BUDGET |
| USMS011I EXTRACT TABLE PROCEDURE - SCHEMA: DEMO, PROCEDURE: TESTPROC_1 |
| USMS012I EXTRACT KEY - KEYNAME: PROC1_PRIME_KEY, ON SCHEMA: DEMO, PROCEDURE: TESTPROC_1 |
| USMS012I EXTRACT KEY - KEYNAME: PROC1_UNIQ_KEY, ON SCHEMA: DEMO, PROCEDURE: TESTPROC_1 |
| USMS012I EXTRACT KEY - KEYNAME: PROC1_STD_KEY, ON SCHEMA: DEMO, PROCEDURE: TESTPROC_1 |
| USMS001I CATALOG NAVIGATION COMPLETED |
IDMS Log Analyzer Reports

CA IDMS Log Analyzer, a CA IDMS database analysis and management tool, produces a variety of reports that present useful information on database utilization. All of the CA IDMS Log Analyzer reports are generated from information written to the CA IDMS Log. The CA IDMS Log Analyzer assumes if the year is greater than 69 the century is 19; if the year is less than 69 the century is 20.

- What is CA IDMS Log Analyzer? (see page 313)
- CA IDMS Log Analyzer Parameters (see page 317)
- Customizing the Billing Reports (see page 333)
- CA IDMS Log Analyzer Operations (see page 335)
- USLBILX and USLRPT5 Source Code (see page 342)
- External Request Element Extension (see page 342)
- Billing the Record File (see page 343)
- I/O Modules (see page 344)

What is CA IDMS Log Analyzer?

CA IDMS Log Analyzer, a CA IDMS performance analysis and management tool, records information taken from the CA IDMS Log and produces a variety of database analysis reports that gauge resource use and system performance.

CA IDMS Log Analyzer reports provide a clear, accurate, and up-to-date picture of resource use at your installation. This database utility also answers many other system management questions that will help you fine-tune your CA IDMS environment and help make users accountable for database use.

Users who are familiar with CA IDMS Journal Analyzer will recognize certain CA IDMS Log Analyzer reports because they closely resemble some of the reports available through CA IDMS Journal Analyzer. But they are generated using statistics from the Archived Log rather than the Journal File. This means you can get some of the same management information from CA IDMS Log Analyzer without the system overhead associated with reading statistics from the Journal File.

In addition, CA IDMS Log Analyzer uses information from the Log File that is not contained in the Journal File. With CA IDMS and the CA IDMS Log Analyzer version of the CA IDMS SVC exit, billing-related information that can be helpful in a chargeback situation is contained on the log. CA IDMS Log Analyzer takes this information, ties it to a particular user, and produces billing-related reports that can augment billing/chargeback and budgeting procedures at your installation. Source code for this SVC exit and these reports enables you to tailor them to meet your specific needs. Billing data also is available as a separate file. You can customize the CA IDMS Log Analyzer Billing Record File to meet existing system requirements by modifying source code supplied by CA.

For more information, see the following topics:
- How CA IDMS Log Analyzer Improves Productivity (see page 314)
- Reports and Their Functions (see page 314)
- Reports Produced By CA IDMS Log Analyzer (see page 315)
How CA IDMS Log Analyzer Improves Productivity

CA IDMS Log Analyzer reports will expand your management perspective and allow you to improve productivity at your CA IDMS installation.

Using CA IDMS Log Analyzer reports, you can establish system controls with discretion, forecast trends concerning use of the information resource and gain a better understanding of CA IDMS performance capabilities. Specifically, CA IDMS Log Analyzer reports:

- Identify the user, transaction, terminal, or account number that is associated with database use.
- Present CA IDMS statistics in a format you can use to measure and evaluate database resource consumption and system performance.
- Provide quantitative statistics on I/Os, CPU cycles, total run-units that give you a picture of total system efficiency.

Reports and Their Functions

Contents

- Billing Reports (see page 314)
- Billing Record File (see page 314)
- Program Reports (see page 315)
- Management Reports (see page 315)
- Audit Report (see page 315)

Billing Reports

Billing Reports use statistics from the CA IDMS Log to produce four reports that can serve as a functional model for building an effective billing system in your environment. Depending on the parameters you select, run-unit activity can be tied to a specific user, transaction, terminal, or account number. These reports can be customized to meet your unique needs.

- Billing Details Report
- Billing Summary Report
- Billing System Summary Report
- Billing Grand Summary Report

Billing Record File

This file can be input to your existing billing/chargeback system and can also be customized to meet your unique requirements.
Program Reports

Program Reports use statistics from the CA IDMS Log to offer detailed and summarized reports that show how efficiently your application programs and dialogs are using CA IDMS.

- Program Details Report
- Program Summary Report
- Program System Summary Report
- Program Grand Summary Report

Management Reports


- Management Highlights/Summary Reports
  - Highlights Program Summary Report
  - Highlights System Summary Report
  - Highlights Grand Summary Report
- Management Highlights/Buffer Pool Utilization Report
- Management Ranking Report

Audit Report

Generated dynamically, the Audit Report summarizes processing, lists messages received during an execution of CA IDMS Log Analyzer, and lists all the parameters you supplied.

Reports Produced By CA IDMS Log Analyzer

Contents

- Three Types of CA IDMS Log Analyzer Reports (see page 316)
  - Reports on CA ADS Dialogs (see page 316)
  - Audit Report Summarizes Processing (see page 316)
  - Using the Billing Record File (see page 316)
  - Customizing the Billing Record File (see page 316)

CA IDMS Log Analyzer generates a total of 13 log reports and an audit report, and it also creates a Billing Record File. This database utility is parameter-driven: you control the output by supplying the proper parameters. All CA IDMS Log Analyzer reports use statistics from the CA IDMS Log to produce reports.
CA IDMS Log Analyzer reports can optionally show all log record times in UTC time. By default all log records are shown in Local time.

Three Types of CA IDMS Log Analyzer Reports

CA IDMS Log Analyzer produces three major types of log reports, including:

Billing Reports—Four Billing Reports relate CA IDMS statistics to the user(s) of the database. While CA IDMS Log Analyzer is not a billing package, the reports provide useful CA IDMS statistics that are tied to job accounting data for batch users and to terminal ID, user ID, or transaction ID for CICS, CA IDMS/DC, or CA ADS dialog transactions. These reports are open-ended, because CA also provides source code that allows you to tailor them to the existing billing system in your environment. See Customizing the Billing Reports.

Management Reports—Five Management Reports present information on CPU cycle

Program Reports—Four Program Reports contain both detailed and summarized information by application program or dialog. These reports provide statistics on pages read, ratios, counts, and other significant CA IDMS statistics.

Reports on CA ADS Dialogs

CA ADS dialogs are included in the reports, listed as online programs. The statistics generated are the same as the statistics for programs. CA IDMS Log Analyzer also reports overhead records for CA ADS, containing system usage statistics that cannot be attributed to any specific dialog. The overhead records are reported as separate programs (named $ADS@@OH and $ADS@@AO). If you are using these records for billing/chargeback purposes, you should divide the usage in these reports proportionately among the dialogs executed by each user.

Audit Report Summarizes Processing

The Audit Report monitors and summarizes CA IDMS Log Analyzer processing, lists all messages generated during each execution, and also lists the parameters you specified.

Using the Billing Record File

The Billing Record File collects database utilization information from the CA IDMS Log. This file can be used as input to an existing billing system in your environment. This file adds new flexibility to the CA IDMS environment by providing CA IDMS statistics such as CPU cycle information and I/Os, then ties this information to user ID by transaction, terminal, or operator.

Customizing the Billing Record File

The module that produces the Billing Record File is supplied as source code. This functional billing model can be customized to meet your particular needs, as dictated by your billing/chargeback system, and then used as input to your in-house billing system.

SVC User Exit Module

The SVC User Exit Module is a feature provided by CA IDMS Log Analyzer that helps control data from the CA IDMS Log.
Invoked in the CA IDMS SVC as a BIND RUN-UNIT is processed, this module makes it possible for you to not only log information on run-unit execution for inclusion in the billing file, but also to log information that relates run-unit information back to the specific batch job or TP task from which it originated.

As supplied, the module will capture user identification information (by terminal, transaction, or operator for each CICS transaction, or by account for each batch job) so that it can be written to the CA IDMS Log. Then CA IDMS statistics can be tied back to other statistics for the same user transaction or program.

The module is supplied as source code. This means you can use the identification information selected by CA or you can tailor it to meet your organization's needs. When combined with the ability to customize the billing reports and the billing file, this exit module gives you control over the content and format of statistical (billing) data taken from the CA IDMS Log.

**CA IDMS Log Analyzer Parameters**

This section describes the parameters needed to produce the CA IDMS Log Analyzer. There are two primary parameters: PROCESS and REPORT. The COMMENTS option allows you to include notes, observations, or comments with parameter statements.

Two parameters control CA IDMS Log Analyzer output: PROCESS and REPORT. The PROCESS parameter initiates CA IDMS Log Analyzer processing. The REPORT parameter specifies which CA IDMS Log Analyzer report is to be printed. The COMMENTS option gives you the ability to store comments, observations, or notes about certain reports and their uses with the parameter statements that request those reports.

**PROCESS**

The PROCESS parameter is mandatory and should precede all report parameters. It supplies certain global parameters that initiate all processing performed by CA IDMS Log Analyzer.

**REPORT**

The REPORT parameter specifies which type of CA IDMS Log Analyzer report is to be created and defines the data that is to be printed. Up to 20 reports can be requested for each execution of CA IDMS Log Analyzer.

For more information, see the following topics:

- Parameters and Their Uses (see page 318)
- Billing Reports/Billing Record File Parameters (see page 320)
- Program Report Parameters (see page 324)
- Management Highlights Summary Report Parameters (see page 326)
- Buffer Pool Utilization Report Parameters (see page 328)
- Management Ranking Report Parameters (see page 329)
Parameters and Their Uses

COMMENTS Option

The COMMENTS option included in CA IDMS Log Analyzer is designed for your convenience and can be used at your discretion. When you place an asterisk (*) in the first column of a parameter statement, you can insert notes, observations, comments on reports and their uses, or any other information that will be helpful for future reference. CA IDMS Log Analyzer parameter syntax follows the notation conventions shown in Table 3-1 and syntax rules shown in Table 3-2. Please review these conventions and rules.

Notation Conventions for Parameter Statements:

<table>
<thead>
<tr>
<th>Example</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESS</td>
<td>All keywords are written in UPPERCASE. Those portions of the keyword that must be entered are UNDERSCORED. When part of a keyword is not underscored, you may omit it without altering the meaning of the statement.</td>
</tr>
<tr>
<td>CONTINUE= YES</td>
<td>A keyword phrase is made up of a major keyword followed by an equal sign (=), followed by a minor keyword or a variable. A keyword phrase cannot be split between two parameter cards.</td>
</tr>
<tr>
<td>INTERVAL=60, RANK=4</td>
<td>Variables appear in lowercase italic. Substitute an appropriate value for each variable if the L=interva keyword phrase is required.</td>
</tr>
<tr>
<td>[NAME= name]</td>
<td>Brackets indicate optional keyword phrases. If you omit the entire parameter, CA IDMS Log Analyzer will supply a default value.</td>
</tr>
<tr>
<td>Points to the default in a list of choices.</td>
<td></td>
</tr>
<tr>
<td>FILE= {NO YES}</td>
<td>Braces enclose two or more options in a column. You must choose one of them.</td>
</tr>
</tbody>
</table>

Parameter Syntax Rules:

<table>
<thead>
<tr>
<th>Item</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order of Parameter Statements</td>
<td>Parameter statements are free-form. They can be entered in any order.</td>
</tr>
<tr>
<td>Continuing a Parameter Statement</td>
<td>To continue a parameter statement onto the next card, key in a trailing comma.</td>
</tr>
<tr>
<td>Item</td>
<td>Rule</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Maximum Number of Reports Possible Per Execution</td>
<td>A total of 20 reports can be requested during each execution of CA IDMS Log Analyzer. This means you can choose 20 Program Detail Reports, for example, or any combination of reports and options available through CA IDMS Log Analyzer.</td>
</tr>
<tr>
<td>Entering Blanks on Parameter Statements</td>
<td>You can enter blanks (character spaces) to separate keywords to improve readability in a parameter statement without affecting the parameter. Do not enter blanks within a keyword or value field.</td>
</tr>
<tr>
<td>Positions of Keyword Phrases on Parameter Statements</td>
<td>All keyword phrases must be entered between positions 1 and 72 on each parameter card.</td>
</tr>
</tbody>
</table>

```
PROCESS
CONTINUE= [YES NO]  RHDCRUAL= [YES NO]  IDMSXXX= [YES NO]  ABND= [YES NO]

BILLING
START= mmddyyhhmm  STOP= mmddyyhhmm  INTERVAL= nnnnnn
FILE= [ONLY NO YES]
RUNAME= [OPER ID TERM ID TRANS ID ACCOUNT]
RUTYPE= [BATCH ONLINE SYSTEM]

PROGRAM
START= mmddyyhhmm  STOP= mmddyyhhmm  INTERVAL= nnnnnn
LEVEL= [DETAIL SUMMARY SYSTEM]
NAME= name

REPORT =

HI-SUM
START= mmddyyhhmm  STOP= mmddyyhhmm  INTERVAL= nnnnnn
LEVEL= [DETAIL SUMMARY SYSTEM]
NAME= name

HI-BPU
START= mmddyyhhmm  STOP= mmddyyhhmm

RANK
START= mmddyyhhmm  STOP= mmddyyhhmm  INTERVAL= nnnnnn
RUTYPE= [BATCH ONLINE SYSTEM]
RANKWHAT= value  RANKWHATJ = [MEAN MEDIAN ABSOLUTE]
RANKHOW= [HIGHEST LOWEST LT LE GT value GE]
RANK# = nn

COMMENTS Option
```
Billing Reports/Billing Record File Parameters

Contents

- REPORT Syntax (see page 320)
- START Syntax (see page 320)
- STOP Syntax (see page 321)
- INTERVAL Syntax (see page 321)
- LEVEL Syntax and Options (see page 321)
- FILE Syntax and Options (see page 322)
- RUTYPE Syntax and Options (see page 322)
- RUNAME Syntax and Options (see page 322)
- NAME Syntax and Options (see page 323)
  - How RUNAME, RUTYPE, and NAME Parameters Interrelate (see page 324)

The Billing Reports are available at four levels: details, summary, system, and grand summary. Grand summary, the fourth report level, is produced automatically when you select multiple time intervals. To generate these reports or the Billing Record File, use the parameter syntax listed here.

The parameters are:

```
REPORT=BILLING [,,START=mmddyyhhmm] [,,STOP=mmddyyhhmm]

[,,INTERVAL=mmmm]
[,,LEVEL={SUMMARY
       DETAIL
       SYSTEM}]
[,,FILE= {ONLY
        NO
        YES}]
[,,RUTYPE= {ONLINE
          BATCH
          SYSTEM}]
[,,RUNAME= {OPER-ID
            TERM-ID
            TRANS-ID
            ACCOUNT}]
[,,NAME=name or * ]
```

REPORT Syntax

REPORT=BILLING

Indicates that CA IDMS Log Analyzer is to process a request for a BILLING Report.

START Syntax

START=mmddyyhhmm

Specifies the starting date and time for a single report request.
Default: The default is the earliest Log date/time found in the input file.

Rules:

1. Start time must be specified in mmddyyhhmm format, where mmddyy represents the Gregorian date (month/day/year) and hhmm is the time (hour/minute) using the 24-hour clock.

2. All zeros must be entered as placeholders. For example, 0102032301 would be entered for 11:01 p.m. on January 2, 2003.

**STOP Syntax**

\[ \text{STOP}=\text{mmddyyhhmm} \]

Specifies the end of a selected time period for a single report request.

Default: The default for this keyword is the latest Log date/time found in the input file.

Rules:

1. Stop time must be specified in mmddyyhhmm format, where mmddyy represents the Gregorian date (month/day/year) and hhmm is the time (hour/minute) using the 24-hour clock.

2. All zeros must be entered as placeholders. For example, 0102032301 would be entered for 11:01 p.m. on January 2, 2003.

**INTERVAL Syntax**

\[ \text{INTERVAL}=\text{mmmm} \]

Specifies whether the time period you selected using the START and STOP parameters is to be divided into intervals and also specifies the length of those intervals.

Default: Zero. Therefore, the time period will be reported as a single interval.

Rules:

1. \text{INTERVAL} must be specified in the mmmmm (minutes) format. (It is not necessary to include leading zeros.)

2. If you specify multiple intervals, a Grand Summary Report will be produced automatically.

**LEVEL Syntax and Options**

\[ \text{LEVEL}=(\text{SUMMARY} \quad \text{DETAIL} \quad \text{SYSTEM}) \]

Use this parameter to specify the level of reporting that you want printed.

- \text{DETAIL}--indicates that you want CA IDMS Log Analyzer to print the Billing Details Report. (A Billing Summary Report and a Billing System Summary Report also will be produced for each time interval. A Grand Summary is produced only when you request multiple intervals.)
SUMMARY--indicates that you want a Billing Summary Report by program name. (A system summary will be produced for each time interval specified by the INTERVAL parameter. A Grand Summary is produced automatically whenever multiple intervals are reported.)

SYSTEM--indicates that you want only system summaries to be produced for each time interval. A Grand Summary is produced if multiple intervals are reported.

Default: SYSTEM

FILE Syntax and Options

FILE= {ONLY
NO
YES}

Use the FILE parameter to control the creation of a Billing Record File and a Billing Report.

ONLY--indicates that you want CA IDMS Log Analyzer to create only the Billing Record File but not the Billing Report(s).

NO--indicates that you want a Billing Report(s) but not the Billing Record File.

YES--indicates that you want CA IDMS Log Analyzer to create the Billing Record File along with a Billing Report(s).

Default: YES

RUTYPE Syntax and Options

RUTYPE= {ONLINE
BATCH
SYSTEM}

When each run-unit was originally executed by CA IDMS, it was executed as either a batch or online task. Using this parameter, you can specify whether the Billing Report will select only batch or only online run-units, or you can use the SYSTEM parameter to indicate that all system tasks (both batch and online) are to be reported.

Use this parameter to specify which type of run-units are to be selected for reporting.

BATCH--indicates that only batch run-units are to be reported on the Billing Report.

ONLINE--indicates that only ONLINE run-units are to be reported on the Billing Report.

SYSTEM--indicates that both batch and online run-units are to be reported on the Billing Report.

Default: SYSTEM

RUNAME Syntax and Options

RUNAME= {OPER-ID
TERM-ID
TRANS-ID
ACCOUNT}
An individual run-unit or CA ADS dialog may be identified in a number of ways. The way CA IDMS Log Analyzer identifies a run-unit is determined by two things:

- Whether this run-unit originates from an online transaction or a batch transaction.
- Whether this is a request for a Billing Report or a Program Report.

Billing Report parameters allow you to specify which field will be chosen as the identifying element of the run-unit (the run-unit name or RUNAME). Only the Billing Reports offer you this option. Online run-units can be identified by operator, terminal, or transaction of origin. Batch run-units can only be identified by the batch job's accounting information (i.e., account number, etc.).

- OPER-ID--specifies that online run-units (or dialogs) are to be identified by operator ID.
- TERM-ID--specifies that online run-units (or dialogs) are to be identified by terminal ID.
- TRANS-ID--specifies that online run-units (or dialogs) are to be specified by transaction ID. TRANS-ID is the default for online run-units.
- ACCOUNT--specifies that batch run-units are to be identified by account (i.e., account number or account name). ACCOUNT is the default for all batch run-units.

Default: ACCOUNT is the default for all batch run-units. TRANS-ID is the default for all online run-units.

**NAME Syntax and Options**

NAME = name or *

This parameter lets you select only those run-unit records that have a specific (or generic) run-unit name. The field that is to contain this name is specified by the RUNAME parameter.

Use this parameter to specify the actual (or generic) operator ID, terminal ID, transaction ID, or account information that a run-unit must have in order for that run-unit to be selected for analysis on the Billing Report.

Default: *. See note below for more information.

⚠️ **Note:** CA IDMS Log Analyzer will perform generic processing. For example, if you key in an asterisk (*) before a name field (i.e., NAME = *ABC), all run-units whose name field (as specified by the RUNAME parameter) begins with ABC will be included in the report.

The overhead records for CA ADS can be accessed by specifying RUNAME=TRANS-ID, NAME=$ADS@@OH.

Rules:

1. A maximum of eight characters can be entered for OPER-ID, TERM-ID, or TRANS-ID.
2. When BATCH is selected as RUTYPE and ACCOUNT is chosen as RUNAME, up to eleven (11) characters can be supplied for this value field.

How RUNAME, RUTYPE, and NAME Parameters Interrelate

RUNAME is different for online run-units than it is for batch run-units. This means you should use NAME with RUTYPE = SYSTEM only after carefully considering whether the RUNAME fields of batch and online run-units will both contain the same value. For example, if you specify NAME = ABC, will batch run-units have an ACCOUNT field of ABC and online run-units have a TRANS-ID (or OPER-ID or TERM-ID) field of ABC?

Program Report Parameters

Contents
- REPORT Syntax (see page 324)
- START Syntax (see page 324)
- STOP Syntax (see page 325)
- INTERVAL Syntax (see page 325)
- LEVEL Syntax and Options (see page 325)
- NAME Syntax and Options (see page 326)

Four Program Reports are available at four different levels: detail, summary, and system. The fourth level, grand summary, is generated automatically if you select multiple intervals. To get these reports, use the parameter syntax listed on this page.

REPORT=PROGRAM [, START=mmddyyhhmm] [, STOP=mmddyyhhmm] [, INTERVAL=mmmm] [, LEVEL={SUMMARY, DETAIL, SYSTEM}] [, NAME=name or * ]

REPORT Syntax

REPORT= PROGRAM

Specifies that CA IDMS Log Analyzer is to create and print a PROGRAM Report.

START Syntax

START = mmddyyhhmm

specifies the starting date and time for a single report request.

Default: The earliest Log date/time found in the input file.

Rules:

1. Start time must be specified in mmddyyhhmm format, where mmddyy represents the Gregorian date (month/day/year) and hhmm is the time (hour/minute) using the 24-hour clock.
2. All zeros must be entered as placeholders. For example, 0102032301 would be entered for 11:01 p.m. on January 2, 2003.

**STOP Syntax**

```
STOP = mmddyyhhmm
```

Specifies the end of a selected time period for a single report request.

Default: The default for this keyword is the latest Log date/time found in the input file.

Rules:

1. Stop time must be specified in mmddyyhhmm format, where mmddyy represents the Gregorian date (month/day/year) and hhmm is the time (hour/minute) using the 24-hour clock.

2. All zeros must be entered as placeholders. For example, 0102032301 would be entered for 11:01 p.m. on January 2, 2003.

**INTERVAL Syntax**

```
INTERVAL = mmmmm
```

Specifies whether the time period you selected using the START and STOP parameters is to be divided into intervals and also specifies the length of those intervals.

Default: Zero. Therefore, the time period will be reported as a single interval.

Rules:

1. INTERVAL must be specified in the mmmmm (minutes) format. (It is not necessary to include leading zeros.)

2. If you specify multiple intervals, a Grand Summary Report will be produced automatically.

**LEVEL Syntax and Options**

```
LEVEL={SUMMARY DETAIL SYSTEM}
```

Use this parameter to specify the level of report that you want printed.

- **DETAIL**--indicates that you want CA IDMS Log Analyzer to print the Program Details Report (A Program Summary Report and a Program System Summary Report also will be produced for each time interval. A Grand Summary is produced only when you request multiple intervals.)

- **SUMMARY**--indicates that you want a Program Summary Report for all run units selected. In addition, a system summary will be produced for each time interval specified by the INTERVAL parameter. A grand summary is produced if multiple intervals are reported. SYSTEM is the default value.
CA IDMS - 19.0

- SYSTEM--indicates that only Program System Summary Reports will be produced for each time interval. A grand summary is produced only when multiple intervals are requested.

Default: SYSTEM

NAME Syntax and Options

NAME = name or *

Allows you to supply the actual (or generic) name of the program(s) or dialog(s) to be reported on by CA IDMS Log Analyzer.

For CA ADS dialogs, NAME identifies the name of the dialog. Overhead records for CA ADS are reported under the name ADS@@OH.

For tasks executed within CA IDMS/DC, NAME is not the name of the program. Instead, it is the name of the CA IDMS/DC task.

Default: *. See note below for further information.

⚠️ Note: CA IDMS Log Analyzer will perform generic processing. For example, if you key in an asterisk (*) before a name field (i. e., NAME = *ABC), all run-units whose program name begins with ABC will be included in the report.

Management Highlights Summary Report Parameters

Contents

- REPORT Syntax (see page 326)
- START Syntax (see page 327)
- STOP Syntax (see page 327)
- INTERVAL Syntax (see page 327)
- LEVEL Syntax (see page 328)

The Management Highlights Summary Reports are available at three levels of detail: program, system, and grand. To generate these reports, use the parameter syntax listed on this page.

REPORT=HI-SUM [, START=mmddyyhhmm] [, STOP=mmddyyhhmm]

[ , INTERVAL=mmmmmm]

[ , LEVEL=(PROGRAM
SYSTEM
GRAND)]

REPORT Syntax

REPORT = HI-SUM
Indicates that CA IDMS Log Analyzer is to create and print a Management Highlights Summary Report.

**START Syntax**

```plaintext
START = mmddyyhhmm
```

Specifies the starting date and time for a single report request.

Default: The earliest Log date/time found in the input file.

Rules:

1. Start time must be specified in mmddyyhhmm format, where mmddyy represents the Gregorian date (month/day/year) and hhmm is the time (hour/minute) using the 24-hour clock.

2. All zeros must be entered as placeholders. For example, 0102032301 would be entered for 11:01 p.m. on January 2, 2003.

**STOP Syntax**

```plaintext
STOP = mmddyyhhmm
```

Specifies the end of a selected time period for a single report request.

Default: The default is the latest Log date/time found in the input file.

Rules:

1. Stop time must be specified in mmddyyhhmm format, where mmddyy represents the Gregorian date (month/day/year) and hhmm is the time (hour/minute) using the 24-hour clock.

2. All zeros must be entered as placeholders. For example, 0102032301 would be entered for 11:01 p.m. on January 2, 2003.

**INTERVAL Syntax**

```plaintext
INTERVAL = mmmmm
```

specifies whether the time period you selected using the START and STOP parameters is to be divided into intervals and also specifies the length of those intervals.

Default: Zero. Therefore, the time period will be reported as a single interval.

Rules:

1. INTERVAL must be specified in the mmmmm (minutes) format. (It is not necessary to include leading zeros.)

2. If you specify multiple intervals, a Grand Summary Report will be produced automatically.
LEVEL Syntax

LEVEL={(PROGRAM
       SYSTEM
       GRAND)}

Use this parameter to specify the level of report you want printed.

- PROGRAM--indicates accumulation for each program or dialog in a time interval. A Management Highlights System Summary Report also will be produced for each time interval. (A Management Highlights Grand Summary Report is produced only when you request multiple intervals.)

- SYSTEM--indicates accumulation for all programs and dialogs within a time interval. A Management Highlights Grand Summary Report is produced if multiple intervals are reported.

- GRAND--indicates accumulation of all programs and dialogs within all time intervals. GRAND is identical to SYSTEM if the INTERVAL parameter is not supplied.

Default: GRAND

Buffer Pool Utilization Report Parameters

Contents
- REPORT Syntax (see page 328)
- START Syntax (see page 328)
- STOP Syntax (see page 329)

To generate the Management Highlights Buffer Pool Utilization Report, use the parameter syntax listed on this page.

REPORT= HI-BPU [,START =mmddyyhhmm] [,STOP =mmddyyhhmm]

REPORT Syntax

REPORT= HI-BPU

Specifies that CA IDMS Log Analyzer is to create and print a Management Highlights Buffer Pool Utilization Report.

⚠️ Note: Buffer Pool Utilization Reports are generated on 24-hour intervals only. The built-in interval begins at midnight and continues until midnight of the next day. CA IDMS Log Analyzer will override any INTERVAL you attempt to supply.

START Syntax

START = mmddyyhhmm

Specifies the starting date and time for a single report request.
Default: The default is the earliest Log date/time found in the input file.

Rules:

1. Start time must be specified in mmddyyhhmm format, where mmddyy represents the Gregorian date (month/day/year) and hhmm is the time (hour/minute) using the 24-hour clock.

2. All zeros must be entered as placeholders. For example, 0102032301 would be entered for 11:01 p.m. on January 2, 2003.

STOP Syntax

STOP = mmddyyhhmm

Specifies the end of a selected time period for a single report request.

Default: The default for this keyword is the latest Log date/time found in the input file.

Rules:

1. Stop time must be specified in mmddyyhhmm format, where mmddyy represents the Gregorian date (month/day/year) and hhmm is the time (hour/minute) using the 24-hour clock.

2. All zeros must be entered as placeholders. For example, 0102032301 would be entered for 11:01 p.m. on January 2, 2003.

Management Ranking Report Parameters

Contents

- REPORT Syntax (see page 330)
- START Syntax (see page 330)
- STOP Syntax (see page 330)
- INTERVAL Syntax (see page 330)
- RUTYPE Syntax and Options (see page 331)
- RANKWHAT Syntax and Options (see page 331)
- RANKVALUE Syntax and Options (see page 332)
- RANKWHOW Syntax and Options (see page 332)
- RANK Syntax (see page 333)

The REPORT parameter syntax for the Management Ranking Report is listed below.

REPORT=RANK [,START=mmddyyhhmm] [,STOP=mmddyyhhmm] [,INTERVAL=mmmmmm] [,RUTYPE={ONLINE BATCH SYSTEM}] [,RANKWHAT={item DETAIL}]}
CA IDMS - 19.0

[ ,RANKVALU= {MEAN
MEDIAN
ABSOLUTE}]
[ ,RANKHOW= {HIGHEST
LOWEST
LT: value
LE: value
GT: value
GE: value}]
[ ,RANK# =nn]

REPORT Syntax
   REPORT =RANK

Indicates that CA IDMS Log Analyzer is to create and print a Management Ranking Report.

START Syntax
   START =mmddyyhhmm

Specifies the starting date and time for a single report request.

Default: The earliest Log date/time found in the input file.

Rules:
   1. Start time must be specified in mmddyyhhmm format, where mmddyy represents the
      Gregorian date (month/day/year) and hhmm is the time (hour/minute) using the 24-hour
      clock.
   2. All zeros must be entered as placeholders. For example, 0102032301 would be entered for 11:
      01 p.m. on January 2, 2003.

STOP Syntax
   STOP = mmddyyhhmm

Specifies the end of a selected time period for a single report request.

Default: The default for this keyword is the latest Log date/time found in the input file.

Rules:
   1. Stop time must be specified in mmddyyhhmm format, where mmddyy represents the
      Gregorian date (month/day/year) and hhmm is the time (hour/minute) using the 24-hour
      clock.
   2. All zeros must be entered as placeholders. For example, 0102032301 would be entered for 11:
      01 p.m. on January 2, 2003.

INTERVAL Syntax
   INTERVAL = mmmm

16-Jan-2018  330/898
Specifies whether the time period you selected using the START and STOP parameters is to be divided into intervals and also specifies how long those intervals should be.

Default: An interval of zero means that the time period will be reported as a single interval. This is the default.

Rule: INTERVAL must be specified in the mmmmm (minutes) format. (It is not necessary to include leading zeros.)

**RUTYPE Syntax and Options**

```
RUTYPE= {ONLINE
          BATCH
          SYSTEM}
```

When each run-unit was originally executed by CA IDMS, it was executed as either a batch or online task. Using this parameter, you can specify whether the Management Ranking Report will select only batch or only online run-units, or you can use the SYSTEM parameter to indicate that all system tasks (both batch and online) are to be reported.

Use this parameter to specify which type of run-units are to be selected for ranking.

- **ONLINE**--indicates that only online programs or dialogs are to be ranked.
- **BATCH**--indicates that only batch programs are to be ranked.
- **SYSTEM**--indicates that both batch and online programs and dialogs will be ranked.

Default: SYSTEM

**RANKWHAT Syntax and Options**

```
RANKWHAT =item
```

Specifies which attribute is to be ranked in the report.

Default: None. You must select one of the following items:

- **#RU**--number of run units (RANKVALU must be ABSOLUTE)
- **PG-READ**--pages read
- **PG-WRITTEN**--pages written
- **PG-IO**--pages written + pages read (total I/Os)
- **PG-RATIO**--pages requested/pages read
- **CALC-RATIO**--calc records overflow/calc records on home page
- **VIA-RATIO**--via records overflow/via records on owner page
- **CPU-TIME**--user-mode-time + system-mode-time
CA IDMS - 19.0

RANKVALUE Syntax and Options

\[ \text{RANKVALUE} = \{\text{MEAN} \]
\[ \quad \text{MEDIAN} \]
\[ \quad \text{ABSOLUTE}\} \]

Run-unit information may be ranked according to the actual value of the attribute selected. As an alternative, you may specify that the MEAN or MEDIAN value for all program occurrences of the attribute is to be the basis of the ranking.

- **MEAN**—indicates that programs are to be ranked by the average value of the attribute for all run-units of the ranked program.
- **MEDIAN**—indicates that programs are to be ranked by the median value of the attribute for all run-units of the ranked program.
- **ABSOLUTE**—indicates that run-units are to be ranked by the value of the attribute from each run-unit of the ranked program.

Default: ABSOLUTE

Rule: ABSOLUTE must be specified when RANKWHAT=#RU is specified.

RANKHOW Syntax and Options

\[ \text{RANKHOW} = \{\text{HIGHEST} \]
\[ \quad \text{LOWEST} \]
\[ \quad \text{LT} : \text{value} \]
\[ \quad \text{LE} : \text{value} \]
\[ \quad \text{GT} : \text{value} \]
\[ \quad \text{GE} : \text{value}\} \]

Use this keyword to specify how the attribute you selected is to be ranked. There are six methods to choose from.

- **HIGHEST**—the attribute will be ranked from its highest value in a descending order.
- **LOWEST**—the attribute will be ranked from its lowest value in an ascending order.
- **LT: value**—the attribute will be ranked from a value less than the specified value in a descending order.
- **LE: value**—the attribute will be ranked from a value less than or equal to the specified value in a descending order.
- **GT: value**—the attribute will be ranked from a value greater than the specified value in an ascending order.
- **GE: value**—the attribute will be ranked from a value greater than or equal to the specified value in an ascending order.

Default: None. You must select one of the six available ranking methods.

Rules:
1. The specified values for LT, LE, GT, and GE must include two decimal positions when ranking medians or ratios (e. g., median of five pages read is 500; a ratio of 3.14 is 314).

2. Value (for LT, LE, GT, and GE) must be nine digits or less. Leading zeros are not required.

3. HIGHEST or LOWEST must be specified when RANKWHAT=#RU is specified.

4. HIGHEST or LOWEST must be specified when either RANKVALU=MEAN or RANKVALU=MEDIAN is specified.

**RANK Syntax**

```
RANK# = nn
```

After all of the run-unit records have been selected and ranked, you may also request how many you want to see on the Management Ranking Report. For example, if you only want to see the first 10 when ranked according to your RANKHOW parameter, specify RANK# = 10.

Use this parameter to specify the number of items to be reported on the RANKING report.

Default: 20

Rule: The maximum number of items that can be reported is 50. (It is not necessary to include leading zeros.)

---

**Customizing the Billing Reports**

This section contains sectionlines for customizing the billing data produced by CA IDMS Log Analyzer. Billing data for batch jobs or CICS transactions is initially created by GSISVCX, the CA IDMS Tools module that operates as an exit to the CA IDMS SVC (billing data for CA IDMS/DC tasks are created by CA IDMS). Billing data is reported by USLRPT5, the Billing Report module, and formatted as a file by USLBILX, the Billing Formatter Exit module. Source code is provided for these modules so that you can easily change them. Tailoring the format of the billing data can provide you with data that conforms to the processing requirements of your billing system. The billing data contained in the Billing Record File can then be merged and processed with other cost and billing information in your system.

- Customizing the GSISVCX Module (see page 334)
- Customizing the USLBILX Module (see page 335)
- Customizing the USLRPTS Module (see page 335)

Two portions of the CA IDMS Log Analyzer system are supplied to you in executable object code format and also in source code format. With object and source code available, you can run CA IDMS Log Analyzer just as it is supplied to you, or you can customize any aspect of the billing functions CA IDMS Log Analyzer performs.

Source code is supplied for:

- **GSISVCX**--the CA IDMS Tools version of the CA IDMS SVC exit module.
Customizing the GSISVCX Module

GSISVCX, the CA IDMS Tools version of the assembler language module IDMSSVCX is designed to create a 40-position extension to the CA IDMS External Request Element control block (ERE). The type of data that is placed into these 40 positions by the module depends, in part, on whether the run-unit being processed is identified by CA IDMS as BATCH or CICS.

For any type of run-unit--the JOBNAME, the run-unit start date and time, and the step start time are moved into the ERE by GSISVCX.

For BATCH run-units--up to 16 bytes of information contained in the account field of the job card are moved into the ERE by GSISVCX.

For CICS run-units--the transaction ID, terminal ID, and operator ID are moved into the ERE by GSISVCX. You will need to customize this module if your installation already uses a version of IDMSSVCX and that function must be retained for continued use: if your account number is not in the first field of the z/OS JOB ACCT parameter; if your installation uses a TP monitor other than CICS; or if the data moved into the ERE is not sufficient for your billing system requirements.

To customize GSISVCX, your systems programmer must make the desired changes to the source code. See the CA IDMS Installation and Maintenance Section.

When altering the source code for GSISVCX, follow these sectionlines:

- The ERE may be defined as any length between 40 and 32767 but only the first 40 positions will be written by CA IDMS to the Task Statistics Record.

- All CA IDMS Log Analyzer Billing Report functions depend upon the data in the ERE. This is especially true of the ERE fields containing the ONLINE/BATCH designation, and the ACCOUNT/TRANS-ID, TERM-ID, OPER-ID data. If the position, size, or format of these data fields is altered, it will have serious impact on your ability to select log records for inclusion in the Billing Reports or Billing File. Specifically, if the ONLINE/BATCH indicator is moved or altered, you will no longer be able to specify RUTYPE=ONLINE or RUTYPE=BATCH on any request for Billing Reports or Billing Files. See Parameters. If ACCOUNT/TRANS-ID, TERM-ID, or OPER-ID are moved or altered, you will no longer be able to use the NAME parameter for any Billing request. Finally, if any of these fields are moved or altered, the USLBILX and USLRPT5 modules will have to be modified to accept the revised format of the ERE. See External Request Element Extension for the ERE descriptions.

- After the GSISVCX source code is modified, the resulting load module must be identified to the CA IDMS SVC. See the CA IDMS Installation and Maintenance Section for detailed information.
Customizing the USLBILX Module

The COBOL language Billing File formatter module, USLBILX, is designed to access the information placed into the ERE. In addition, this module is designed to combine ERE extension information with other data from the Task Statistics Archive Log Record to create a Billing Record in a predefined format. See Billing Record file for a description of the Billing Record File.

When a CA IDMS/DC task is processed, there will not be any ERE extension information.

You must customize this module if the content of the ERE extension is changed when GSISVCX is changed; if the content or format of the Billing Record is not compatible with the billing system in your environment; or if data from additional sources must be merged into the Billing Record. See USLBILX and USLRPT5 Source Code for information on printing the source module.

After making the desired changes to the source code, recompile USLRPT5 specifying NODYNAM, NORES and relink the CA IDMS Log Analyzer module USLRPT5. Target or Distribution s ( z/OS), TOOLJCL library member USLRLNK5.S(Z/VSE), the USLRLNK5 EXEC (Z/VM), contains the linkage editor control statements for USRLNK5.

Customizing the USLRPT5 Module

The COBOL language Billing Reporter module, USLRPT5, is designed to access the information placed into the ERE extension by GSISVCX. For CA IDMS/DC tasks, the ERE extension portion of the log contains no data, but all necessary information is contained elsewhere in the log record. Using this information, along with other data from the Task Statistics Record, USLRPT5 produces the CA IDMS Log Analyzer Billing Report.

Extracted data from the log file is presented, one record at a time, to USLRPT5 in a predefined sequence depending upon the value of RUNAME on the Billing request. Because the field defined by RUNAME resides in the ERE extension, the sequence may be adversely affected by alterations to the ERE extension.

You must customize this module if the content of the ERE extension is altered when you changed GSISVCX or if you need to change the Billing Report format to conform with standards in your environment. See USLBILX and USLRPT5 Source Code for information on printing the source module.

After making the desired changes to the source code, recompile USLRPT5 specifying NODYNAM, NORES and relink the CA IDMS Log Analyzer module USLRPT5. Target or Distribution s ( z/OS), TOOLJCL library member USLRLNK5.S(Z/VSE), the USLRLNK5 EXEC (Z/VM), contains the linkage editor control statements for USRLNK5.

CA IDMS Log Analyzer Operations

This section lists and explains the JCL necessary to successfully execute CA IDMS Log Analyzer in an z/OS, Z/VSE, or Z/VM environment.

- z/OS Environment (see page 336)
- Z/VSE Environment (see page 337)
z/OS Environment

Contents
- z/OS JCL (see page 336)
  - z/OS Operation Notes (see page 336)

The z/OS JCL necessary to execute CA IDMS Log Analyzer is contained in Target or Distribution source library member USLEXEC. A sample of the supplied JCL is listed, followed by notes.

z/OS JCL

```plaintext
//USMAIN JOB(job accounting information)
//LOGAPROC PROC OUTCLASS=A, (1)
//    LOGALIB='YOUR.LOGA.LOADLIB', (2)
//    PROG=USLMAIN, (3)
//    LOGDSN='IDMS.LGARCHIV', (4)
//    LOGUNIT=, (5)
//    LOGVOL=, (5)
//    SORTCYL='(5,5)', (6)
//    EXTCYL='(5,5)', (6)
//    EXTDISP=(NEW,DELETE,DELETE)', (8)
//    BILLDSN='YOUR.LOGA.BILLFILE', (9)
//    BILLCYL='(5,5)', (6)
//    BILLBLK=, (10)
//    BILDISP=CATLG (11)
//LOGA EXEC PGM=&PROG.,
//STEPLIB DD DSN=&LOGALIB.,DISP=SHR
//SORTMSG DD SYSSOUT=&OUTCLAS.
//SYSOUT DD SYSSOUT=&OUTCLAS.
//AUDIT DD SYSSOUT=&OUTCLAS.
//REPORTS DD SYSSOUT=&OUTCLAS.
//SYSUDUMP DD SYSSOUT=&OUTCLAS.
//LOGFILE DD DSN=6LOGDSN.,
//    DISP=OLD&LOGUNIT.&LOGVOL.
//SORTWK01 DD UNIT=DISK,SPACE=(CYL,&SORTCYL).
//SORTWK02 DD UNIT=DISK,SPACE=(CYL,&SORTCYL).
//SORTWK03 DD UNIT=DISK,SPACE=(CYL,&SORTCYL).
//SORTWK04 DD UNIT=DISK,SPACE=(CYL,&SORTCYL).
//SORTWK05 DD UNIT=DISK,SPACE=(CYL,&SORTCYL).
//EXTRACT DD DSN=6EXTRACT.
//    UNIT=DISK,SPACE=(CYL,&EXTCYL),
//    DISP=EXTDISP.&EXTBLK.
//BILLFILE DD DSN=6BILLFILE,
//    UNIT=DISK,SPACE=(CYL,&BILLCYL),
//    DISP=(NEW,&BILDISP.,DELETE)&BILLBLK.
//SYSIPT DD DDNAME=SYSIN PEND
//LOGA EXEC LOGAPROC,
//    * SPECIFY SORTLIB DD DSN=.... (if necessary) (12)
//    * ENTER INPUT PARAMETERS HERE
//    PROCESS, CONT=Y, IDMSXXXX=YES, RHDCRUAL=N0
//    REPORT=HI-BPU
//    REPORT=HI-SUM
//    REPORT=BILLING,LEVEL=DETAIL
//    REPORT=PROGRAM,LEVEL=SUMMARY
//    REPORT=RANK, RANKWHAT=PG-10, RANKHOW = HIGHEST
```

z/OS Operation Notes
1. Specify OUTCLAS to assign print output to other than CLASS=A.

2. Specify LOGALIB to be the same as LOADLIB in STEP1 of the installation procedure.

3. Specify PROG if you have changed the name of the CA IDMS Log Analyzer module as created in the first step of the installation process.

4. Specify LOGDSN to name the input Archive Log File. This file must be the SYS002 file from the CA IDMS ARCHIVE LOG utility or the SYS020 file from CA Culprit Statistics Report 99.

5. Specify LOGUNIT and LOGVOL if the archive log file to be processed by CA IDMS Log Analyzer is not a cataloged dataset. Observe the required format for these parameters.

6. Specify SORTCYL, EXTCYL or BILLCYL if CYL (5,5) is not an appropriate space allocation for that file.
   You can estimate file size if you know how many run-units are likely to match the selection criteria of your REPORT parameter statements. For each run-unit, 19 records are created for each HI-SUM request, and one record is created for every other type of report. These records are sorted and then written to the EXTRACT file. Also, each record for a BILLING request where FILE = YES or FILE = ONLY is written to the BILLFILE.

7. Specify EXTBLOCK to create an EXTRACT file BLKSIZE suited to the type of storage device used in your environment.
   Extract records for Billing or Program reports are 516 bytes long. All other records on the variable length extract file are 164 bytes long. EXTBLOCK may specify any BLKSIZE that is at least 4 bytes larger than the largest record being created. The default EXTBLOCK is 6144.

8. Specify EXTDISP to choose a final disposition of the Extract File. The Extract File is used within CA IDMS Log Analyzer. The default disposition is (NEW, DELETE, DELETE).

9. Specify BILLDSN to name the Billing Record File that CA IDMS Log Analyzer is to create. This may be specified as BILLDSN=NULLFILE if no Billing record File is to be created.

10. Specify BILLBLK to create a BILLING RECORD file BLKSIZE suited to the type of storage device used at your installation. BILLBLK must specify a BLKSIZE that is a multiple of 100, the BILLFILE file record length. The default BILLBLK is 6100.

11. Specify BILDDISP to choose a disposition for the Billing Record File. The default disposition is CATLG. Specify BILDDISP=PASS or BILDDISP=DELETE.

12. If it is required by your installation, insert //SORTLIB DD DSN=sort-library-name, DISP=SHR prior to the parameter statements. This statement names the library containing your SORT utility.

Z/VSE Environment

Contents
- Z/VSE JCL (see page 338)
  - Z/VSE Operation Notes (see page 338)
  - Z/VSE File Processing Alternate Method (see page 339)
A sample of a Z/VSE environment JCL to execute CA IDMS Log Analyzer is contained in TOOLJCL library member USLEXEC.S. This sample must be modified to reflect your hardware. A sample of the supplied JCL is listed below followed by the notes.

**Z/VSE JCL**

```plaintext
// OPTION PARTDUMP
// ASSGN SYS005,SYSIPT PARAMETER INPUT
// ASSGN SYS006,SYSIPT AUDIT REPORT
// ASSGN SYS010,CUU SELECTED REPORTS
// DLBL LOGFILE,'IDMS16.LGARCHIV',999,SD
// EXTENT SYS010,DISK,1,0,417200,10000 (3,4,6)

* ASSGN SYS009,CUU BILLING FILE
// DLBL BILLFIL,'BILLING.LOGA',0,SD
// EXTENT SYS009,DISK,1,0,348000,01999 X (2)

* ASSGN SYS008,CUU EXTRACT FILE
// DLBL EXTRACT,'LOGA.EXTRACT',0,SD
// EXTENT SYS008,DISK,1,0,427200,05270 X (3,6)

* ASSGN SYS001,CUU SORT WORK #1
// DLBL SORTWK1,'SORT.WORK.1',0,SD
// EXTENT SYS001,DISK,1,0,396000,07000 X (2)

* ASSGN SYS002,CUU SORT WORK #2
// DLBL SORTWK2,'SORT.WORK.2',0,SD
// EXTENT SYS002,DISK,1,0,403000,07000 X (2)

* ASSGN SYS003,CUU SORT WORK #3
// DLBL SORTWK3,'SORT.WORK.3',0,SD
// EXTENT SYS003,DISK,1,0,410000,07000 X (2)

* PRIVATE CORE IMAGE LIBRARY(S)
// DLBL CILIB1,'YOUR-LOGA-CORELIB' LOG ANALYZER LIBRARY
// EXTENT,VOL=VOLSER
// DLBL CILIB2,'YOUR-IDMS-CORELIB' IDMS LIBRARY
// EXTENT,VOL=VOLSER
// LIBDEF CL,SEARCH=(CILIB1,CILIB2),TEMP

* EXEC USLMAIN,SIZE=(AUTO,48K) (5)
*ENTER INPUT PARAMETERS
PROCESS , CONT=Y , IDMSXXXX=YES , RHDCRUAL=NO
REPORT=HI-BPU
REPORT=HI-SUM
REPORT=BILLING,LEVEL=DETAIL
REPORT=PROGRAM,LEVEL=SUMMARY
REPORT=RANK, RANKWHAT=PG-I0, RANKHOW = HIGHEST

/*
```

**Z/VSE Operation Notes**

1. Modify the unit addresses to refer to your installation’s unit(s).

2. Specify extents and volume serial number(s) appropriate to your volume(s).
   You can estimate size if you know how many run-units is likely to match the selection criteria of your REPORT parameter statements. For each run-unit, 19 records are created for each HI-SUM request, and one record is created for each other type of request. These records are sorted and then written to the EXTRACT file. Also, each record for a BILLING request where FILE = YES or FILE = ONLY is also written to the BILLFILE.

3. Block sizes are assigned for all files by the GSSGNCB module.
4. This file must be the SYS002 file from the CA IDMS ARCHIVE LOG utility or the SYS020 file from CA Culprit Statistics Report 99.

5. Ensure that a 1024K partition is available for this job.

6. Even if you use a storage management tool such as CA-DYNAM, an ASSGN statement is required by CA IDMS Log Analyzer for every file except SORTWKnn. This assignment is necessary because CA IDMS Log Analyzer has its own device-independent support which dynamically builds a DTF based on the device type indicated by the assignment of the logical unit. The logical unit required for each work file is provided in the table in 5-2. The device may be defined with DLBL or TLBL.

Z/VSE File Processing Alternate Method

Occasionally you will receive a message that a file is not VSAM. The message indicates that the dataset will be processed SAM instead of VSAM because CA IDMS Log Analyzer was not able to find the dataset in the VSAM catalog. The allocation will not affect processing results.

Z/VM Environment

Contents
- Z/VM EXEC (see page 339)
  - Key to Z/VM EXEC (see page 340)

A model Z/VM EXEC for executing CA IDMS Log Analyzer is shown below. Variables (boldface) are explained in the key following the EXEC.

Z/VM EXEC

```plaintext
/* * /
TRACE OFF; SIGNAL ON ERROR
CA_LOADLIB_FN = 'yourlib'
IDMS_LOADLIB_FN = 'idmslib'
SORTLIB_FN = 'sortlib'
LOG_ARCHIVE_FN = 'your.log.archive'
LOG_ARCHIVE_FT = 'filetype'
LOG_ARCHIVE_FM = '*'
/* */
/* Link and access the Minidisks containing the required librarie(s) */
/* and database file(s). */
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'GLOBAL LOADLIB ' CA_LOADLIB_FN IDMS_LOADLIB_FN
'GLOBAL TXTLIB ' SORTLIB_FN
/* */
/* Files needed for all runs. */
'FILEDEF SORTMSG PRINTER'
'FILEDEF SYSUDUMP PRINTER'
'FILEDEF SYSOUT PRINTER'
'FILEDEF AUDIT DISK LOGA AUDIT fm'
'FILEDEF REPORTS DISK LOGA REPORTS fm'
'FILEDEF EXTRACT DISK LOGA EXTRACT fm'
'FILEDEF EXTRACT DISK ',
LOG_ARCHIVE_FN LOG_ARCHIVE_FT LOG_ARCHIVE_FM
/* */
/* You must create a file 'USLEXEC SYSIPT A' containing the input */
/* parameter statements prior to executing this EXEC. */
```
CA IDMS - 19.0

/* This file must include a PROCESS statement and other statements */
/* for the reports and displays that you want generated. See CA IDMS */
/* Log Analyzer User Guide for further details. */

/* FILEDEF SYSIPT DISK USLEXEC SYSIPT A */

/* Insert FILEDEF statements for SORT work space as required by */
/* your SORT product. */

/* FILEDEF SORTWK01 DISK sort_fn sort_ft sort_fm4 ( XTENT 100 ' */

/* SIGNAL OFF ERROR */
SAY 'STARTING EXECUTION OF CA IDMS/LOG ANALYZER'
USLEXEC_RC = RC
'EXECOS OSRUN USLMAIN'
USLEXEC_RC = RC
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME LOGA LISTING'
'SIGNAL OFF ERROR
SAY 'NON-ZERO RETURN CODE ENCOUNTERED IN EXEC AT LINE' SIGL
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME LOGA LISTING'
'GLOBAL LOADLIB'
'GLOBAL TXTLIB'
'FILEDEF * CLEAR'
EXIT USLEXEC_RC

/* ERROR:
*/

Key to Z/VM EXEC

Z/VM JCL Sample Key:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>yourlib</td>
<td>The file name of the library into which you downloaded CA IDMS Log Analyzer.</td>
</tr>
<tr>
<td>idmslib</td>
<td>The file name of the load library containing your CA IDMS SUBSCHEMA and DMCL modules.</td>
</tr>
<tr>
<td>sortlib</td>
<td>The file name of the text library containing your sort modules.</td>
</tr>
<tr>
<td>your.log. archive</td>
<td>The file name of your log archive file. This file must be the SYS002 file from the CA IDMS utility RHDCPRLG or the SYS020 file from CA Culprit Statistics Report 99.</td>
</tr>
<tr>
<td>LOGA REPORTS fm</td>
<td>The file name, file type, and file mode of your REPORTS file.</td>
</tr>
<tr>
<td>LOGA EXTRACT fm</td>
<td>The file name, file type, and file mode of your EXTRACT file.</td>
</tr>
<tr>
<td>sort_fn sort_ft sort_fm4</td>
<td>The file name, file type, and file mode of your sort work files. The size of the sort work files can be adjusted depending on the size of the EXTRACT file. You can estimate extract file size if you know how many records are likely to match the selection criteria of your</td>
</tr>
</tbody>
</table>
Parameter Description

REPORT parameter statements. For each run-unit, 19 records are created for each HI-SUM request, and one record is created for each other type of report. These records are sorted and then written to the EXTRACT file.

\textbf{fm}

\textbf{Note:} Ensure that your virtual machine has been IPL'd with enough storage. Contact your systems programmer for information on increasing its size, if necessary. The Log must be archived using the CA IDMS utility with a file mode of x4, to indicate z/OS file-type simulation, and a DCB of: (RECFM U LRECL 4096.

<table>
<thead>
<tr>
<th>FILE/ NAME</th>
<th>LOGICAL UNIT</th>
<th>RECORD SIZE</th>
<th>BLOCK SIZE - RDR</th>
<th>PRT</th>
<th>DISK/ TAPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSIPT</td>
<td>SYS005</td>
<td>80</td>
<td>80</td>
<td>*</td>
<td>6080 ***</td>
</tr>
<tr>
<td>AUDIT</td>
<td>SYS006</td>
<td>133</td>
<td>*</td>
<td>133</td>
<td>6118</td>
</tr>
<tr>
<td>REPORTS</td>
<td>SYS007</td>
<td>133</td>
<td>*</td>
<td>133</td>
<td>6118</td>
</tr>
<tr>
<td>EXTRACT</td>
<td>SYS008</td>
<td>164-516 **</td>
<td>*</td>
<td>*</td>
<td>6144</td>
</tr>
<tr>
<td>BILLFILE</td>
<td>SYS009</td>
<td>100</td>
<td>*</td>
<td>*</td>
<td>6100</td>
</tr>
<tr>
<td>LOGFILE</td>
<td>SYS010</td>
<td>24-376 **</td>
<td>*</td>
<td>*</td>
<td>6144 ***</td>
</tr>
<tr>
<td>DISKLOG</td>
<td>SYS011</td>
<td>133</td>
<td>*</td>
<td>*</td>
<td>6118 ***</td>
</tr>
</tbody>
</table>

\textbf{Notes:}

* This file cannot be assigned to this device type.

** This record is variable length: minimum and maximum shown.

*** The size shown is the maximum that can be processed. Smaller block sizes that are an appropriate multiple of the record size can also be processed.
USLBILX and USLRPT5 Source Code

This section gives instructions for printing the CA source code for USLBILX and USLRPT5. The source code will allow you to tailor the CA IDMS Log Analyzer billing reports. The source code for USLBILX and USLRPT5 was cataloged into your source statement library during the installation procedure. The procedure for listing these modules differs for Z/OS and Z/VSE users. See the CA IDMS Installation and Maintenance for z/OS for further information.

Listing Modules for Z/OS

We suggest that Z/OS users use the utility IEBGENER to print or punch the source code for the desired member.

Listing Modules for Z/VSE

We suggest that Z/VSE users use the LIBR function LIST or PUNCH to display and/or punch the desired modules. USLBILX and USLRPT5 are in the COBOL statement library.

External Request Element Extension

This appendix provides a description of the External Request Element (ERE) extension.

Altering the ERE description is necessary if you want to tailor the CA IDMS Log Analyzer Billing Reports or the Billing Record file.

To change the ERE extension you must alter GSISVCX, USLBILX, and USLRPT5.

*------------------------------------------------------------------------------*
* ERE LAYOUT (AS CREATED BY GSISVCX FOR USE IN CA IDMS/LOG ANALYZER) *
* NOTE: THESE FIELDS ARE CONTAINED WITHIN THE EXTRACT RECORD *
*------------------------------------------------------------------------------*
* 25  EXT-LOG-ERE-ID       PIC X(04).
* 30  EXT-LOG-ERE-JOB-NAME PIC X(08).
* *** THE FOLLOWING DATES ARE BINARY FIELDS, IN JULIAN FORMAT (YYDDD)
* *** FOR OS      THE TIME IS IN 1/100 SEC
* *** FOR OS      IE. DIVISION BY 100 YIELDS HHMMSS
* *** FOR DOS BATCH/CICS, THE TIME IS IN UNITS OF 1/10000 SEC *
* 30  EXT-LOG-ERE-JOB-START-TIME PICX (03).
* 30  EXT-LOG-ERE-JOB-START-DATE PIC S9(5) COMP-.
* 30  EXT-LOG-ERE-JOB-STEP-TIME PIC X(03).
* 30  EXT-LOG-ERE-ID       PIC X(04).
* 88  EXT-LOG-ERE-VM VALUE 'V'.
* 88  EXT-LOG-ERE-CICS VALUE 'C'.
* 30  EXT-LOG-ERE-EXT-ONL.
* 35  EXT-LOG-ERE-TRAN-ID PIC X(04).
* 35  EXT-LOG-ERE-TERM-ID PIC X(04).
*------------------------------------------------------------------------------*
Billing the Record File

This appendix provides a description of the CA IDMS Log Analyzer Billing Record File.

Altering this record layout is necessary if you want to tailor the CA IDMS Log Analyzer Billing file.

---

```
02 BILLING-RECORD-INIT VALUE LOW-VALUE PIC X(100).
   BILLING-RECORD-R REDEFINES BILLING-RECORD-INIT SYNC.
   05 BILL-JOB-NAME PIC X(08).
   05 BILL-RU-START.
      10 BILL-RU-START-DATE PIC S9(7) COMP-3.
      10 BILL-RU-START-TIME PIC S9(9) COMP.
   05 BILL-RU-STOP PIC S9(18) COMP.
      05 BILL-CPU-TIME PIC S9(9) COMP.
      05 BILL-TOTAL-IO PIC S9(9) COMP.
      05 BILL-ONL-BTC PIC S9(9) COMP.
      05 BILL-IDMS-TASK-ID PIC S9(9) COMP.
      05 BILL-ONL-BTC PIC X(01).
      88 BILL-BTC VALUE 'B'.
      88 BILL-CICS VALUE 'C'.
```
I/O Modules

Included here are specific names of I/O modules needed in the core image library by CA IDMS Log Analyzer.

Also shown are the conditions under which each I/O module is needed and typical JCL for linking the module into the core image library.

IJCFZII0 for Parameter card input

IJDFAZIZ for Printer output

IJFVZZZZ for Archive file input on tape and for EXTRACT file input on tape

IJFFZZZZ for Sequential log file input on tape

IJFVZZWZ for EXTRACT file output on tape and for Billing file output on tape if the Billing file is customized to have variable length records.

IJFFZZWZ for Billing file output on tape

// OPTION CATAL,NODUMP,NOFASTTR ACTION MAP,AUTO PHASE xxxxxxxxx,* INCLUDE xxxxxxxx/ // EXEC LNKEDT

Where xxxxxxxx is the name of I/O module to be linked.
IDMS Performance Monitor System Reports

If statistics are written to the DDLDCLOG area during online processing, they are maintained in the DDLDCLOG area of the dictionary. Statistics are written to the DDLDCLOG area as follows:

- Interval Monitor -- Performance Monitor writes statistics to DDLDCLOG if the system administrator specifies IMDCLOG=YES in the #PMOPT macro.
- Application Monitor -- Performance Monitor writes statistics to DDLDCLOG if the system administrator specifies AMDCLOG=YES in the #PMOPT macro.

To archive the statistics from the DDLDCLOG area, use the batch component of the command facility to enter the ARCHIVE LOG statement.

- For more information on ARCHIVE LOG, see the CA IDMS Utilities section.
- For more information on the command facility, see the CA IDMS Common Facilities section.

Sample JCL

You can use the sample JCL in the following sections to archive the statistics from the log area. Remember to supply the appropriate values for variables (shown in italics). Descriptions of variables are provided.

Archiving -- z/OS

IDMSBCF (ARCHIVE LOG STATEMENT) (z/OS)

```plaintext
//archlog EXEC PGM=IDMSBCF,REGION=1024K
//STEPLIB DD DSN=idsms.dba.loadlib,DISP=SHR
// DD DSN=idsms.custom.loadlib,DISP=SHR
// DD DSN=idsms.cagjload,DISP=SHR
//dcllog DD DSN=idsms.system.ddldclog,DISP=SHR
//dcmmsg DD DSN=idsms.sysmsg.ddldcmsg,DISP=SHR
//secdd DD DSN=idsms.sysuser.ddlsec,DISP=SHR
//sysjrn1 DD DUMMY
//SYS001 DD DUMMY
//SYS002 DD DSN=idsms.archive,DISP=(NEW,CATLG),UNIT=tape,
// DCB=(RECFM=VB,LRECL=280,BLKSIZE=23244)
//SYSLST DD SYSOUT=A
//SYSIDMS DD *
DMCL=dmcl-name Other SYSIDMS parameters, as appropriate/*
//SYSIPT DD *
ARCHIVE LOG;
/*
```

For more information on IDMSBCF (the batch command facility), see the CA IDMS Common Facilities section.
### Archive

**Name of job step for archiving**

**idms.dba.loadlib**

Dataset name of the load library containing the DMCL and database name table load modules

**idms.cagjload**

Dataset name of the load library containing the CA IDMS executable modules that do not require customization

**idms.custom.loadlib**

Dataset name of the load library containing the customized CA IDMS executable modules

**dclog**

Ddname of the log area of the dictionary

**idms.system.ddldclog**

Dataset name of the log area of the dictionary

**dcmgs**

Ddname of the system message (DDLDCMSG) area

**idms.sysmsg.ddlcmmsg**

Dataset name of the system message (DDLDCMSG) area

**secdd**

Ddname of the user catalog (required if security is turned on)

**idms.sysuser.ddlsec**

Dataset name of the user catalog; this dataset can be defined dynamically through the DMCL

**sysjrnl**

Ddname of the journal file

**idms.archive**

Name of the archive file

**tape**

Symbolic device name of the archive dataset file

**dmcl-name**

Name of the DMCL load module to use in local mode

---

**Note:** For more information on all SYSIDMS parameters, see the *CA IDMS Database Administering section*.  

---

## Archiving Statistics From the DDLDCCLOG Area

### Archiving -- z/VSE

**IDMSBCF (ARCHIVE LOG STATEMENT) (z/VSE)**

```
// EXEC PROC=IDMSLBLS
// TLBL sysjrnl,'idms.tapejrnl',nnnnnn,f:// ASSGN SYS008,TAPE,VOL=nnnnnn:// ASSGN SYS012,IGN
// ASSGN SYS009,IGN
// ASSGN SYS001,IGN
// TLBL V002,'idms.archive'
// ASSGN SYS002,'ttt'
// EXEC IDMSBCF,SIZE=1024K
ARCHIVE LOG;
/*
```
**Note:** For more information on IDMSBCF (the batch command facility), see the *Using section.*

**IDMSLBS** Name of the procedure (provided at installation) that contains the file definitions for CA IDMS dictionaries and databases. **Note:** For a complete listing of IDMSLBS, see "IDMSLBS procedure" that follows.

<table>
<thead>
<tr>
<th><strong>sysjrnl</strong></th>
<th>Name of the tape journal file</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>idms. tapejrnl</strong></td>
<td>ID of the tape journal file</td>
</tr>
<tr>
<td><strong>nnnnnn</strong></td>
<td>Volume serial number</td>
</tr>
<tr>
<td><strong>f</strong></td>
<td>Number of the tape journal file</td>
</tr>
<tr>
<td><strong>idms. archive</strong></td>
<td>Name of archive tape</td>
</tr>
<tr>
<td><strong>ttt</strong></td>
<td>Physical device assignment</td>
</tr>
</tbody>
</table>

**Runtime parameters**

IDMSLBS references the SYSIDMS file, a file in which you can specify parameters that describe physical requirements (such as DMCL or dictionary to access), runtime parameters, or operating system-specific file information. For this job stream, you should specify the DICTNAME parameter.

**Note:** For more information on all SYSIDMS parameters, see the *CA IDMS Database Administering section.*

**IDMSLBS procedure**

IDMSLBS is a procedure that contains file definitions for the dictionaries, sample databases, disk journal files, and SYSIDMS file provided during installation.

You can tailor the following IDMSLBS procedure (provided on the installation media) to reflect the filenames and definitions in use at your site. Reference IDMSLBS as shown in the previous z/VSE JCL job stream.

```
* LIBDEFS
 /* LIBDEF * ,SEARCH=idmslib.sublib
 /* LIBDEF * ,CATALOG=user.sublib
/* LABELS
 /* DLBL idmslib,'idms.library',yyy/ddd
 /* EXTENT ,nnnnnn, ,ssss,1500
 /* DLBL dccat,'idms.system.dccat',yyy/ddd,DA
 /* EXTENT SYSnnn,nnnnnn, ,ssss,31
 /* ASSGN SYSnnn,DISK, VOL=nnnnnn, SHR
 /* DLBL dccatx,'idms.system.dccatx',yyy/ddd,DA
 /* EXTENT SYSnnn,nnnnnn, ,ssss,6
 /* ASSGN SYSnnn,DISK, VOL=nnnnnn, SHR
 /* DLBL dcdml, 'idms.system.ddldml',yyy/ddd,DA
 /* EXTENT SYSnnn,nnnnnn, ,ssss,11
 /* ASSGN SYSnnn,DISK, VOL=nnnnnn, SHR
```
idmslib.sublib  Name of the sublibrary within the library containing CA IDMS modules
user.sublib  Name of the sublibrary within the library containing user modules
idmslib  Name of the file containing CA IDMS modules
idms.library  ID associated with the file containing CA IDMS modules
SYSnnn  Logical unit of the volume for which the extent is effective
nnnnnn  Volume serial identifier of appropriate disk volume
ssss  Starting track (CKD) or block (FBA) of disk extent
dccat  Filename of the system dictionary catalog (DDLCAT) area
idms.system.dccat  ID of the system dictionary catalog (DDLCAT) area
dccatl  Filename of the system dictionary catalog load (DDLCATLOD) area
idms.system.dccatlod  ID of the system dictionary catalog load (DDLCATLOD) area
dccatx  Name of the system dictionary catalog index (DDLCATX) area
idms.system.dccatx  ID of the system dictionary catalog index (DDLCATX) area
dcdml  Name of the system dictionary definition (DDLDMML) area
idms.system.ddldml  ID of the system dictionary definition (DDLDMML) area
dclod  Name of the system dictionary definition load (DDLDCLOD) area
idms.system.ddldclod  ID of the system dictionary definition load (DDLDCLOD) area
dclog  Name of the system log area (DDLDCLOG) area
idms.system.ddldclog  ID of the system log area (DDLDCLOG) area
dcscr  Name of the system scratch (DDLDCSCR) area
idms.system.ddldcscr  ID of the system scratch (DDLDCSCR) area
dmsg  Name of the system message (DDLDCMSG) area
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>idms.sysmsg.ddldcmsg</td>
<td>ID of the system message (DDLDCMSG) area</td>
</tr>
<tr>
<td>dclscr</td>
<td>Name of the local mode system scratch (DDLOCSCR) area</td>
</tr>
<tr>
<td>idms.sysloc.ddlocscr</td>
<td>ID of the local mode system scratch (DDLOCSCR) area</td>
</tr>
<tr>
<td>dirldb</td>
<td>Name of the IDMSDIRL definition (DDLDMML) area</td>
</tr>
<tr>
<td>idms.sysdirl.ddldml</td>
<td>ID of the IDMSDIRL definition (DDLDMML) area</td>
</tr>
<tr>
<td>dirllod</td>
<td>Name of the IDMSDIRL definition load (DDLDCLOD) area</td>
</tr>
<tr>
<td>idms.sysdirl.dirllod</td>
<td>ID of the IDMSDIRL definition load (DDLDCLOD) area</td>
</tr>
<tr>
<td>empdemo</td>
<td>Name of the EMPDEMO area</td>
</tr>
<tr>
<td>idms.empdemo1</td>
<td>ID of the EMPDEMO area</td>
</tr>
<tr>
<td>insdemo</td>
<td>Name of the INSDEMO area</td>
</tr>
<tr>
<td>idms.insdemo1</td>
<td>ID of the INSDEMO area</td>
</tr>
<tr>
<td>orgdemo</td>
<td>Name of the ORGDEMO area</td>
</tr>
<tr>
<td>idms.orgdemo1</td>
<td>ID of the ORGDEMO area</td>
</tr>
<tr>
<td>empldem</td>
<td>Name of the EMPLDEMO area</td>
</tr>
<tr>
<td>idms.sqldemo.empldemo</td>
<td>ID of the EMPLDEMO area</td>
</tr>
<tr>
<td>infodem</td>
<td>Name of the INFODEMO area</td>
</tr>
<tr>
<td>idms.sqldemo.infdemo</td>
<td>ID of the INFODEMO area</td>
</tr>
<tr>
<td>projdem</td>
<td>Name of the PROJDEMO area</td>
</tr>
<tr>
<td>idms.projseg.projdemo</td>
<td>ID of the PROJDEMO area</td>
</tr>
<tr>
<td>indxdem</td>
<td>Name of the INDXDEMO area</td>
</tr>
<tr>
<td>idms.sqldemo.indxdemo</td>
<td>ID of the INDXDEMO area</td>
</tr>
<tr>
<td>sysctl</td>
<td>Name of the SYSCTL file</td>
</tr>
<tr>
<td>idms.sysctl</td>
<td>ID of the SYSCTL file</td>
</tr>
<tr>
<td>secdd</td>
<td>Name of the system user catalog (DDLSEC) area</td>
</tr>
<tr>
<td>idms.sysuser.ddlsec</td>
<td>ID of the system user catalog (DDLSEC) area</td>
</tr>
<tr>
<td>dictdb</td>
<td>Name of the application dictionary definition area</td>
</tr>
<tr>
<td>idms.appldict.ddldml</td>
<td>ID of the application dictionary definition (DDLDMML) area</td>
</tr>
<tr>
<td>dloddb</td>
<td>Name of the appliCSIDMS Name of the SYSIDMS parameter file definition load area</td>
</tr>
<tr>
<td>idms.appldict.ddldclod</td>
<td>ID of the application dictionary definition load (DDLDCLOD) area</td>
</tr>
<tr>
<td>sqldd</td>
<td>Name of the SQL catalog (DDLDCAT) area</td>
</tr>
<tr>
<td>idms.syssql.ddlcat</td>
<td>ID of the SQL catalog (DDLDCAT) area</td>
</tr>
<tr>
<td>sqllod</td>
<td>Name of the SQL catalog load (DDLDCATL) area</td>
</tr>
<tr>
<td>idms.syssql.ddlcatl</td>
<td>ID of SQL catalog load (DDLDCATL) area</td>
</tr>
<tr>
<td>sqlxdd</td>
<td>Name of the SQL catalog index (DDLDCATX) area</td>
</tr>
<tr>
<td>idms.syssql.ddlcatx</td>
<td>ID of the SQL catalog index (DDLDCATX) area</td>
</tr>
<tr>
<td>asfdml</td>
<td>Name of the asf dictionary definition (DDLDMML) area</td>
</tr>
</tbody>
</table>
Archiving -- z/VM

IDMSBCF (ARCHIVE LOG STATEMENT) (z/VM)

*******************************************************************
* ARCHIVE LOG STATEMENT ENTERED USING IDMSBCF                  *
* (creates the input file for running reports)                   *
*******************************************************************
EXEC IDMSFD  
OSRUN  IDMSBCF

Note: For more information on IDMSBCF (the batch command facility), see the Using section.

IDMSFD   Exec which defines all FILEDEFS, TXTLIBs, and LOADLIBs required by the system

Runtime parameters

IDMSFD references the SYSIDMS file, a file in which you can specify parameters that describe physical requirements (such as DMCL or dictionary to access), runtime parameters, or operating system-specific file information. For this job stream, you should specify the DICTNAME parameter.

Note: For more information on all SYSIDMS parameters, see the CA IDMS Database Administering section.
Executing in local mode

For the ARCHIVE LOG statement, you must specify that IDMSBCF is executing in local mode. To specify this, do one of the following:

- Link IDMSBCF with an IDMSOPTI program that specifies local execution mode
- Modify the OSRUN statement, as follows:

  OSRUN  IDMSBCF  PARM='*LOCAL*'

**Note:** This option is valid only if you issue the OSRUN command from a System Product interpreter or an EXEC2 file.

Creating the SYSIPT file

To create the SYSIPT file, enter these z/VM commands:

```
XEDIT sysipt data a (NOPROF
INPUT
.
..Source statements .
.
FILE
```

Using SMF to Archive Statistics (z/OS only)

Under z/OS, statistics are written to the z/OS SMF job accounting file as follows:

- Application Monitor -- If AMSMF=YES is specified in the #PMOPT macro
- Application Monitor SMF record type 30 -- If SMFTY30=YES is specified in the #PMOPT macro
- Application Monitor SMF record type 4 -- If SMFTYP4=YES is specified in the #PMOPT macro
- Interval Monitor -- If IMSMF=YES is specified in the #PMOPT macro

Using PMSMFEX to archive

To archive the statistics from the z/OS SMF file, use the PMSMFEX module which is supplied with Performance Monitor and stored in the dictionary. Sample central version and local mode JCL follow.

**SMF archive using PMSMFEX macro ('Central version')**

```
//*********************************************************
//*                                                       *
//*             SMF ARCHIVE                               *
//*                                                       *
//*       READS THE SMF FILE AND CREATES THE              *
//*       INPUT FILE FOR RUNNING REPORTS                 *
//*                                                       *
```
****** CA IDMS - 19.0 ******

/* *************************************************************/
/* CULPRIT EXEC PGM=CULPRIT,REGION=1024K */
/* STEPLIB DD DSN=idms.dba.loadlib,DISP=SHR */
// DD DSN=idms.custom.loadlib,DISP=SHR */
// DD DSN=idms.cagjload,DISP=SHR */
// SORTLIB DD DSN=sysl.sortlib,DISP=SHR */
// SYOUT DD SYSOUT=A */
// SYSPRINT DD SYSOUT=A */
// SORTMSG DD SYSOUT=A */
// SYS004 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
// SYS005 DD DSN=&.USRPRMWORK,,DISP=(NEW,DELETE),
// UNIT=disk,SPACE=(CYL,(5,2)),
// DCB=(RECFM=F,B,LRECL=320,BLKSIZE=1600)
// SYS006 DD DSN=&.USRTPWORK,,DISP=(NEW,DELETE),
// UNIT=disk,SPACE=(CYL,(5,2)),
// DCB=(RECFM=F,B,LRECL=2044,BLKSIZE=4628)
// SYS007 DD DSN=&.USRRTWORK,,DISP=(NEW,DELETE),
// UNIT=disk,SPACE=(CYL,(5,2)),
// DCB=(RECFM=F,B,LRECL=80,BLKSIZE=80)
// SYS008 DD DSN=&.USRSTWORK,,DISP=(NEW,DELETE),
// UNIT=disk,SPACE=(CYL,(5,2)),
// DCB=(RECFM=F,B,LRECL=512,BLKSIZE=4628)
// SORTWK01 DD DSN=&.USRWRKAWORK,,
// UNIT=disk,SPACE=(CYL,(5,2))
// SORTWK02 DD DSN=&.USRWRKBWORK,,
// UNIT=disk,SPACE=(CYL,(5,2))
// SORTWK03 DD DSN=&.USRWKVWORK,,
// UNIT=disk,SPACE=(CYL,(5,2))
// SORTWK04 DD DSN=&.USRWKRWORK,,
// UNIT=disk,SPACE=(CYL,(5,2))
// CULSRRT1I DD DSN=yourHLQ.CAGJSRC(SORT1),DISP=SHR
// SYS04 DD DUMMY,DCB=BLKSIZE=80
// VSAMCTRL DD DUMMY */
// CULLIB DD DSN=yourHLQ.CAGJSRC,DISP=SHR */
// syst1 DD DSN=idms.sysctl,DISP=SHR */
// dcmssl DD DSN=idms.sysmsg,ddldcmsg,DISP=SHR */
// SYS010 DD DSN=user.smf.file,DISP=SHR */
// SYS011 DD DUMMY */
// SYS020 DD DSN=user.pmsmfex.outfile,DISP=(NEW,CATLG,DELETE),
// UNIT=disk,SPACE=(CYL,(5,2)),
// DCB=(RECFM=F,B,LRECL=280,BLKSIZE=23244)
// SYS030 DD DSN=user.pmsmfex.outsmf30,DISP=(NEW,CATLG,DELETE),
// UNIT=disk,SPACE=(CYL,(5,2)),
// DCB=(RECFM=F,B,LRECL=698,BLKSIZE=23038)
// SYS040 DD DSN=user.pmsmfex.outsmf4,DISP=(NEW,CATLG,DELETE),
// UNIT=disk,SPACE=(CYL,(5,2)),
// DCB=(RECFM=F,B,LRECL=259,BLKSIZE=23055)
// SYSIDMS DD * */
// SYSIDMS DD * DMCL=dmcl-nameOther SYSIDMS parameters, as appropriate/*
// SYSIN DD * DATABASE DICTNAME=sysdirl=MACRO 'PMSMFEX' (nnn)
  =MEND */

<table>
<thead>
<tr>
<th>idms.dba.loadlib</th>
<th>Name of the load library containing the DMCL and the database name table load modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>idms.cagjload</td>
<td>Name of the load library containing the CA IDMS executable modules that do not require customization</td>
</tr>
<tr>
<td>idms.custom.loadlib</td>
<td>Name of the load library containing customized CA IDMS executable modules</td>
</tr>
<tr>
<td>yourHLQ.CAGJSRC</td>
<td>CA IDMS source library</td>
</tr>
<tr>
<td>sysctl</td>
<td>Ddname of the SYSCTL file</td>
</tr>
</tbody>
</table>
### CA IDMS - 19.0

<table>
<thead>
<tr>
<th><strong>idms.sysctl</strong></th>
<th>Dataset name of the SYSCTL file</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>dcmmsg</strong></td>
<td>Ddname of the dictionary message area</td>
</tr>
<tr>
<td><strong>idms.sysmsg</strong></td>
<td>Dataset name of the dictionary message area (DDLDCMSG)</td>
</tr>
<tr>
<td><strong>ddldcmsg</strong></td>
<td></td>
</tr>
<tr>
<td><strong>sys1.sortlib</strong></td>
<td>System sort library</td>
</tr>
<tr>
<td><strong>disk</strong></td>
<td>Symbolic device name of the file</td>
</tr>
<tr>
<td><strong>user.smf.file</strong></td>
<td>z/OS SMF job accounting file</td>
</tr>
<tr>
<td><strong>user.pmsmfex.outfile</strong></td>
<td>Name of the file created by the extract</td>
</tr>
<tr>
<td><strong>user.pmsmfex.outsmf30</strong></td>
<td>Name of the file created by the extract - SMF records type 30</td>
</tr>
<tr>
<td><strong>user.pmsmfex.outsmf4</strong></td>
<td>Name of the file created by the extract - SMF records type 4</td>
</tr>
<tr>
<td><strong>dmcl-name</strong></td>
<td>Name of the DMCL to access at runtime</td>
</tr>
<tr>
<td><strong>dictionary-name</strong></td>
<td>Name of the dictionary to access (probably SYSDIRL)</td>
</tr>
<tr>
<td><strong>nnn</strong></td>
<td>SMF user record type coded on the #PMOPT macro SMFRCID parameter; the default is 230</td>
</tr>
</tbody>
</table>

**Note:** For more information on other SYSIDMS parameters, see the CA IDMS Database Administering section.

| **Warning:** If the input SMF file to the SMF extract was created as a variable blocked spanned (VBS) file (RECFM=VBS), you must include the parameter DCB=BFTEK=A in the SYS010 DD statement for the `user.smf.file` dataset. Alternatively, add BFTEK=A to existing data control block (DCB) parameters.

If the CA Culprit dictionary security option is turned on in the dictionary that contains the SMF extract report source, a PROFILE statement naming an authorized user and password is required.

**SMF archive using PMSMFEX macro (‘Local mode’)**

```plaintext
//CULPRIT EXEC PGM=CULPRIT,REGION=1024K
//STEPLIB DD DSN=idms.dba.loadlib,DISP=SHR
// DD DSN=idms.custom.loadlib,DISP=SHR
// SORTLIB DD DSN=sys1.sortlib,DISP=SHR
//SYSOUT DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//SORTPRNT DD SYSOUT=A
//SORTMSG DD SYSOUT=A
//SORT005 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133),UNIT=disk,SPACE=(CYL,(5,2)),
// DB=(RECFM=FB,LRECL=320,BLKSIZE=1600)
//SYS006 DD SYSOUT=A,DCB=(RECFM=FB,LRECL=320,BLKSIZE=1600)
//SYS007 DD SYSOUT=A,DCB=(RECFM=FB,LRECL=320,BLKSIZE=1600)
//SYS008 DD SYSOUT=A,DCB=(RECFM=FB,LRECL=320,BLKSIZE=1600)

//SYS007 DD SYSOUT=A,DCB=(RECFM=FB,LRECL=320,BLKSIZE=1600)
//SYS008 DD SYSOUT=A,DCB=(RECFM=FB,LRECL=320,BLKSIZE=1600)
```
For descriptions of variables, see the preceding JCL for executing under the central version.

Using PMSMFEX to extract data for a specific CV or system

To use PMSMFEX to extract data for a specific central version or DC system, you must modify the source statements for the PMSMFEX module stored in the dictionary.

- Extracting Interval Monitor data -- To extract data for the Interval monitor by central version number or DC system number, change the source statements for PMSMFEX as follows:

```plaintext
00$ INTERVAL MONITOR RECORD SELECTION
00$ UNCOMMENT AND CHANGE FOLLOWING CARDS TO SELECT BY SYSTEM
00$ VERSION. SPECIFY THE SYSTEM NUMBER(S) DESIRED IN HEX.
00$ EX:  007135  IF SMFHDCV# EQ X'0010' 150
00$ 007139  DROP  $ DON'T WANT THIS ONE
00$ WILL SELECT ONLY RECORDS FOR DC SYSTEM VERSION # 16
00$
00$ 005135  IF SMFHDCV# EQ X'NNNN' 150  $ WANT THIS
00$ 005136  IF SMFHDCV# EQ X'NNNN' 150  $ WANT THIS
00$ 005137  IF SMFHDCV# EQ X'NNNN' 150  $ WANT THIS
00$ 005138  IF SMFHDCV# EQ X'NNNN' 150  $ WANT THIS
00$ 005139  DROP  $ NOT THIS
00$--------------------------
00$ UNCOMMENT AND CHANGE FOLLOWING CARDS TO SELECT BY CV
00$ VERSION. SPECIFY THE CV NUMBER(S) DESIRED IN HEX.
00$ EX:  007135  IF SMFHCV# EQ X'10' 150
00$ 007139  DROP  $ DON'T WANT THIS ONE
00$ WILL SELECT ONLY RECORDS FOR DC VERSION # 16
00$
00$ 005135  IF SMFHCV# EQ X'NN' 150  $ WANT THIS
00$ 005136  IF SMFHCV# EQ X'NN' 150  $ WANT THIS
```
• Extracting Application Monitor data -- To extract data for the Application monitor by central version number or DC system version number, change the source statements for PMSMFEX as follows:

```
00$ APPLICATION MONITOR RECORD SELECTION
00$ UNCOMMENT AND CHANGE FOLLOWING CARDS TO SELECT BY SYSTEM
00$ VERSION. SPECIFY THE SYSTEM NUMBER(S) DESIRED IN HEX.

EX: 007235 IF SMFHDCV# EQ X'0010' 240
00$ 007239 DROP $ DON'T WANT THIS ONE
00$ WILL SELECT ONLY RECORDS FOR DC SYSTEM VERSION # 16
00$
00$235 IF SMFHDCV# EQ X'NNNN' 240 $ WANT THIS
00$236 IF SMFHDCV# EQ X'NNNN' 240 $ WANT THIS
00$237 IF SMFHDCV# EQ X'NNNN' 240 $ WANT THIS
00$238 IF SMFHDCV# EQ X'NNNN' 240 $ WANT THIS
00$239 DROP $ NOT THIS
00$--------------------------
00$ UNCOMMENT AND CHANGE FOLLOWING CARDS TO SELECT BY CV
00$ VERSION. SPECIFY THE CV NUMBER(S) DESIRED IN HEX.

EX: 007235 IF SMFHCV# EQ X'10' 240
00$ 007239 DROP $ DON'T WANT THIS ONE
00$ WILL SELECT ONLY RECORDS FOR DC VERSION # 16
00$
00$235 IF SMFHCV# EQ X'NN' 240 $ WANT THIS
00$236 IF SMFHCV# EQ X'NN' 240 $ WANT THIS
00$237 IF SMFHCV# EQ X'NN' 240 $ WANT THIS
00$238 IF SMFHCV# EQ X'NN' 240 $ WANT THIS
00$239 DROP $ NOT THIS
```

Sample Job Streams For Running Reports

Contents

- Running reports -- z/OS (see page 356)
- Running reports -- z/VSE (see page 360)
- Running reports -- z/VM (see page 361)

This section provides sample job streams for running reports under:

- z/OS
- z/VSE
- z/VM

Running reports -- z/OS

**CULPRIT for running Performance Monitor reports (z/OS)**

```
//******************************************************************************
//*                                                                           *
//*                      Performance Monitor REPORTS                           *
//*                                                                           *
//* THE JOB EXECUTES THE CULPRIT REPORTS USING THE ARCHIVE                     *
//* FILES AS INPUT AND PRODUCES THE REPORTS AND/OR A                             *
******************************************************************************
```
MACHINE-READABLE FILE AS OUTPUT. THE USER HAS THE RESPONSIBILITY OF DEFINING THE FOLLOWING OPTIONS:

1. REPORTS SELECTION - //SYSIN DD *
   EACH REPORT REQUESTED IS SPECIFIED BY AN = COPY PARAMETER INSERTED IMMEDIATELY AFTER THE SYSIN DD * STATEMENT:
   APPLICATION MONITOR .. PMARPTnn
   INTERVAL MONITOR ..... PMIRPTnn

2. SELECTION CRITERIA - //SYS010 DD
   SELECTION CRITERIA ARE SPECIFIED BY THE SELECTION CRITERIA PARAMETER CARDS INSERTED IMMEDIATELY AFTER THE SYS010 DD * STATEMENT. TO SPECIFY NO SELECTION CRITERIA, INCLUDE THE FOLLOWING:
   //SYS010 DD DUMMY

3. ARCHIVE INPUT SET DEFINITION - //SYS011 DD DSN=
   DEFINE THE ARCHIVE FILES BY CODING THE FOLLOWING:
   //SYS011 DD DSN=idms.archive,DISP=OLD
   // DD DSN=idms.archiven,DISP=OLD
   // UNIT=AFF=SYS011

4. MACHINE-READABLE OUTPUT SET (PMARPT90)
   DEFINITION - //SYS020 DD DSN=
   TO PRODUCE MACHINE-READABLE OUTPUT, THE OUTPUT FILE MUST BE DEFINED AS FOLLOWS:
   //SYS020 DD DSN=rpt90.output.dataset

CULPRIT EXEC PGM=CULPRIT,REGION=1024K
STEPLIB DD DSN=idms.dba.loadlib,DISP=SHR
   DD DSN=idms.custom.loadlib,DISP=SHR
   DD DSN=idms.cagjload,DISP=SHR
SORTLIB DD DSN=sys1.sortlib,DISP=SHR
SYSOUT DD SYSOUT=A
SYSPRINT DD SYSOUT=A
SORTPRNT DD SYSOUT=A
SORTMSG DD SYSOUT=A
SYS004 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
SYS005 DD DSN=&.&SUPRMWORK.,DISP=(NEW,DELETE),
   UNIT=DISK,SPACE=(CYL,(5,2)),
   DCB=(RECFM=F8,LRECL=320,BLKSZ=1600)
SYS006 DD DSN=6.&UXTWWORK.,DISP=(NEW,DELETE),
   UNIT=DISK,SPACE=(CYL,(5,2)),
   DCB=(RECFM=VB,LRECL=2044,BLKSZ=4628)
SYS007 DD DSN=6.&SRTPWWORK.,DISP=(NEW,DELETE),
   UNIT=DISK,SPACE=(TRK,(1,1)),
   DCB=(RECFM=F,LRECL=80,BLKSZ=80)
SYS008 DD DSN=6.&NSRTWORK.,DISP=(NEW,DELETE),
   UNIT=DISK,SPACE=(CYL,(5,2)),
   DCB=(RECFM=VB,LRECL=512,BLKSZ=4628)
SORTWK01 DD DSN=6.&WRKAWORK.,
   UNIT=DISK,SPACE=(CYL,(5,2))
SORTWK02 DD DSN=6.&WRKAWORK.,
   UNIT=DISK,SPACE=(CYL,(5,2))
SORTWK03 DD DSN=6.&WRKAWORK.,
   UNIT=DISK,SPACE=(CYL,(5,2))
SORTWK04 DD DSN=6.&WRKAWORK.,
   UNIT=DISK,SPACE=(CYL,(5,2))
CULSRT11 DD DSN=yourHLO.CAGJSRC(SORT1),DISP=SHR
SYSIN04 DD DUMMY,DCB=BLKSZ=80
YSAAMCTD DD DUMMY
CULLIB DD DSN=yourHLO.CAGJSRC,DISP=SHR
SYSCtl DD DSN=idms.sysctl,DISP=SHR
IDCMSG DD DSN=idms.sysmsg.ddldcmsg,DISP=SHR
SYS010 DD *
REPORT FROM 09:00 ON 5/15/10

SYS011 DD DSN=idms.archive,DISP=OLD,UNIT=tape
SYS020 DD DSN=rpt90.output.dataset,DISP=(NEW,CATLG,DELETE),
idms.dba.loadlib  Name of the load library containing the DMCL and database name table load modules
idms.custom.loadlib  Name of the load library containing customized CA IDMS executable modules
idms.cagjload  Name of the load library containing CA IDMS executable modules that do not require customization
yourHLQ.CAGJSRC  CA IDMS source library
sysctl  The ddname of the SYSCTL file
idms.sysctl  Dataset name of the SYSCTL file
dcmg  Ddname of the dictionary message area (DDLDCMSG)
idms.sysmsg.ddldcmg  Filename of the dictionary message area (DDLDCMSG)

sys1.sortlib  System sort library
idms.archive  Names of archive logs (n is nth log)
idms.archiven  Names of archive logs (n is nth log)
rpt90.output.dataset  Machine-readable output
dmcl-name  Name of the DMCL

Note: For more information on other SYSIDMS parameters, see the CA IDMS Database Administering section.

dictionary-name  Dictionary name (probably SYSDIRL)

z/OS blocksize considerations

The input JCL for the statistics input file (SYS011) must specify a DCB=BLKSIZE=nnnnn parameter. Nnnnn must be at least 280 bytes larger than the actual block size of the file. If the DCB specified is not large enough, CA Culprit may receive an OC4 abend.

For example, if the input file has a blocksize of 11476, an appropriate SYS011 DD statement is:

//SYS011   DD   DSN=PM.STATS,DISP=OLD,DCB=BLKSIZE=12000

Executing in local mode
CULPRIT for running Performance Monitor reports (z/OS)

//CULPRIT EXEC PGM=CULPRIT,REGION=1024K
//STEPLIB DD DSN=idms.dbz.loadlib,DISP=SHR
// DD DSN=idms.custom.loadlib,DISP=SHR
// DD DSN=idms.cagjload,DISP=SHR
//SORTLIB DD DSN=sys1.sortlib,DISP=SHR
//SYSOUT DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//SORTMSG DD SYSOUT=A
//SYS004 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
//SYS005 DD DSN=&.SUPRMWORK.,DISP=(NEW,DELETE),
// UNIT=disk,SPACE=(CYL,(5,2)),
// DCB=(RECFM=FB,LRECL=320,BLKSIZE=1600)
//SYS006 DD DSN=&.UEXTWORK.,DISP=(NEW,DELETE),
// UNIT=disk,SPACE=(CYL,(5,2)),
// DCB=(RECFM=FB,LRECL=2044,BLKSIZE=4628)
//SYS007 DD DSN=&.SRTPWORK.,DISP=(NEW,DELETE),
// UNIT=disk,SPACE=(TRK,(1,1)),
// DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYS008 DD DSN=&.NSRTWORK.,DISP=(NEW,DELETE),
// UNIT=disk,SPACE=(CYL,(5,2)),
// DCB=(RECFM=FB,LRECL=512,BLKSIZE=4628)
//SORTWK01 DD DSN=&.WRKAWORK.,
// UNIT=disk,SPACE=(CYL,(5,2))
//SORTWK02 DD DSN=&.WRKBWORK.,
// UNIT=disk,SPACE=(CYL,(5,2))
//SORTWK03 DD DSN=&.WRKCWORK.,
// UNIT=disk,SPACE=(CYL,(5,2))
//SORTWK04 DD DSN=&.WRKDCWOK.,
// UNIT=disk,SPACE=(CYL,(5,2))
//CULSRT1I DD DSN=yourHLQ.CAGJSRC(SORT1),DISP=SHR
//SYSIN4 DD DUMMY,DCB=BLKSIZE=80
//VSAMCTRL DD DUMMY
//CULLIB DD DSN=yourHLQ.CAGJSRC,DISP=SHR
//dirldb DD DSN=idms.sysdirl.ddldml,DISP=SHR
//dcmsg DD DSN=idms.sysmsg.ddldcmsg,DISP=SHR
//sysjrnl DD DUMMY
//SYS010 DD *
REPORT FROM 09:00 ON 5/15/10
/*/ 
//SYS011 DD DSN=idms.archive1,DISP=OLD,UNIT=tape
//SYS020 DD DSN=rpt90.output.dataset,DISP=(NEW,CATLG,DELETE),
// UNIT=disk,SPACE=(CYL,(1,1)),
// DCB=(RECFM=VB,LRECL=280,BLKSIZE=23244)
//SYS020 DD DUMMY
//SYSIDMS DD *
DMCL=dmcl-nameOther SYSIDMS parameters, as appropriate/* 
//SYSIN DD *
DATABASE DICTNAME=sysdirl PARAM=NOLIST
=COPY 'PMIRPT00'
=COPY 'PMNAME'
=COPY 'PMIRPT01'

=COPY 'PMIRPT99'
/*
//* 

⚠️ **Note:** For more information on the descriptions of variables, see the preceding JCL for running under the central version.
Running reports -- z/VSE

CULPRIT for running Performance Monitor reports (z/VSE)

*******************************************************************
* Performance Monitor REPORTS                               *
*                                                              *
* THIS JOB READS THE FILE CREATED IN THE PREVIOUS           *
* STEP AND PRODUCES THE CULPRIT REPORTS.                   *
*                                                              *
*******************************************************************

// EXEC PROC=IDMSLBLS
// DLBL SORTWK1, 'SORTWK1', 0, SD
// EXTENT SYSnnn, vvvvvv, 1, 0, sssss, nnnn// ASSGN SYS011, TAPE
// TLBL SYS011, 'ARCHIVE,PRINTLOG'
// ASSGN SYS020, TAPE
// TLBL SYS020, 'OUTPUT.TAPE'
// DLBL SYS005, 'SCRATCH1', 0
// EXTENT SYS005, vvvvvv, sssss, nnnn// ASSGN SYS005, DISK, VOL=vvvvvv, SHR
// DLBL SYS006, 'SCRATCH2', 0
// EXTENT SYS006, vvvvvv, sssss, nnnn// ASSGN SYS006, DISK, VOL=vvvvvv, SHR
// DLBL SYS007, 'SCRATCH3', 0
// EXTENT SYS007, vvvvvv, sssss, nnnn// ASSGN SYS007, DISK, VOL=vvvvvv, SHR
// DLBL SYS008, 'SCRATCH4', 0
// EXTENT SYS008, vvvvvv, sssss, nnnn// ASSGN SYS008, DISK, VOL=vvvvvv, SHR
// ASSGN SYS004, SYSLST
// ASSGN SYS010, SYSIPT
// UPSI 1
// EXEC CULPRIT, SIZE=400K
// PARAM=NOLIST
// COPY 'PMIRPT00'
// COPY 'PMIRPT01'
// COPY 'PMIRPT02'
// COPY 'PMIRPT99'
// REPORT FROM 09:00 ON 5/15/10
//

<table>
<thead>
<tr>
<th>nnnn</th>
<th>Number of tracks (CKD) or blocks (FBA) in disk extent</th>
</tr>
</thead>
<tbody>
<tr>
<td>sssss</td>
<td>Starting track (CKD) or block (FBA) of disk extent</td>
</tr>
<tr>
<td>vvvvvv</td>
<td>Volume serial number</td>
</tr>
</tbody>
</table>

IDMSLBLS procedure

The IDMSLBLS procedure (provided at installation) contains the file definitions for CA IDMS dictionaries and databases.

For more information:

Archiving -- z/VSE (see page ).

IDMSLBLS references the SYSIDMS parameters file. In SYSIDMS, you can specify physical requirements (such as DMCL or dictionary), runtime parameters, and operating system-dependent file information.
Running reports -- z/VM

CULPRIT for running Performance Monitor reports (z/VM)

FILEDEF SYSIN DISK sysin data a (LRECL 80 BLKSIZE 80 RECFM F)
EXEC CULPFD
OSRUN CULPRIT

<table>
<thead>
<tr>
<th>sysin data a</th>
<th>Filename, type, and mode of the file containing CA Culprit statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>CULPFD</td>
<td>Exec which defines all file definitions required by the system</td>
</tr>
</tbody>
</table>

Runtime parameters

CULPFD references the SYSIDMS parameters file. In this file you can specify physical requirements (like DMCL or dictionary), runtime parameters, and operating system-dependent file information.

Executing in local mode

To specify that CA Culprit is executing in local mode, do one of the following:

- Link CA Culprit with an IDMSOPTI program that specifies local execution mode.
- Specify *LOCAL* as the first input parameter of the sysin data a file identified in the FILEDEF SYSIN statement.
- Modify the OSRUN statement:
  OSRUN CULPRIT PARM='*LOCAL'*

**Note:** This option is available only if the OSRUN command is issued from a System Product interpreter or an EXEC2 file.

To create the SYSIN file, enter these z/VM commands:

```
xedit sysin data a (noprof
input
  database cvmach=TS10
  param=list
  =copy 'PMIRPT00'
  =copy 'PMNAME'
```
Replacing the COPY Parameters (z/VSE only)

CA Culprit cannot create variable-length records in a z/VSE environment. Therefore, make the changes described in this section to compensate for fixed-length records.

Changing PMARPT90 and PMIRPT90

Replace the COPY parameters in your job stream with USE and CHANGE statements, as follows:

- **For PMARPT90:**
  ```
  USE 'PMARPT00'
  USE 'PMNAME'
  USE 'PMARPT90'
  CHANGE '90OUT 280 D PS' TO '90OUT 280 8120 D PS'
  USE 'PMARPT99' optional
  ```

- **For PMIRPT90:**
  ```
  USE 'PMIRPT00'
  USE 'PMNAME'
  USE 'PMIRPT90'
  CHANGE '90OUT 280 D PS' TO '90OUT 280 8120 D PS'
  USE 'PMIRPT99' optional
  ```

Changing PMARPT00 and PMIRPT00

Replace the COPY parameters in your job stream with USE and CHANGE statements, as follows:

- **For PMARPT00:**
  ```
  USE 'PMARPT00'
  CHANGE 'IN 280 V 6000' TO 'IN 280 F 8120'
  USE 'PMNAME'
  USE 'PMARPTXX' specify the required report(s) USE 'PMARPT99' optional
  ```

- **For PMIRPT00:**
  ```
  USE 'PMIRPT00'
  CHANGE 'IN 280 V 6000' TO 'IN 280 F 8120'
  USE 'PMNAME'
  USE 'PMIRPTXX' specify the required report(s) USE 'PMIRPT99' optional
  ```
Replacing the COPY Parameters For Tape Input (z/VSE only)

Archive log tapes created under z/VSE have a blocksize of 32760, unless file overrides are specified in the SYSIDMS parameters. Therefore, make the changes described in this section to run Performance Monitor reports with tape input. If SYSIDMS file overrides have been used, substitute that blocksize for 32760.

Changing PMARPT00

Replace the COPY parameters in your job stream with USE and CHANGE statements, as follows:

USE 'PMARPT00'
CHANGE 'IN 280 V 6000' TO 'IN 280 V 32760'
USE 'PMNAME'
USE 'PMARPTXX' specify the required report(s)

Changing PMIRPT00

Replace the COPY parameters in your job stream with USE and CHANGE statements, as follows:

USE 'PMIRPT00'
CHANGE 'IN 280 V 6000' TO 'IN 280 V 32760'
USE 'PMNAME'
USE 'PMIRPTXX' specify the required report(s)

Note For DDR-Only Shops

DDR-only shops must use the CULPRIT USE statement to request the following reports:

- PMIRPT00
- PMARPT00

Instead of using the =COPY statement, as demonstrated by the examples earlier in this section, use the following:

PMIRPT00

PMCULLID for z/VSE

USE 'PMIRPT00'
CHANGE 'IN 80' TO 'IN 363 V 367 UM(PMCULLIM) $'
CHANGE 'REC FILE1-EOF' TO 'REC FILE1-EOF 283 1 2 $'
CHANGE 'CARD-REC 1' TO 'CARD-REC 284'
CHANGE 'CARD-GRP 1' TO 'CARD-GRP 284'
CHANGE 'IN 280' TO '08$'
CHANGE 'IN 457 F 457 PS MB=DUMMY' TO 'IN 457 F 457 MB=DUMMY'
CHANGE 'IN 301 F 301 PS MB=DUMMY' TO 'IN 301 F 301 MB=DUMMY'
USE 'PMNAME'
USE 'PMIRPTnn' Specify the required report(s)
USE 'PMIRPT99' Optional
PMARPT00

USE 'PMARPT00'
CHANGE 'IN 80' TO 'IN 363 V 367 UM(PMCULLIM) $'
CHANGE 'REC FILE1-EOF' TO 'REC FILE1-EOF 283 1 2 $'
CHANGE 'CARD-REC 1' TO 'CARD-REC 284'
CHANGE 'CARD-GRP 1' TO 'CARD-GRP 284'
CHANGE 'IN 280' TO '00$'
CHANGE 'IN 496 F 496 PS MB=DUMMY' TO 'IN 496 F 496 MB=DUMMY'
USE 'PMNAME'
USE 'PMARPTnn' Specify the required report(s)
USE 'PMARPT99' Optional

For more information:

For more information on the reports, see Interval Monitor Batch Reports (see page 407), and Application Monitor Batch Reports (see page 407). For more information on the USE statement, see the CULPRIT Reference.

z/OS blocksize considerations

The input JCL for the statistics input file (SYS011) must specify a DCB=BLKSIZE=nnnn parameter. $nnnn must be at least 280 bytes larger than the actual block size of the file. If the DCB specified is not large enough, CULPRIT may receive an OC4 abend.

Example:

If the input file has a blocksize of 11476, an appropriate SYS011 DD statement is:

```
//SYS011 DD DSN=PM.STATS,
//       DISP=OLD,DCB=BLKSIZE=12000
```

Interval Monitor Batch Reports

You request Interval Monitor reports using a CA Culprit job stream. The job control language you need to run the reports is shown in Preparing to Run Reports (see page 345). In the job stream, you supply:

- Selection criteria parameters -- for including and/or excluding specific information from the reports
- Report specification parameters -- for specifying the dictionary to use, formatting options, and the appropriate report names

You can request any or all of the reports in a single run.

General rules for parameter input

- Every parameter is optional.
- Include any or all of these parameters in a single run.
- Use a single line for each separate parameter.
- If you specify more than one parameter, all conditions that you specify must be met in order for you to select an interval for reporting.

- Use columns 1 through 72. Input beyond column 72 is ignored. No error is flagged (unless a quoted description is truncated).

- An asterisk (*) in column 1 indicates a comment line.

- Specify either the 3-letter abbreviation or the whole word. For example, EXCLUD is invalid. The syntax rules indicate (in uppercase characters) any other allowable abbreviations or synonyms.

- Blank lines are ignored but generate a warning message.

### Selection criteria parameters

Include selection criteria parameters in your CA Culprit JCL to include information in or exclude information from your Performance Monitor reports.

Selection criteria parameters apply to all of the reports you request in the same run. For example, if you specify a time interval using the REPORT FROM/THRU parameter, that interval is used for all of the reports in the run.

#### Positioning selection criteria parameters

Position your selection criteria parameters in the JCL stream as follows:

<table>
<thead>
<tr>
<th>System</th>
<th>Position in JCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS</td>
<td>Following the //SYS010 DD * statement</td>
</tr>
<tr>
<td>z/VSE</td>
<td>Following the /* in the EXEC CULPRIT step</td>
</tr>
<tr>
<td>z/VM</td>
<td>In the SYS010 file</td>
</tr>
</tbody>
</table>

#### When you don't need selection parameters

If you don't need selection parameters for the run, then for:

- z/OS -- Use //SYS010 DD DUMMY
- z/VM -- Leave out the parameters
- z/VSE -- Use SYS010 DUMMY

### Syntax

Descriptions for Interval Monitor selection criteria parameters follow the syntax diagram. You can omit leading zeros where syntax uses a number, unless otherwise noted.

```
►► SPArse
► CV NUMBER dc/ucf-version-number
►► DATe FORmat MDY DMY
```
Parameters

- **SPArse**
  Suppresses display of blank lines in these reports:
  - PMIRPT11 (I/O by Area Summary)
  - PMIRPT12 (I/O by File Summary)
  - PMIRPT14 (CDMSLIB Summary)
  - PMIRPT16 (TP Line Summary)

- **CV NUMber dc/ucf-version-number**
  Identifies the DC/UCF system for which Performance Monitor is to report interval statistics; dc /ucf-version-number is a number between 0 and 9999. You can specify a system version value up to 20 times. You can place multiple values on one line. An acceptable abbreviation for NUMBER is NBR.

- **DATe FORmat MDY/DMY/MDY/DMY/YMD/YDM**
  Specifies the date format that appears on the reports. Additionally, the date format you choose is used for any date specification parameters. For example, if you specify DMY, Performance Monitor expects the REPORT FROM/THRU start-date and end-date to be in the format DMY. The default is MDY. An acceptable abbreviation for FORMAT is FMT.
- **DESCRIPTION report-description**
  Specifies a description to appear in the report footers. *Report-description* is a 1- through 64-character value. If it contains embedded spaces, you must use single quotes. Use two quotation marks to indicate a quotation mark that is part of the description.

- **REPORT FROM/THRU**
  Selects intervals to be included in the report. If you want to report on the entire input file, do not include this parameter. You can specify this parameter once per run, and you must specify at least one FROM or one THRU. The default is FROM 00:00 ON 00/001 THRU 24:00 ON 99/365.
  
  **Regarding the time specification:**
  - Specify the time as *hh:mm* or *hhmm* (00:00 through 24:00).
  - Times include the entire minute. For example, THRU 14:34 means up to 14:34:59.9999.
  - Times must include the leading 0. For example, 09:00 is valid, but 9:00 is not.
  - If you specify a time range, the FROM time must be earlier than the THRU time.

  **Regarding the date specification:**
  - Julian: *yy/ddd*
  - Gregorian: as specified by DATE FORMAT
  - The FROM date must be earlier or matching the THRU date.
  - Slashes are optional in date specifications.

- **EXCLUDE INTERVALS FROM/THRU**
  Specifies intervals to be excluded from the report. EXCLUDE INTERVALS follows the same general rules as REPORT FROM/THRU.

- **SCALING FACTOR nnnn.nnnn**
  Defines a scaling factor for report graphs; *nnnn.nnnn* specifies the scaling factor (for example, .01 scaling data in hundredths). An acceptable synonym for the keyword is SCALE FACTOR.
  *nnnn.nnnn* is a numeric value. The decimal point is not required and, if present, can be leading or trailing. Any more than four digits to the right of the decimal point are truncated. For example, 1.2345678 will be truncated to 1.2345. About the value you can specify:
  - 0 is invalid.
  - The default is 1.0.
  - The maximum is 999999.9999.
  - Examples of valid values:
    - 3456
    - 1234 .5678
    - 000000 .01
    - .3456
    - 45 .
    - 0 .3
**EXClude WAIt TYPe wait-type**
Excludes specified wait types from PMIRPT01 (the Management Summary Report). You can specify multiple wait types and include them all on the same line.

<table>
<thead>
<tr>
<th>Wait type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREA</td>
<td>Area waits</td>
</tr>
<tr>
<td>BUFFER</td>
<td>Buffer waits</td>
</tr>
<tr>
<td>CKUSER</td>
<td>Check-user waits</td>
</tr>
<tr>
<td>DBGROUP</td>
<td>DBGroup waits</td>
</tr>
<tr>
<td>DBKEY</td>
<td>Db-key waits</td>
</tr>
<tr>
<td>DDS</td>
<td>DDS waits</td>
</tr>
<tr>
<td>ERUS</td>
<td>Run unit/request unit waits</td>
</tr>
<tr>
<td>EXTERNAL</td>
<td>External waits (outside the system)</td>
</tr>
<tr>
<td>INTERNAL</td>
<td>Internal waits (in the system)</td>
</tr>
<tr>
<td>IO</td>
<td>I/O waits</td>
</tr>
<tr>
<td>JOURNAL</td>
<td>Journal waits</td>
</tr>
<tr>
<td>JRNLBUF</td>
<td>Journal buffer waits</td>
</tr>
<tr>
<td>LDRSINGLE</td>
<td>Loader single-threaded waits</td>
</tr>
<tr>
<td>LINE</td>
<td>TP line waits</td>
</tr>
<tr>
<td>LOADS</td>
<td>Load-area waits</td>
</tr>
<tr>
<td>LOG</td>
<td>Log waits</td>
</tr>
<tr>
<td>LOGSINGLE</td>
<td>Log single-threaded waits</td>
</tr>
<tr>
<td>LOGFULL</td>
<td>Log full waits</td>
</tr>
<tr>
<td>MAXTASK</td>
<td>Waits because of maxtasks condition</td>
</tr>
<tr>
<td>PGMPOOL</td>
<td>Program-pool waits</td>
</tr>
<tr>
<td>PRIOR</td>
<td>Waits for a prior I/O (z/VSE only)</td>
</tr>
<tr>
<td>QUEUE</td>
<td>Queue-area waits</td>
</tr>
<tr>
<td>SCRATCH</td>
<td>Scratch-area waits</td>
</tr>
<tr>
<td>SCRSINGLE</td>
<td>Scratch single-thread waits</td>
</tr>
<tr>
<td>SHCACHE</td>
<td>Shared cache waits</td>
</tr>
<tr>
<td>STORAGE</td>
<td>Storage waits</td>
</tr>
<tr>
<td>XESLIST</td>
<td>Data sharing XES list waits</td>
</tr>
<tr>
<td>XESLOCK</td>
<td>Data sharing XES lock waits</td>
</tr>
</tbody>
</table>

**INClude WAIt TYPe wait-type**
Specifies that the named wait types be tallied together for PMIRPT02 (the Trend Analysis Report). You can specify multiple wait types and include them all on the same line. See EXCLUDE WAIT TYPE for acceptable wait-type values.
**INClude AREA area-name**  
Includes the specified area or areas in PMIRPT05 (the DBkey/Area Detail Wait report) and PMIRPT11 (the I/O by Area Summary report).  
**General rules:**  
- Specify up to 100 areas, as needed  
- You can have multiple area names on one line  
- Area names can contain as many as 16 characters  
- You cannot specify excludes and includes in a single run  
- Criteria requested for one run applies to both the DBkey/Area and the I/O detail reports  

**EXClude AREA area-name**  
Excludes the specified area or areas from PMIRPT05 (the DBkey/Area Detail Wait report) and PMIRPT11 (the I/O by Area Summary report).  

**INClude FILE file-name**  
Specifies files to be included in PMIRPT09 (the Shared Cache Summary report) and PMIRPT12 (the I/O by File Summary report). A synonym for FILE is FILES. The same rules that apply to INCLUDE FILES also apply to INCLUDE AREAS.  

**EXClude FILE file-name**  
Excludes the specified file or files from PMIRPT09 (the Shared Cache Summary report) and PMIRPT12 (the I/O by File Summary report).  

---  

**Example**  
The parameters below select only those intervals between 9:30 a.m. and 11:30 a.m., on June 16, 1999. The footers include the description PEAK MORNING PROCESSING ONLY, and the areas PAYAREA and PERSAREA are excluded:  

```
REPORT FROM 09:30 ON 6/16/10 THRU 11:30 ON 6/16/10  
REPORT DESCRIPTION 'PEAK MORNING PROCESSING ONLY'  
EXCLUDE AREAS PAYAREA PERSAREA  
```  

---  

**Request Reports (Interval Monitor)**  
**Report selection parameters**  
Report selection parameters define:  
- The dictionary that contains the report definitions  
- Whether to print CA Culprit parameters  
- Which reports to produce  

**Positioning report selection parameters**
Position these parameters in the report-request JCL stream, using one line for each parameter:

<table>
<thead>
<tr>
<th>System</th>
<th>Position in JCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS</td>
<td>Following the //SYSIN DD * statement</td>
</tr>
<tr>
<td>z/VSE</td>
<td>Following the EXEC CULPRIT statement</td>
</tr>
<tr>
<td>z/VM</td>
<td>Following the DATABASE statement</td>
</tr>
</tbody>
</table>

Syntax and parameter descriptions for report selection parameters follow.

**Syntax**

```
<table>
<thead>
<tr>
<th>DATabase</th>
<th>DICTNAME= dictionary-name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARAM=</td>
<td>LIST/NOLIST/EJECT</td>
</tr>
<tr>
<td></td>
<td>=COPY 'report-name'</td>
</tr>
</tbody>
</table>
```

**Parameters**

- **DATabase**
  Defines the data dictionary that contains the report definitions (DICTNAME option) or the node that controls the dictionary (DICTNODE option). Start this parameter in column 2.

- **PARAM=LIST/NOLIST/EJECT**
  Controls printing of the CA Culprit Sequential Input Parameter List:
  - LIST (default) prints all parameters
  - NOLIST prints no parameters
  - EJECT starts each new listing at the top of a new page

  Start this parameter in column 2.

- **=COPY 'report-name'**
  Requests the named report; begin =COPY in column 1; you can repeat the parameter any number of times. Report-name must be enclosed in quotes.

<table>
<thead>
<tr>
<th>Value for report-name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMIRPT00</td>
<td>Performs housekeeping functions and extracts statistics for input to other reports; required, but not an output report</td>
</tr>
<tr>
<td>PMNAME</td>
<td>Supply the user site or company name to be printed in the heading of each report; required, but not an output report</td>
</tr>
<tr>
<td>PMIRPT99</td>
<td>List an input processing summary based on the selection criteria specified</td>
</tr>
<tr>
<td>PMIRPTnn</td>
<td>Produce the report defined by the number (nn) specified: 01 Management Summary Report 02 Trend Analysis Report 04 Detail Wait Report - Summary</td>
</tr>
</tbody>
</table>
### Value for report-name

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>05</td>
<td>Detail Report - Db-key/Area</td>
</tr>
<tr>
<td>09</td>
<td>Summary Report - Shared Cache</td>
</tr>
<tr>
<td>10</td>
<td>Summary Report - DBGroup</td>
</tr>
<tr>
<td>11</td>
<td>Summary Report - I/O by Area</td>
</tr>
<tr>
<td>12</td>
<td>Summary Report - I/O by File</td>
</tr>
<tr>
<td>13</td>
<td>Summary Report - Buffer</td>
</tr>
<tr>
<td>14</td>
<td>Summary Report - CDMSLIB</td>
</tr>
<tr>
<td>15</td>
<td>Summary Report - Journal</td>
</tr>
<tr>
<td>16</td>
<td>Summary Report - TP Line</td>
</tr>
<tr>
<td>17</td>
<td>Summary Report - Program Pool</td>
</tr>
<tr>
<td>18</td>
<td>Summary Report - Storage Pool</td>
</tr>
<tr>
<td>19</td>
<td>Summary Report - Storage Waits</td>
</tr>
<tr>
<td>21</td>
<td>Detail Report - I/O by Area</td>
</tr>
<tr>
<td>22</td>
<td>Detail Report - I/O by File</td>
</tr>
<tr>
<td>23</td>
<td>Detail Report - Buffer</td>
</tr>
<tr>
<td>24</td>
<td>Detail Report - CDMSLIB</td>
</tr>
<tr>
<td>25</td>
<td>Detail Report - Journal</td>
</tr>
<tr>
<td>27</td>
<td>Detail Report - Program Pool</td>
</tr>
<tr>
<td>29</td>
<td>Detail Report - Storage Type</td>
</tr>
<tr>
<td>30</td>
<td>Summary Report - Interval Statistics</td>
</tr>
<tr>
<td>32</td>
<td>Summary Report - Run Unit Statistics</td>
</tr>
<tr>
<td>38</td>
<td>Detail Report - Journal Block Full</td>
</tr>
<tr>
<td>40</td>
<td>Detail Report - Data Sharing SYSPLEX</td>
</tr>
</tbody>
</table>

### Example

The following report parameters select all printed reports. The CA Culprit report definitions are stored in the DICTCAS dictionary (DATABASE DICTNAME=DICTCAS). The report source (PARAM=NOLIST) is not printed.

```plaintext
DATABASE DICTNAME=DICTCAS
PARAM=NOLIST
=COPY 'PMIRPT00'
=COPY 'PMNAME'
=COPY 'PMIRPT99'
=COPY 'PMIRPT01'
=COPY 'PMIRPT02'
=COPY 'PMIRPT04'
=COPY 'PMIRPT05'
=COPY 'PMIRPT09'
=COPY 'PMIRPT10'
=COPY 'PMIRPT11'
=COPY 'PMIRPT12'
=COPY 'PMIRPT13'
=COPY 'PMIRPT14'
=COPY 'PMIRPT15'
=COPY 'PMIRPT16'
=COPY 'PMIRPT17'
=COPY 'PMIRPT18'
=COPY 'PMIRPT19'
=COPY 'PMIRPT21'
=COPY 'PMIRPT22'
=COPY 'PMIRPT23'
=COPY 'PMIRPT24'
=COPY 'PMIRPT25'
=COPY 'PMIRPT27'
=COPY 'PMIRPT29'
=COPY 'PMIRPT30'
```
Report Samples

The remainder of this section describes each report. These two required reports have no output:

- **PMIRPT00** -- Reads the input (archive) tape and formats it into global data fields; the data fields provide the input for all other reports

- **PMNAME** -- Reads the PMNAME module and inserts its contents into a global field called COMPANY-NAME; this produces the heading for each report

Optional reports

The remaining optional reports for the Interval Monitor are described in numeric order. Each report description includes:

- An overview description
- A sample report
- A description of the fields in the report

Contents

- **PMIRPT01** Management summary report (see page 373)
- **PMIRPT02** Trend analysis report (see page 374)
- **PMIRPT04** Summary wait detail report (see page 375)
- **PMIRPT05** DBkey/Area detail report (see page 376)
- **PMIRPT09** Shared cache summary report (see page 377)
- **PMIRPT10** DBGroup summary report (see page 378)
- **PMIRPT11** I/O by area summary report (see page 379)
- **PMIRPT12** I/O by file summary report (see page 380)
- **PMIRPT13** Buffer summary report (see page 381)
- **PMIRPT14** CDMSLIB summary report (see page 383)
- **PMIRPT15** Journal summary report (see page 384)
- **PMIRPT16** TP line summary report (see page 385)
- **PMIRPT17** Program pool summary report (see page 386)
- **PMIRPT18** Storage pool summary report (see page 388)
- **PMIRPT19** Storage waits summary report (see page 389)
- **PMIRPT21** I/O by area detail report (see page 390)
- **PMIRPT22** I/O by file detail report (see page 392)
- **PMIRPT23** Buffer detail report (see page 393)
- **PMIRPT24** CDMSLIB detail report (see page 394)
- **PMIRPT25** Journal detail report (see page 395)
- **PMIRPT27** Program pool detail report (see page 396)
PMIRPT01 Management summary report

PMIRPT01 is a summary report for all wait types not excluded by input selection parameters. The report shows the total wait count and time across all intervals on a graphic representation of wait time for that wait type. If any wait types were excluded by input selection parameters, the word EXCLUDED appears in the graph.

Sample report

<table>
<thead>
<tr>
<th>WAIT TYPE</th>
<th>WAITS</th>
<th>WAIT TIME</th>
<th>SCALED BY TIME ('X' REPRESENTS 1.0000 SECONDS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB I/O</td>
<td>179</td>
<td>3.42</td>
<td>XXX</td>
</tr>
<tr>
<td>PRIOR FILE I/O</td>
<td>0</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>DB BUFFER</td>
<td>4</td>
<td>.23</td>
<td></td>
</tr>
<tr>
<td>JRNL I/O</td>
<td>20</td>
<td>.12</td>
<td></td>
</tr>
<tr>
<td>JRNL BUFFER</td>
<td>0</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>DBKEY</td>
<td>0</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>LOG I/O</td>
<td>492</td>
<td>8.81</td>
<td>XXXXXX</td>
</tr>
<tr>
<td>LOG SNGL THRD</td>
<td>3</td>
<td>.01</td>
<td></td>
</tr>
<tr>
<td>LOG FULL</td>
<td>0</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>SCR I/O</td>
<td>0</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>SCR SNGL THRD</td>
<td>0</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>QUEUE I/O</td>
<td>1,000</td>
<td>4.48</td>
<td>XXX</td>
</tr>
<tr>
<td>STORAGE</td>
<td>0</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>PROGRAM POOL</td>
<td>0</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>PROGRAM LOADS</td>
<td>0</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>LOAD SNGL THRD</td>
<td>0</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>AREA</td>
<td>0</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>ERUS</td>
<td>0</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>DDS</td>
<td>0</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>CHECK USER</td>
<td>0</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>TP LINE I/O</td>
<td>362</td>
<td>189.51</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>MAX TASK</td>
<td>0</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>DBGROUP</td>
<td>821</td>
<td>25.91</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>SHARED CACHE</td>
<td>312</td>
<td>2.23</td>
<td>XX</td>
</tr>
<tr>
<td>OTHER EXTERNAL</td>
<td>1,024</td>
<td>10.09</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>OTHER INTERNAL</td>
<td>0</td>
<td>.00</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>4,157</td>
<td>164.81</td>
<td></td>
</tr>
<tr>
<td>EXCLUDED TOTAL</td>
<td>0</td>
<td>.00</td>
<td></td>
</tr>
</tbody>
</table>

PMIRPT01 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait</td>
<td>Category of wait. Type</td>
</tr>
<tr>
<td>Count of waits for the indicated category.</td>
<td></td>
</tr>
<tr>
<td>Total time spent in waits for the category, across the reported intervals.</td>
<td></td>
</tr>
</tbody>
</table>
Field Description

Wait Time

Scale Graphic representation of total wait time for reported intervals spent for each category of wait. Each X represents a certain number of seconds (as specified by the scaling factor). The symbol ── indicates that the line goes past the right side of the graph.

PMIRPT02 Trend analysis report

PMIRPT02 contains one summary line for each reported interval. Each line shows the total wait count and time and a graphic representation of wait time for the wait type.

Sample report

<table>
<thead>
<tr>
<th>INTERVAL START TIME</th>
<th>WAITS</th>
<th>WAIT TIME (SECS)</th>
<th>SCALED BY TIME (*'X' REPRESENTS 1.0000 SECONDS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>14:58:16</td>
<td>2169</td>
<td>98.36</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>15:00:00</td>
<td>102</td>
<td>10.70</td>
<td>XXXXXXX</td>
</tr>
<tr>
<td>15:10:00</td>
<td>169</td>
<td>6.11</td>
<td>XXXXX</td>
</tr>
<tr>
<td>15:20:00</td>
<td>100</td>
<td>5.02</td>
<td>XXXX</td>
</tr>
<tr>
<td>15:30:00</td>
<td>900</td>
<td>33.63</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>15:40:00</td>
<td>3937</td>
<td>118.03</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>15:50:00</td>
<td>6677</td>
<td>321.10</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>16:00:00</td>
<td>2236</td>
<td>91.54</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>16:10:00</td>
<td>12755</td>
<td>382.87</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>16:20:00</td>
<td>17018</td>
<td>457.57</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>16:30:00</td>
<td>4946</td>
<td>182.55</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>16:40:00</td>
<td>485</td>
<td>36.43</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>16:50:00</td>
<td>333</td>
<td>18.30</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>17:00:00</td>
<td>83</td>
<td>2.83</td>
<td>XX</td>
</tr>
<tr>
<td>17:10:00</td>
<td>92</td>
<td>3.23</td>
<td>XXX</td>
</tr>
<tr>
<td>17:20:00</td>
<td>92</td>
<td>3.57</td>
<td>XXX</td>
</tr>
<tr>
<td>17:30:00</td>
<td>102</td>
<td>8.39</td>
<td>XXXXXXX</td>
</tr>
<tr>
<td>17:40:00</td>
<td>92</td>
<td>3.38</td>
<td>XXX</td>
</tr>
<tr>
<td>17:50:00</td>
<td>172</td>
<td>6.54</td>
<td>XXXX</td>
</tr>
<tr>
<td>18:00:00</td>
<td>89</td>
<td>8.83</td>
<td>XXXXXXX</td>
</tr>
<tr>
<td>18:10:00</td>
<td>4317</td>
<td>78.06</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>18:20:00</td>
<td>85</td>
<td>2.94</td>
<td>XX</td>
</tr>
<tr>
<td>18:30:00</td>
<td>96</td>
<td>6.68</td>
<td>XXXX</td>
</tr>
<tr>
<td>18:40:00</td>
<td>93</td>
<td>3.44</td>
<td>XXX</td>
</tr>
<tr>
<td>18:50:00</td>
<td>483</td>
<td>11.81</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
<tr>
<td>19:00:00</td>
<td>291</td>
<td>18.19</td>
<td>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</td>
</tr>
</tbody>
</table>

PMIRPT02 fields

Field Description

Interv...
PMIRPT04 Summary wait detail report

PMIRPT04 is a summary report for each selected interval. The report shows information on the task activity that occurred during each interval.

**Sample report**

<table>
<thead>
<tr>
<th>REPORT NO. 04</th>
<th>CA, INC.</th>
<th>mm/dd/yy PAGE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA IDMS/PM nn.n volser</td>
<td>SUMMARY WAIT DETAIL REPORT</td>
<td>&lt;----------------- YOUR COMPANY NAME --------------&gt;</td>
</tr>
<tr>
<td>DC SYSTEM VERSION #: 56</td>
<td>DATA FROM: mm/dd/yy</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>START TIME</th>
<th>TASKS STRTD</th>
<th>TASKS ENDED</th>
<th>SYSTEM CPU Waits</th>
<th>USER CPU Waits</th>
<th>DB I/O Waits</th>
<th>DB I/O Wait Time</th>
<th>OTHER I/O Waits</th>
<th>OTHER PGM Waits</th>
<th>MISCELLANEOUS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(SECS)</td>
<td>(SECS)</td>
<td>I/O waits</td>
<td>I/O wait time</td>
<td>I/O waits</td>
<td>I/O wait time</td>
<td>I/O waits</td>
<td>I/O wait time</td>
<td>PGM waits</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>WAIT TIME</td>
<td>WAIT TIME</td>
<td>WAIT TIME</td>
<td>WAIT TIME</td>
<td>WAIT TIME</td>
<td>WAIT TIME</td>
<td>PGM WAIT TIME</td>
</tr>
<tr>
<td>15:30:00</td>
<td>74</td>
<td>75</td>
<td>2.426</td>
<td>756</td>
<td>24.752</td>
<td>129</td>
<td>8.868</td>
<td>15</td>
<td>0.10</td>
</tr>
<tr>
<td>15:40:00</td>
<td>166</td>
<td>164</td>
<td>2.853</td>
<td>3481</td>
<td>99.392</td>
<td>436</td>
<td>16.642</td>
<td>20</td>
<td>2.000</td>
</tr>
<tr>
<td>15:50:00</td>
<td>31</td>
<td>32</td>
<td>5.732</td>
<td>6229</td>
<td>296.651</td>
<td>433</td>
<td>24.249</td>
<td>8</td>
<td>0.983</td>
</tr>
<tr>
<td>16:00:00</td>
<td>2</td>
<td>1</td>
<td>.818</td>
<td>1611</td>
<td>64.828</td>
<td>621</td>
<td>27.502</td>
<td>4</td>
<td>0.11</td>
</tr>
<tr>
<td>16:10:00</td>
<td>22</td>
<td>22</td>
<td>11.519</td>
<td>10989</td>
<td>334.751</td>
<td>1750</td>
<td>46.569</td>
<td>13</td>
<td>0.888</td>
</tr>
<tr>
<td>16:20:00</td>
<td>17</td>
<td>18</td>
<td>2.433</td>
<td>15998</td>
<td>411.194</td>
<td>997</td>
<td>44.953</td>
<td>10</td>
<td>0.083</td>
</tr>
<tr>
<td>16:30:00</td>
<td>49</td>
<td>48</td>
<td>14.282</td>
<td>4608</td>
<td>146.877</td>
<td>281</td>
<td>22.819</td>
<td>10</td>
<td>1.677</td>
</tr>
<tr>
<td>16:40:00</td>
<td>87</td>
<td>89</td>
<td>.403</td>
<td>14</td>
<td>1.277</td>
<td>446</td>
<td>22.132</td>
<td>4</td>
<td>21</td>
</tr>
<tr>
<td>16:50:00</td>
<td>53</td>
<td>53</td>
<td>.261</td>
<td>23</td>
<td>.859</td>
<td>298</td>
<td>11.553</td>
<td>14</td>
<td>2.888</td>
</tr>
</tbody>
</table>

**PMIRPT04 fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>Tasks Strtd</td>
<td>Count of tasks that were initiated during the interval</td>
</tr>
<tr>
<td>Tasks Ended</td>
<td>Count of tasks that terminated during the interval</td>
</tr>
<tr>
<td>System CPU</td>
<td>Total system CPU time used during the interval</td>
</tr>
<tr>
<td>User CPU</td>
<td>Total user CPU time used during the interval</td>
</tr>
<tr>
<td>DB I/O Waits</td>
<td>Count of database I/O waits during the interval</td>
</tr>
<tr>
<td>DB I/O Wait Time</td>
<td>Time spent in database I/O waits during the interval (ssss.ttt)</td>
</tr>
<tr>
<td>Other I/O Waits</td>
<td>Count of additional I/O waits during the interval, including waits for: Journal DDLDCLOG DDLDCRUN DDLDCMSG Program-load reads</td>
</tr>
<tr>
<td>Other I/O Wait Time</td>
<td>Time spent in additional I/O waits during the interval (ssss.ttt)</td>
</tr>
<tr>
<td>Other Pgm Waits</td>
<td>Count of additional program waits during the interval</td>
</tr>
<tr>
<td>Other Pgm Wait Time</td>
<td>Time (ssss.ttt) spent in additional program waits during the interval, including waits for: Db-keys Buffers Journal buffers Program pool Storage pool TPIO Area shared/protected/exclusive DBGroup</td>
</tr>
</tbody>
</table>
Misc System Waits | Miscellaneous system waits that occurred during the interval
---|---
Misc System Wait Time | Time spent in miscellaneous waits during the interval (ssss.ttt), including waits for:
External request units
Check user waits
Log single threading and log full conditions
Scratch single threading
Loader single threading
DDS
New task conditions
Unidentified external and internal waits

PMIRPT05 DBkey/Area detail report

PMIRPT05 contains detailed area-access information for each reported interval. The report shows one line for each area accessed during each interval that shows information on db-key and area waits.

Sample report

<table>
<thead>
<tr>
<th>START TIME</th>
<th>AREA NAME</th>
<th>DBKEY WAIT</th>
<th>AREA NAME</th>
<th>DBKEY WAIT</th>
<th>AREA NAME</th>
<th>DBKEY WAIT</th>
<th>AREA NAME</th>
<th>DBKEY WAIT</th>
<th>AREA NAME</th>
<th>DBKEY WAIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>16:20:00</td>
<td>CA30NWK.DDLDCLOD</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CA30NWK.DDLDML</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CFAXNWK.DDLDCLOD</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CFAXNWK.DDLDML</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CG30NWK.DDLDCLOD</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CG30NWK.DDLDML</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CKSXNWK.DDLDCLOD</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CKSXNWK.DDLDML</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CSADNWK.DDLDCLOD</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CSADNWK.DDLDML</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLDCLOD</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLDCLOG</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLDCRUN</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLDCSCR</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLDCML</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td>16:30:00</td>
<td>CA30NWK.DDLDCLOD</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CA30NWK.DDLDML</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CFAXNWK.DDLDCLOD</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CFAXNWK.DDLDML</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CG30NWK.DDLDCLOD</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CG30NWK.DDLDML</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CKSXNWK.DDLDCLOD</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CKSXNWK.DDLDML</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CSADNWK.DDLDCLOD</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CSADNWK.DDLDML</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLDCLOD</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLDCLOG</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLDCRUN</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLDCSCR</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLDCML</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
<td>.0000</td>
<td>0</td>
</tr>
</tbody>
</table>

PMIRPT05 fields
PMIRPT09 Shared cache summary report

PMIRPT09 contains information on the use of the Shared Cache in the Coupling Facility. The report shows all the shared cache that were active in the corresponding intervals, and for each shared cache, all the files that were assigned to it. Files that were excluded by input selection parameters do not appear on the report.

Sample report
### PMIRPT09 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>Shared Cache Name</td>
<td>Name of the shared cache</td>
</tr>
<tr>
<td>File Name</td>
<td>Name of the file that is assigned to the corresponding shared cache</td>
</tr>
<tr>
<td>Number of Reads</td>
<td>Number of read requests from a specific file in the shared cache</td>
</tr>
<tr>
<td>Found in Cache</td>
<td>Number of times a database page we want to read was already present and valid in the shared cache</td>
</tr>
<tr>
<td>Number of Writes</td>
<td>Number of write requests to a specific file in the shared cache</td>
</tr>
<tr>
<td>Sh-Cache Waits</td>
<td>Number of waits for a specific file in the shared cache</td>
</tr>
<tr>
<td>Sh-Cache Wait Time</td>
<td>Amount of time spent waiting for a specific file in the shared cache</td>
</tr>
<tr>
<td>Avg Sh-Cache Wait Time</td>
<td>Average wait time for a specific file in the shared cache</td>
</tr>
</tbody>
</table>

### PMIRPT10 DBGroup summary report

PMIRPT10 contains information on the use of the dynamic routing of database sessions. The report shows all the DBGroups to which database sessions have been dynamically routed for processing.

**Sample report**

<table>
<thead>
<tr>
<th>START TIME</th>
<th>DBGroup Name</th>
<th>NUMBER OF REQUESTS</th>
<th>DBGroup WAITS</th>
<th>DBGroup Wait Time (SECS)</th>
<th>AVG DBGroup Wait Time (SECS)</th>
<th>SERVER Node</th>
<th># Requests Processed</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:00:00</td>
<td>DBDCGR</td>
<td>1</td>
<td>1</td>
<td>.002</td>
<td>.0022</td>
<td>SYSTEM71</td>
<td>1</td>
</tr>
<tr>
<td>8:10:00</td>
<td>DBDCGR</td>
<td>1019</td>
<td>820</td>
<td>25.904</td>
<td>.0316</td>
<td>SYSTEM71</td>
<td>472</td>
</tr>
<tr>
<td>8:12:00</td>
<td>DBDCGR</td>
<td>69</td>
<td>6</td>
<td>2.021</td>
<td>.077</td>
<td>SYSTEM71</td>
<td>472</td>
</tr>
<tr>
<td>8:15:00</td>
<td>DBDCGR</td>
<td>1</td>
<td>1</td>
<td>.002</td>
<td>.0022</td>
<td>SYSTEM71</td>
<td>1</td>
</tr>
<tr>
<td>8:18:00</td>
<td>DBDCGR</td>
<td>1</td>
<td>2</td>
<td>.002</td>
<td>.0022</td>
<td>SYSTEM71</td>
<td>1</td>
</tr>
</tbody>
</table>

**PMIRPT10 fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>DBGroup Name</td>
<td>Name of the DBGroup</td>
</tr>
<tr>
<td>Number of Requests</td>
<td>Total number of requests that have been submitted to the DBGroup</td>
</tr>
<tr>
<td>DBGroup Waits</td>
<td>Total number of waits for the DBGroup</td>
</tr>
</tbody>
</table>
**PMIRPT11 I/O by area summary report**

PMIRPT11 contains detailed I/O information for each reported interval. The report shows one line for each area accessed during each interval that shows information on I/O requests for the area.

### Sample report

<table>
<thead>
<tr>
<th>START TIME</th>
<th>AREA NAME</th>
<th>READ I/O</th>
<th>READ WAIT TIME (SECS)</th>
<th>AVG READ WAIT TIME (SECS)</th>
<th>WRITE I/O</th>
<th>WRITE WAIT TIME (SECS)</th>
<th>AVG WRITE WAIT TIME (SECS)</th>
<th>BUFFER Reads</th>
<th>BUFFER WAITS</th>
<th>BUFFER WAIT TIME (SECS)</th>
<th>AVG BUFFER WAIT TIME (SECS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16:20:00</td>
<td>CA30NWK.DDLCLCLOD</td>
<td>6</td>
<td>345.800</td>
<td>.0236</td>
<td>1</td>
<td>9.000</td>
<td>.1500</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CA30NWK.DDLCLCLOD</td>
<td>14627</td>
<td>145.800</td>
<td>.0318</td>
<td>28</td>
<td>9.000</td>
<td>.1429</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CFAXNWK.DDLCLCLOD</td>
<td>777</td>
<td>33.800</td>
<td>.0425</td>
<td>16</td>
<td>9.000</td>
<td>.1500</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CG30NWK.DDLCLCLOD</td>
<td>147</td>
<td>7.800</td>
<td>.0476</td>
<td>12</td>
<td>9.000</td>
<td>.1500</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CKSXNWK.DDLCLCLOD</td>
<td>427</td>
<td>25.800</td>
<td>.0585</td>
<td>13</td>
<td>9.000</td>
<td>.1500</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CSADNWK.DDLCLCLOD</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLCLCLOD</td>
<td>4</td>
<td>9.000</td>
<td>12</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLCLCLOD</td>
<td>60</td>
<td>9.000</td>
<td>15</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLCLCLOD</td>
<td>936</td>
<td>35.000</td>
<td>.0374</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLCLCLOD</td>
<td>22</td>
<td>1</td>
<td>22</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>16:30:00</td>
<td>CA30NWK.DDLCLCLOD</td>
<td>5</td>
<td>145.800</td>
<td>.0318</td>
<td>28</td>
<td>9.000</td>
<td>.1429</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CA30NWK.DDLCLCLOD</td>
<td>4565</td>
<td>145.800</td>
<td>.0318</td>
<td>28</td>
<td>9.000</td>
<td>.1429</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CFAXNWK.DDLCLCLOD</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>CG30NWK.DDLCLCLOD</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>CKSXNWK.DDLCLCLOD</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>CSADNWK.DDLCLCLOD</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLCLCLOD</td>
<td>6</td>
<td>9.000</td>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLCLCLOD</td>
<td>70</td>
<td>18.000</td>
<td>.1429</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLCLCLOD</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLCLCLOD</td>
<td>199</td>
<td>13.000</td>
<td>.0653</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLCLCLOD</td>
<td>17</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### PMIRPT11 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area Name</td>
<td>Name of the DC/UCF area</td>
</tr>
<tr>
<td>Read I/O Waits</td>
<td>Count of physical read I/Os that resulted in a wait</td>
</tr>
<tr>
<td>Read Wait Time</td>
<td>Total time spent waiting for physical read I/Os (ss.ttt)</td>
</tr>
<tr>
<td>Avg Read Wait Time</td>
<td>Average amount of time spent waiting for physical read I/Os against the area (ss.ttt)</td>
</tr>
<tr>
<td>Write I/O Waits</td>
<td>Count of physical write I/Os that resulted in a wait</td>
</tr>
<tr>
<td>Write Wait Time</td>
<td>Total time spent waiting for physical write I/Os (ss.ttt)</td>
</tr>
<tr>
<td>Avg Write Wait Time</td>
<td>Average amount of time spent waiting for physical write I/Os against the area (ss.ttt)</td>
</tr>
<tr>
<td>Buffer Hits</td>
<td>Count of requests that could be processed within the buffer, without a physical I/O</td>
</tr>
<tr>
<td>Buffer Waits</td>
<td>Count of waits for buffer requests; that is, the number of times a buffer was requested for the area but not available</td>
</tr>
<tr>
<td>Buffer Wait Time</td>
<td>Total time spent on buffer waits (ss.ttt)</td>
</tr>
<tr>
<td>Avg Bufr Wait Time</td>
<td>Average amount of time spent waiting for a buffer (ss.ttt)</td>
</tr>
<tr>
<td>Buffer Name</td>
<td>Buffer name for the area</td>
</tr>
</tbody>
</table>

### PMIRPT12 I/O by file summary report

PMIRPT12 contains detailed I/O information for each reported interval. The report shows one line for each file accessed during each interval that shows information on I/O requests for the file.

### Sample report

<table>
<thead>
<tr>
<th>START TIME</th>
<th>FILE NAME</th>
<th>READ I/O WAITS</th>
<th>READ WAIT TIME (SECS)</th>
<th>AVG READ WRITE I/O WAITS</th>
<th>WRITE WAIT TIME (SECS)</th>
<th>AVG WRITE BUFFER WAITS</th>
<th>BUFFER WAIT TIME (SECS)</th>
<th>AVG BUFR WAIT TIME (SECS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16:20:00</td>
<td>CA30NWK.CA30DML1</td>
<td>7355</td>
<td>173.000</td>
<td>.0235</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CA30NWK.CA30DML2</td>
<td>7272</td>
<td>172.000</td>
<td>.0237</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CA30NWK.CA30LOD</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CFAAXNWK.CFADML</td>
<td>777</td>
<td>33.000</td>
<td>.0425</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CG30NWK.CG30DML</td>
<td>147</td>
<td>7.000</td>
<td>.0476</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CG30NWK.CG30LOD</td>
<td>427</td>
<td>25.000</td>
<td>.0585</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DCDML</td>
<td>936</td>
<td>35.000</td>
<td>.0374</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>START TIME</th>
<th>FILE NAME</th>
<th>READ I/O WAITS</th>
<th>READ WAIT TIME (SECS)</th>
<th>AVG READ WRITE I/O WAITS</th>
<th>WRITE WAIT TIME (SECS)</th>
<th>AVG WRITE BUFFER WAITS</th>
<th>BUFFER WAIT TIME (SECS)</th>
<th>AVG BUFR WAIT TIME (SECS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16:30:00</td>
<td>SYSTEM.DCLOG</td>
<td>60</td>
<td>9.000</td>
<td>.1560</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DCSCR</td>
<td>936</td>
<td>35.000</td>
<td>.0374</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CA IDMS - 19.0

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>File Name</td>
<td>Name of the file</td>
</tr>
<tr>
<td>Read I/O Waits</td>
<td>Count of physical read I/Os that resulted in a wait</td>
</tr>
<tr>
<td>Read Wait Time</td>
<td>Total time spent waiting for physical read I/Os (ss.ttt)</td>
</tr>
<tr>
<td>Avg Read Wait Time</td>
<td>Average amount of time spent waiting for physical read I/Os against the file (ss.tttt)</td>
</tr>
<tr>
<td>Write I/O Waits</td>
<td>Count of physical write I/Os that resulted in a wait</td>
</tr>
<tr>
<td>Write Wait Time</td>
<td>Total time spent waiting for physical write I/Os (ss.ttt)</td>
</tr>
<tr>
<td>Avg Write Wait Time</td>
<td>Average amount of time spent waiting for physical write I/Os against the file (ss.tttt)</td>
</tr>
<tr>
<td>Buffer Hits</td>
<td>Number of times a request was filled by a page already in the buffer</td>
</tr>
<tr>
<td>Buffer Waits</td>
<td>Number of times a task had to wait because all the pages in the buffer pool were in use by other tasks</td>
</tr>
<tr>
<td>Buffer Wait Time</td>
<td>Total time spent waiting for a buffer to become available (ss.ttt)</td>
</tr>
<tr>
<td>Avg Bufr Wait Time</td>
<td>Average amount of time spent waiting for a buffer (ss.tttt)</td>
</tr>
<tr>
<td>Buffer Name</td>
<td>Name of the buffer with which the file is associated</td>
</tr>
</tbody>
</table>

Pميرت13 Buffer summary report

Pميرت13 contains information related to database and journal buffer use for each reported interval. The report shows one line of information for each buffer accessed.

Sample report
### PMIRPT13 fields

<table>
<thead>
<tr>
<th>Field Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss).</td>
</tr>
<tr>
<td>Buffer Name</td>
<td>Name of the DC/UCF buffer pool.</td>
</tr>
<tr>
<td>Buff Req</td>
<td>Total number of database requests made against the buffer.</td>
</tr>
<tr>
<td>Flshs</td>
<td>Count of times a page had to be written to disk because another transaction required it.</td>
</tr>
<tr>
<td>Hits</td>
<td>Count of database requests that could be processed in the buffer without a physical I/O.</td>
</tr>
<tr>
<td>Hit Ratio</td>
<td>Ratio of the number of database requests that could be processed in the buffer without a physical I/O (hits) to the total number of buffer requests. For example, a hit ratio of 1.00 indicates that all database pages requested were available in the buffer. A hit ratio of 0.00 indicates that none of the database pages requested was available in the buffer.</td>
</tr>
<tr>
<td>Disk I/O</td>
<td>Count of requests that could not be processed in the buffer, and therefore required a physical I/O.</td>
</tr>
<tr>
<td>I/O Wait</td>
<td>Time spent waiting for I/O to complete requests that could not be processed in the buffer (ss.ttt).</td>
</tr>
</tbody>
</table>

### Table

<table>
<thead>
<tr>
<th>TIME</th>
<th>NAME</th>
<th>RQSTS</th>
<th>FLSHS</th>
<th>HITS (%)</th>
<th>I/O (SECS)</th>
<th>(SECS)</th>
<th>WAITS (SECS)</th>
<th>(SECS)</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>16:30:00</td>
<td>CA30DB-BUFFER</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CSA30DB-BUFFER</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DCML-BUFFER</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>978560</td>
<td>7517.000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DCLOD-BUFFER</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>687616</td>
<td>16884.000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DCMS- BUFFER</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>417280</td>
<td>2857.000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DCRUN-BUFFER</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>961664</td>
<td>58.000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DCSEC-BUFFER</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>95680</td>
<td>3343.000</td>
<td>0.001</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DEFAULT-BUFFER</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>720000</td>
<td>65872.000</td>
<td>0.001</td>
<td>0</td>
</tr>
<tr>
<td>16:40:00</td>
<td>CA30DB-BUFFER</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CSA30DB-BUFFER</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DCML-BUFFER</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DCLOD-BUFFER</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>829184</td>
<td>23481.000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DCMS- BUFFER</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DCRUN-BUFFER</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>65536</td>
<td>216784</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DCSEC-BUFFER</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DEFAULT-BUFFER</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16:50:00</td>
<td>CA30DB-BUFFER</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>CSA30DB-BUFFER</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DCML-BUFFER</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>988224</td>
<td>4567.000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DCLOD-BUFFER</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>783872</td>
<td>12301.000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DCMS- BUFFER</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>20.000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DCRUN-BUFFER</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DCSEC-BUFFER</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>26.000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>DEFAULT-BUFFER</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Field Description

Aver Average amount of time spent waiting for I/O to complete requests that could not be processed age I in the buffer (ss.tttt).

/O Tim e

Buff Count of waits for buffer requests; that is, the number of times the buffer was requested but er not available.
Wait t

Buff Total time spent on buffer waits (ss.tttt).
 er Wait Tim e

Aver Average amount of time spent on buffer waits (ss.tttt).
age Wait Tim e

Bufr Size of the largest page maintained in the buffer pool, in bytes.
Page Size

Buf Number of times a database page was read from disk, not from the buffer.
Read

Buf Number of times a buffer page was discarded from the journal buffers in order to read another Writ e

PMIRPT14 CDMSLIB summary report

PMIRPT14 contains CDMSLIB information for each reported interval.

Sample report

<table>
<thead>
<tr>
<th>REPORT NO. 14</th>
<th>CA, INC.</th>
<th>DC SYSTEM VERSION #: 56</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPORTS/PMM n.n volser</td>
<td>CDMSLIB SUMMARY REPORT</td>
<td>YOUR COMPANY NAME</td>
</tr>
<tr>
<td>DATA FROM: mm/dd/yy</td>
<td>PAGE 1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>START TIME</th>
<th>COMS LIBRARY</th>
<th>PROGRAM LOAD</th>
<th>LOAD WAITS</th>
<th>AVG LOAD</th>
<th>WAIT TIME</th>
<th>AVG LOAD TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>14:58:16</td>
<td>CDMSLIB</td>
<td>27</td>
<td>1.128</td>
<td>.0418</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:00:00</td>
<td>CDMSLIB</td>
<td>1</td>
<td>.100</td>
<td>.1001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:10:00</td>
<td>CDMSLIB</td>
<td>14</td>
<td>.618</td>
<td>.0441</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:20:00</td>
<td>CDMSLIB</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:30:00</td>
<td>CDMSLIB</td>
<td>12</td>
<td>.418</td>
<td>.0349</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:40:00</td>
<td>CDMSLIB</td>
<td>42</td>
<td>2.422</td>
<td>.0481</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:50:00</td>
<td>CDMSLIB</td>
<td>27</td>
<td>1.507</td>
<td>.0558</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:00:00</td>
<td>CDMSLIB</td>
<td>2</td>
<td>.084</td>
<td>.0419</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:10:00</td>
<td>CDMSLIB</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:20:00</td>
<td>CDMSLIB</td>
<td>1</td>
<td>.066</td>
<td>.0656</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:30:00</td>
<td>CDMSLIB</td>
<td>10</td>
<td>.520</td>
<td>.0520</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:40:00</td>
<td>CDMSLIB</td>
<td>3</td>
<td>.129</td>
<td>.0431</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:50:00</td>
<td>CDMSLIB</td>
<td>11</td>
<td>.489</td>
<td>.0444</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PMIRPT14 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>CDMS Library Name</td>
<td>Name of the load library specified by ddname CDMSLIB</td>
</tr>
<tr>
<td>Program Load Waits</td>
<td>Total number of program load waits during the interval</td>
</tr>
<tr>
<td>Load Wait Time</td>
<td>Total time spent on program load waits during the interval (ss.ttt).</td>
</tr>
<tr>
<td>Avg Load Wait Time</td>
<td>Average amount of time for each program load wait during the interval (ss.ttt)</td>
</tr>
</tbody>
</table>

PMIRPT15 Journal summary report

PMIRPT15 contains detailed journal-access information for each reported interval. The report shows one line for each journal file accessed during each interval that shows information on access requests.

Sample report

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>8:30:00</td>
<td>J1JRNL</td>
<td>540</td>
<td>14.719</td>
<td>.0273</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></td>
<td>J2JRNL</td>
<td>540</td>
<td>9.593</td>
<td>.0178</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></td>
<td>J3JRNL</td>
<td>4329</td>
<td>15.297</td>
<td>.0283</td>
<td>1302</td>
<td>39.673</td>
<td>.0305</td>
<td>1165</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8:45:00</td>
<td>J1JRNL</td>
<td>537</td>
<td>12.358</td>
<td>.0230</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></td>
<td>J2JRNL</td>
<td>537</td>
<td>9.490</td>
<td>.0177</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></td>
<td>J3JRNL</td>
<td>4635</td>
<td>15.605</td>
<td>.0291</td>
<td>1165</td>
<td>41.027</td>
<td>.0352</td>
<td>7856</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PMIRPT15 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>Journal Name</td>
<td>Name of the journal file, as defined in the DMCL</td>
</tr>
<tr>
<td>Blocks Written</td>
<td>Number of blocks written to the journal file during the interval</td>
</tr>
<tr>
<td>Read Waits</td>
<td>Count of physical read (rollback) I/Os against the journal; all physical read I/Os result in a wait</td>
</tr>
<tr>
<td>Read Wait Time</td>
<td>Total time spent waiting for physical read I/Os against the journal (ss.ttt)</td>
</tr>
<tr>
<td>Avg Read Wait Time</td>
<td>Average amount of time spent waiting for physical read I/Os against the journal (ss .tttt)</td>
</tr>
<tr>
<td>Write Waits</td>
<td>Count of physical write I/Os against the journal; all physical write I/Os result in a wait</td>
</tr>
<tr>
<td>Write Wait Time</td>
<td>Total time spent waiting for physical write I/Os against the journal (ss.ttt)</td>
</tr>
</tbody>
</table>
Avg Write Wait Time  Average amount of time spent waiting for physical write I/Os against the journal (s tttt)

Journal Buffer Waits  Number of time the task had to wait because all the journal buffers were in use by other tasks

Jrnl Bufr Wait Time  Total time spent waiting for a journal buffer (ss.tttt)

Avg Bufr Wait Time  Average amount of time spent waiting for a journal buffer (ss.tttt)

Begin Jrnl RBN  Relative block number of the first block written to the journal during the interval

End Jrnl RBN  Relative block number of the last block written to the journal during the interval

PMIRPT16 TP line summary report

PMIRPT16 contains information on teleprocessing line usage and waits for each interval.

Sample report

<table>
<thead>
<tr>
<th>START TIME</th>
<th>LINE NAME</th>
<th>NUM TRMN</th>
<th>TRMNL READS</th>
<th>TRMNL WRITES</th>
<th>READ ERRS</th>
<th>WRITE ERRs</th>
<th>TRMNL I/O WAIT TIME (SECS)</th>
<th>AVG TRMNL I/O TIME (SECS)</th>
<th>NUM RPLS</th>
<th>NUM RPL WAITS</th>
<th>RPL MAINTIME (SECS)</th>
<th>AVG RPL MAINTIME (SECS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16:00:00</td>
<td>CCI56</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CONSOLE</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></td>
<td>JESRDR</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></td>
<td>PRINT56</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UCF56</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>16:10:00</td>
<td>VTAM56</td>
<td>50</td>
<td>2</td>
<td>121</td>
<td>4</td>
<td>.011</td>
<td>.0028</td>
<td>10</td>
<td>1820</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CONSOLE</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></td>
<td>JESRDR</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></td>
<td>PRINT56</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UCF56</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>16:20:00</td>
<td>VTAM56</td>
<td>50</td>
<td>22</td>
<td>137</td>
<td>13</td>
<td>.088</td>
<td>.0067</td>
<td>10</td>
<td>2098</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CONSOLE</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></td>
<td>JESRDR</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></td>
<td>PRINT56</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UCF56</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>16:30:00</td>
<td>VTAM56</td>
<td>50</td>
<td>19</td>
<td>119</td>
<td>10</td>
<td>.003</td>
<td>.0083</td>
<td>10</td>
<td>2333</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CONSOLE</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></td>
<td>JESRDR</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></td>
<td>PRINT56</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UCF56</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>16:40:00</td>
<td>VTAM56</td>
<td>50</td>
<td>45</td>
<td>89</td>
<td>10</td>
<td>1.677</td>
<td>.1677</td>
<td>10</td>
<td>2515</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CONSOLE</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></td>
<td>JESRDR</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></td>
<td>PRINT56</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UCF56</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>16:50:00</td>
<td>VTAM56</td>
<td>50</td>
<td>87</td>
<td>93</td>
<td>4</td>
<td>.0001</td>
<td>.0001</td>
<td>10</td>
<td>2699</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CONSOLE</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></td>
<td>JESRDR</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></td>
<td>PRINT56</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UCF56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PMIRPT16 fields

PMIRPT16 contains information on teleprocessing line usage and waits for each interval.
PMIRPT17 Program pool summary report

PMIRPT17 contains information on the use of program pools for each reported interval. The report shows one line of information for each program pool used during the interval.

### Sample report

<table>
<thead>
<tr>
<th>START TIME</th>
<th>POOL TYPE</th>
<th>POOL SIZE (K)</th>
<th>IN USE (K)</th>
<th>HIGH WATER (K)</th>
<th>SPACE LOADED (K)</th>
<th>PGM LOADS</th>
<th>INTO UNALLOC SPACE</th>
<th>OVERLAY PGM</th>
<th>OVERLAY IN USE</th>
<th>POOL WAYS</th>
<th>POOL LOAD WAITS</th>
<th>POOLLOAD WAIT TIME (SECS)</th>
<th>AVG LOAD WAITS</th>
<th>AVG LOAD TIME (SECS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>14:58:16</td>
<td>XA REENT</td>
<td>3788</td>
<td>1562</td>
<td>1562</td>
<td>1562</td>
<td>113</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:00:00</td>
<td>REENT</td>
<td>1364</td>
<td>394</td>
<td>394</td>
<td>394</td>
<td>32</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:10:00</td>
<td>PROGRAM</td>
<td>508</td>
<td>136</td>
<td>136</td>
<td>136</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:20:00</td>
<td>XA REENT</td>
<td>3788</td>
<td>1583</td>
<td>1583</td>
<td>1583</td>
<td>21</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:30:00</td>
<td>REENT</td>
<td>1364</td>
<td>394</td>
<td>394</td>
<td>394</td>
<td>32</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Field Description

- **Start Time**: Starting time for the interval on a 24-hour clock (hh:mm:ss)
- **Line Name**: Name of line, as defined with system generation LINE statement
- **Num Trms**: Number of terminals on the line, as defined with system generation LTERM and PTERM statements
- **Trmnl Reads**: Number of terminal reads that occurred during the interval
- **Trmnl Writes**: Number of terminal writes that occurred during the interval
- **Read Errs**: Number of read errors that occurred during the interval
- **Write Errs**: Number of write errors that occurred during the interval
- **Trmnl I/O Waits**: Number of waits for terminal I/O during the interval
- **Trmnl I/O Wait Time**: Number of seconds waiting for terminal I/O during the interval (ss.tttt).
- **Avg Trmnl I/O Time**: Average length of a wait for terminal I/O during the interval (ss.tttt).
- **Num RPLs Sgend**: Number of request parameter lists (RPLs) specified with the system generation LINE statement RPL COUNT parameter
- **Num RPL Rqsts**: Number of RPL requests during the interval
- **RPL Waits**: Number of waits for an RPL during the interval
- **RPL Wait Time**: Number of seconds spent waiting for an RPL during the interval (ss.tttt).
- **Avg RPL Wait Time**: Average number of seconds for an RPL wait during the interval (ss.tttt).
### PMIRPT17 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>Pool Type</td>
<td>Type of program pool</td>
</tr>
<tr>
<td>Pool Size</td>
<td>Size of the program pool, specified in kilobytes</td>
</tr>
<tr>
<td>In Use</td>
<td>Amount of the program pool in use at the end of the interval, specified in kilobytes</td>
</tr>
<tr>
<td>High Water</td>
<td>Highest amount of the program pool in use at any point in time during the interval, specified in kilobytes</td>
</tr>
<tr>
<td>Space Loaded</td>
<td>Amount of program pool space loaded from disk during the interval</td>
</tr>
<tr>
<td>Pgm Pool Loads</td>
<td>Count of programs loaded into the pool during the interval</td>
</tr>
<tr>
<td>Into Unalloc Space</td>
<td>Count of loads into unallocated space</td>
</tr>
<tr>
<td>Ovrlay Unused Pgm</td>
<td>Count of loads overlaying a program not currently in use</td>
</tr>
<tr>
<td>Ovrlay Pgm In Use</td>
<td>Count of loads overlaying a program currently in use</td>
</tr>
<tr>
<td>Pool Waits</td>
<td>Number of times an active task had to wait for space in a pool</td>
</tr>
<tr>
<td>Load Waits</td>
<td>Number of times the system had to wait to load a program once storage was available in the pool; usually caused by I/O to the load library or load area</td>
</tr>
<tr>
<td>Pgmload Wait Time</td>
<td>Time spent waiting to load programs</td>
</tr>
<tr>
<td>Average Load Wait Time</td>
<td>Average amount of time spent waiting to load programs</td>
</tr>
</tbody>
</table>
PMIRPT18 Storage pool summary report

PMIRPT18 contains information on storage pool activity for each reported interval. The report shows one line of information for each storage pool accessed during each interval.

Sample report

<table>
<thead>
<tr>
<th>START TIME</th>
<th>POOL NUMBER</th>
<th>POOL SIZE (K)</th>
<th>IN USE (K)</th>
<th>HIGH WATER (K)</th>
<th>STG CUSHION (K)</th>
<th>TIMES SOS</th>
<th>STORAGE GETS</th>
<th>STORAGE FREES</th>
<th>PASS 1</th>
<th>PASS 2</th>
<th>PASS 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>14:58:16</td>
<td>0</td>
<td>1016</td>
<td>112</td>
<td>116</td>
<td>100</td>
<td>325</td>
<td>260</td>
<td>219</td>
<td>106</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:00:00</td>
<td>0</td>
<td>1016</td>
<td>112</td>
<td>116</td>
<td>100</td>
<td>351</td>
<td>284</td>
<td>246</td>
<td>105</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:10:00</td>
<td>0</td>
<td>1016</td>
<td>112</td>
<td>116</td>
<td>100</td>
<td>121</td>
<td>119</td>
<td>121</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:20:00</td>
<td>0</td>
<td>1016</td>
<td>112</td>
<td>116</td>
<td>100</td>
<td>120</td>
<td>120</td>
<td>120</td>
<td>128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:30:00</td>
<td>0</td>
<td>1016</td>
<td>112</td>
<td>116</td>
<td>100</td>
<td>378</td>
<td>319</td>
<td>151</td>
<td>227</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:40:00</td>
<td>0</td>
<td>1016</td>
<td>112</td>
<td>116</td>
<td>100</td>
<td>351</td>
<td>284</td>
<td>246</td>
<td>105</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:50:00</td>
<td>0</td>
<td>1016</td>
<td>112</td>
<td>116</td>
<td>100</td>
<td>1892</td>
<td>1884</td>
<td>522</td>
<td>1370</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:00:00</td>
<td>0</td>
<td>1016</td>
<td>112</td>
<td>116</td>
<td>100</td>
<td>120</td>
<td>120</td>
<td>120</td>
<td>128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:10:00</td>
<td>0</td>
<td>1016</td>
<td>112</td>
<td>116</td>
<td>100</td>
<td>378</td>
<td>319</td>
<td>151</td>
<td>227</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:20:00</td>
<td>0</td>
<td>1016</td>
<td>112</td>
<td>116</td>
<td>100</td>
<td>351</td>
<td>284</td>
<td>246</td>
<td>105</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:30:00</td>
<td>0</td>
<td>1016</td>
<td>112</td>
<td>116</td>
<td>100</td>
<td>120</td>
<td>120</td>
<td>120</td>
<td>128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:40:00</td>
<td>0</td>
<td>1016</td>
<td>112</td>
<td>116</td>
<td>100</td>
<td>378</td>
<td>319</td>
<td>151</td>
<td>227</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PMIRPT18 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>Pool Number</td>
<td>Number that identifies the storage pool, as assigned at system generation</td>
</tr>
<tr>
<td>Pool Size</td>
<td>Size of the storage pool, specified in kilobytes</td>
</tr>
<tr>
<td>In Use</td>
<td>Amount of the storage pool in use at the end of the interval, specified in kilobytes</td>
</tr>
<tr>
<td>High Water</td>
<td>The most storage used in that pool during the interval, specified in kilobytes</td>
</tr>
<tr>
<td>Storage Cushion</td>
<td>Size of the storage cushion, specified in kilobytes</td>
</tr>
<tr>
<td>Times SOS</td>
<td>Number of times the short-on-storage condition occurred during the interval</td>
</tr>
</tbody>
</table>
### PMIRPT19 Storage waits summary report

PMIRPT19 contains information on storage type waits for each reported interval. The report shows 1 column of information for each storage type for each interval.

#### Sample report

<table>
<thead>
<tr>
<th>START TIME</th>
<th>STGLOC</th>
<th>NON-XA</th>
<th>XA</th>
<th>SHRD WAIT</th>
<th>AVG SHRD</th>
<th>SKEPT WAIT</th>
<th>AVG SKEPT</th>
<th>USER WAIT</th>
<th>AVG USER</th>
<th>UKEPT WAIT</th>
<th>AVG UKEPT</th>
<th>OTHER WAIT</th>
<th>AVG OTHER</th>
<th>AVG Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>14:58:16</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15:00:00</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15:10:00</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15:20:00</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15:30:00</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15:40:00</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15:50:00</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16:00:00</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16:10:00</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16:20:00</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16:30:00</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16:40:00</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### PMIRPT19 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage Gets</td>
<td>Count of get-storage (#GETSTG) requests issued against the pool during the interval</td>
</tr>
<tr>
<td>Storage Frees</td>
<td>Count of free-storage (#FREESTG) requests issued against the pool during the interval</td>
</tr>
<tr>
<td>Stg Pass 1</td>
<td>Number of times the space requested by a #GETSTG command was allocated using Scan 1</td>
</tr>
<tr>
<td>Stg Pass 2</td>
<td>Number of times the space requested by a #GETSTG command was allocated using Scan 2</td>
</tr>
<tr>
<td>Stg Pass 3</td>
<td>This field is no longer used</td>
</tr>
</tbody>
</table>
PMIRPT21 I/O by area detail report

PMIRPT21 contains detailed information on an area's input/output during a specific interval.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>Stgloc</td>
<td>Whether the storage resides above the line (XA) or below the line (NON-XA); there should be very few waits for XA storage</td>
</tr>
<tr>
<td>Shrd Stg Waits</td>
<td>Number of waits to acquire shared storage during the interval</td>
</tr>
<tr>
<td>Shrd Wait Time</td>
<td>Amount of time spent waiting to acquire shared storage during the interval (ss.ttt).</td>
</tr>
<tr>
<td>Avg Shrd Time</td>
<td>Average length of a wait to acquire shared storage during the interval (ss.tttt).</td>
</tr>
<tr>
<td>Shrd Kept Waits</td>
<td>Number of waits to acquire shared kept storage during the interval</td>
</tr>
<tr>
<td>Skept Wait Time</td>
<td>Amount of time spent waiting to acquire shared kept storage during the interval (ss.ttt).</td>
</tr>
<tr>
<td>Avg Skept Time</td>
<td>Average length of a wait to acquire shared kept storage during the interval (ss.tttt).</td>
</tr>
<tr>
<td>User Stg Waits</td>
<td>Number of waits to acquire user storage during the interval</td>
</tr>
<tr>
<td>User Wait Time</td>
<td>Amount of time spent waiting to acquire user storage during the interval (ss.ttt).</td>
</tr>
<tr>
<td>Avg User Time</td>
<td>Average length of a wait to acquire user storage during the interval (ss.tttt).</td>
</tr>
<tr>
<td>User Kept Waits</td>
<td>Number of waits to acquire user kept storage during the interval</td>
</tr>
<tr>
<td>Ukept Wait Time</td>
<td>Amount of time spent waiting to acquire user kept storage during the interval (ss.ttt).</td>
</tr>
<tr>
<td>Avg Ukept Time</td>
<td>Average length of a wait to acquire user kept storage during the interval (ss.tttt).</td>
</tr>
<tr>
<td>Other Stg Waits</td>
<td>Number of waits to acquire terminal, database, or system storage</td>
</tr>
<tr>
<td>Other Wait Time</td>
<td>Amount of time spent waiting to acquire terminal, database, or system storage during the interval (ss.ttt).</td>
</tr>
<tr>
<td>Avg Other Time</td>
<td>Average length of a wait to acquire terminal, database, or system storage during the interval (ss.tttt).</td>
</tr>
</tbody>
</table>
### PMIRPT21 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>End Time</td>
<td>Ending time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>Area Name</td>
<td>Name of the area</td>
</tr>
<tr>
<td>File Name</td>
<td>Name of a file to which area maps</td>
</tr>
<tr>
<td>Buffer Name</td>
<td>Name of area's associated buffer</td>
</tr>
<tr>
<td>Area Access Waits</td>
<td>Number of times task waited to ready an area in a required usage mode</td>
</tr>
<tr>
<td>Area Accesses</td>
<td>Number of times task readied an area</td>
</tr>
<tr>
<td>Physical Writes</td>
<td>Number of physical writes for the area</td>
</tr>
<tr>
<td>Physical Reads</td>
<td>Number of physical reads for the area</td>
</tr>
<tr>
<td>Buffer Hits</td>
<td>Number of database area requests that could be processed in the buffer without a physical I/O</td>
</tr>
<tr>
<td>Read I/O Waits</td>
<td>Number of physical read I/Os that resulted in a wait and the total, highest, and average wait times</td>
</tr>
<tr>
<td>Write I/O Waits</td>
<td>Number of physical write I/Os that resulted in a wait and the total, highest, and average wait times</td>
</tr>
</tbody>
</table>
### Field Description

**DB Buffer Waits**  
Number of times a page within the area had to wait for a buffer page to become available and the total, highest, and average wait times.

**Shared Buffer Waits**  
Number of times transactions wanted to access a database page that was exclusively held by another transaction and the total, highest, and average wait times.

**Exclusive Buffer Waits**  
Number of times transactions waited for exclusive access to a database page and the total, highest, and average wait times.

**DBkey Waits**  
Number of waits for db-key and the total, highest, and average wait times.

---

### PMIRPT22 I/O by file detail report

PMIRPT22 contains detailed information on a file's input/output during a specific interval.

#### Sample report

```
PMIRPT22 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>End Time</td>
<td>Ending time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>File Name</td>
<td>Name of the file</td>
</tr>
<tr>
<td>Buffer</td>
<td>Name of buffer associated with file</td>
</tr>
</tbody>
</table>
```
### PMIRPT23 Buffer detail report

PMIRPT23 contains detailed information on a buffer's input/output during a specific interval.

#### Sample report

<table>
<thead>
<tr>
<th>INTERVAL START TIME: 15:00:00</th>
<th>END TIME: 15:10:00</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUFFER NAME: DCDML-BUFFER</td>
<td>BUFFER PAGE SIZE:</td>
</tr>
<tr>
<td>5 BUFFER READS</td>
<td>BUFFER PAGE: 655630</td>
</tr>
<tr>
<td>TOT WAITS: 39,000</td>
<td>TOT WAIT TIME: HIGHEST WAIT TIME</td>
</tr>
<tr>
<td>--DB BUFFER WAITS: --</td>
<td>--SHARED BUFFER WAITS: --</td>
</tr>
<tr>
<td>TOT WAIT TIME: HIGHEST WAIT TIME</td>
<td>AVG WAIT TIME: 26.000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INTERVAL START TIME: 15:00:00</th>
<th>END TIME: 15:10:00</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUFFER NAME: DCDLOD-BUFFER</td>
<td>BUFFER PAGE SIZE:</td>
</tr>
<tr>
<td>10 BUFFER READS</td>
<td>BUFFER PAGE: 655630</td>
</tr>
<tr>
<td>TOT WAITS: 39,000</td>
<td>TOT WAIT TIME: HIGHEST WAIT TIME</td>
</tr>
<tr>
<td>--DB BUFFER WAITS: --</td>
<td>--SHARED BUFFER WAITS: --</td>
</tr>
<tr>
<td>TOT WAIT TIME: HIGHEST WAIT TIME</td>
<td>AVG WAIT TIME: 26.000</td>
</tr>
</tbody>
</table>

### PMIRPT23 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read I/O Waits</td>
<td>Number of physical read I/Os that resulted in a wait and the total, highest, and average wait times</td>
</tr>
<tr>
<td>Write I/O Waits</td>
<td>Number of physical write I/Os that resulted in a wait and the total, highest, and average wait times</td>
</tr>
<tr>
<td>DB Buffer Waits</td>
<td>Number of times a page within the area had to wait for a buffer page to become available and the total, highest, and average wait times</td>
</tr>
<tr>
<td>Shared Buffer Waits</td>
<td>Number of times transactions wanted to access a database page that was exclusively held by another transaction and the total, highest, and average wait times</td>
</tr>
<tr>
<td>Exclusive Buffer Waits</td>
<td>Number of times transactions waited for exclusive access to a database page and the total, highest, and average wait times</td>
</tr>
<tr>
<td>DBkey Waits</td>
<td>Number of waits for a db-key and the total, highest, and average wait times</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>End Time</td>
<td>Ending time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>Buffer Name</td>
<td>Name of buffer</td>
</tr>
<tr>
<td>Buffer Page Size</td>
<td>Size of buffer pages</td>
</tr>
<tr>
<td>Buffer Page Defined</td>
<td>Number of pages defined for buffer</td>
</tr>
<tr>
<td>Buffer Reads</td>
<td>Number of times DBMS requested a new database page for which a physical I/O occurred</td>
</tr>
<tr>
<td>Buffer Writes</td>
<td>Number of times a buffer page was discarded in order to read another page</td>
</tr>
<tr>
<td>Buffer Pgs In Use</td>
<td>Number of pages currently in use in the buffer</td>
</tr>
<tr>
<td>Buffer Flushes</td>
<td>Number of times a page was discarded from the buffer in order to read another page</td>
</tr>
<tr>
<td>Buffer Requests</td>
<td>Total number of buffer requests (the sum of Pages Fnd in Pool and Buffer Reads)</td>
</tr>
<tr>
<td>Pages Fnd In Pool</td>
<td>Number of database area requests that could be processed in the buffer without a physical I/O</td>
</tr>
<tr>
<td>Pages Fnd Ratio (%)</td>
<td>Percent of Pages Fnd in Pool to Buffer Requests; this ratio should be as close to 100 as possible</td>
</tr>
<tr>
<td>Read I/O Waits</td>
<td>Number of physical read I/Os that resulted in a wait and the total, highest, and average wait times</td>
</tr>
<tr>
<td>Write I/O Waits</td>
<td>Number of physical write I/Os that resulted in a wait and the total, highest, and average wait times</td>
</tr>
<tr>
<td>DB Buffer Waits</td>
<td>Number of times a page within the area had to wait for a buffer page to become available and the total, highest, and average wait times</td>
</tr>
<tr>
<td>Shared Buffer Waits</td>
<td>Number of times transactions wanted to access a database page that was exclusively held by another transaction and the total, highest, and average wait times</td>
</tr>
<tr>
<td>Exclusive Buffer Waits</td>
<td>Number of times transactions waited for exclusive access to a database page and the total, highest, and average wait times</td>
</tr>
</tbody>
</table>

**PMIRPT24 CDMSLIB detail report**

PMIRPT24 contains detailed information on program load waits for a CDMSLIB during a specific interval.

**Sample report**
PMIRPT24 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>End Time</td>
<td>Ending time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>CDMSLIB Name</td>
<td>Name of the CDMSLIB library</td>
</tr>
<tr>
<td>Pgm Load Waits</td>
<td>Number of program load waits and the total, highest, and average wait time</td>
</tr>
</tbody>
</table>

PMIRPT25 Journal detail report

PMIRPT25 contains detailed information on journal waits for each reported interval.

Sample report
PMIRPT25 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>End Time</td>
<td>Ending time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>Journal Name</td>
<td>Name of journal, as defined in the DMCL</td>
</tr>
<tr>
<td>File</td>
<td>Name of external file associated with the journal in the DMCL</td>
</tr>
<tr>
<td>Jrnl Pgsiz</td>
<td>Page size defined for the journal</td>
</tr>
<tr>
<td>Begin RBN</td>
<td>Relative block number of the first block written to the journal during the interval</td>
</tr>
<tr>
<td>End RBN</td>
<td>Relative block number of the last block written to the journal during the interval</td>
</tr>
<tr>
<td>Blks Written</td>
<td>Number of blocks written to the journal</td>
</tr>
<tr>
<td>Bytes Written</td>
<td>Number of bytes written to the journal</td>
</tr>
<tr>
<td>Buffer Waits</td>
<td>Number of waits for the buffer (that is, buffer was requested but not available) and the total, highest, and average wait time</td>
</tr>
<tr>
<td>Read Waits</td>
<td>Number of physical read (rollback) I/Os against the journal that resulted in a wait and the total, highest, and average wait time</td>
</tr>
<tr>
<td>JBC Waits</td>
<td>Number of waits for a journal buffer control block and the total, highest, and average wait time</td>
</tr>
</tbody>
</table>

PMIRPT27 Program pool detail report

PMIRPT27 contains information on journal waits for each reported interval.

Sample report
## PMIRPT27 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock ((hh:mm:ss))</td>
</tr>
<tr>
<td>End Time</td>
<td>Ending time for the interval on a 24-hour clock ((hh:mm:ss))</td>
</tr>
<tr>
<td>Pool Type</td>
<td>Type of pool: program, XA program, reentrant and XA reentrant</td>
</tr>
<tr>
<td>Pool Size</td>
<td>Size of pool, in kilobytes</td>
</tr>
<tr>
<td>In Use</td>
<td>Kilobytes of storage occupied by programs at end of interval</td>
</tr>
<tr>
<td>High Water</td>
<td>Highest amount of storage used by programs since startup</td>
</tr>
<tr>
<td>Space Loaded</td>
<td>Kilobytes of storage used to load programs during the interval</td>
</tr>
<tr>
<td>Pgm Pool Loads</td>
<td>Number of programs loaded into the pool during the interval</td>
</tr>
<tr>
<td>Into Unalloc</td>
<td>Number of programs loaded into the pool during the interval without having</td>
</tr>
<tr>
<td></td>
<td>to overlay other Unalloc programs</td>
</tr>
<tr>
<td>Space overlay</td>
<td>Number of programs loaded into the pool during the interval that overlayed</td>
</tr>
<tr>
<td></td>
<td>inactive programs</td>
</tr>
</tbody>
</table>

**Data Format:**

```
<table>
<thead>
<tr>
<th>INTERVAL</th>
<th>START TIME: 15:00:00</th>
<th>END TIME: 15:10:00</th>
</tr>
</thead>
<tbody>
<tr>
<td>POOL TYPE:</td>
<td>XA REENTRANT</td>
<td></td>
</tr>
<tr>
<td>POOL SIZE (K)</td>
<td>3788</td>
<td></td>
</tr>
<tr>
<td>IN USE (K)</td>
<td>1583</td>
<td></td>
</tr>
<tr>
<td>HIGH WATER (K)</td>
<td>1583</td>
<td></td>
</tr>
<tr>
<td>SPACE LOADED (K)</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>OVERLAY PGM IN USE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOT WAITS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOT WAIT TIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG WAIT TIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIGHEST WAIT TIME</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INTERVAL</th>
<th>START TIME: 15:00:00</th>
<th>END TIME: 15:10:00</th>
</tr>
</thead>
<tbody>
<tr>
<td>POOL TYPE:</td>
<td>REENTRANT</td>
<td></td>
</tr>
<tr>
<td>POOL SIZE (K)</td>
<td>1364</td>
<td></td>
</tr>
<tr>
<td>IN USE (K)</td>
<td>394</td>
<td></td>
</tr>
<tr>
<td>HIGH WATER (K)</td>
<td>394</td>
<td></td>
</tr>
<tr>
<td>SPACE LOADED (K)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OVERLAY PGM IN USE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOT WAITS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOT WAIT TIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG WAIT TIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIGHEST WAIT TIME</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INTERVAL</th>
<th>START TIME: 15:00:00</th>
<th>END TIME: 15:10:00</th>
</tr>
</thead>
<tbody>
<tr>
<td>POOL TYPE:</td>
<td>PROGRAM</td>
<td></td>
</tr>
<tr>
<td>POOL SIZE (K)</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>IN USE (K)</td>
<td>136</td>
<td></td>
</tr>
<tr>
<td>HIGH WATER (K)</td>
<td>136</td>
<td></td>
</tr>
<tr>
<td>SPACE LOADED (K)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OVERLAY PGM IN USE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOT WAITS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOT WAIT TIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG WAIT TIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIGHEST WAIT TIME</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
PMIRPT29 Storage type detail report

PMIRPT29 contains information on waits for specific storage types for each reported interval.

Sample report

<table>
<thead>
<tr>
<th>INTERVAL</th>
<th>START TIME</th>
<th>END TIME</th>
<th>STORAGE TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>14:58:16</td>
<td>15:00:00</td>
<td>NON-XA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-- Shared STG Waits</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Tot Waits</td>
<td>Avg Wait Time</td>
<td>Tot Wait Time</td>
<td>Avg Wait Time</td>
</tr>
<tr>
<td>HIGHEST WAIT TIME</td>
<td>HIGHEST WAIT TIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-- User Storage Waits</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Tot Waits</td>
<td>Avg Wait Time</td>
<td>Tot Wait Time</td>
<td>Avg Wait Time</td>
</tr>
<tr>
<td>HIGHEST WAIT TIME</td>
<td>HIGHEST WAIT TIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-- Terminal STG Waits</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Tot Waits</td>
<td>Avg Wait Time</td>
<td>Tot Wait Time</td>
<td>HIGHEST WAIT TIME</td>
</tr>
<tr>
<td>-- System Storage Waits</td>
<td>--</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tot Waits</td>
<td>Avg Wait Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIGHEST WAIT TIME</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PMIRPT29 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>End Time</td>
<td>Ending time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
</tbody>
</table>
### PMIRPT30 Interval statistics summary report

PMIRPT30 contains DC/UCF statistics for each reported interval.

**Sample report**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>Tasks at Start</td>
<td>Number of tasks active at the beginning of the interval</td>
</tr>
</tbody>
</table>

**PMIRPT30 fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>Tasks at Start</td>
<td>Number of tasks active at the beginning of the interval</td>
</tr>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tasks at End</td>
<td>Number of tasks active at the end of the interval</td>
</tr>
<tr>
<td>Tasks Startd</td>
<td>Number of tasks that started during the interval</td>
</tr>
<tr>
<td>Tasks Ended</td>
<td>Number of tasks that ended during the interval</td>
</tr>
<tr>
<td>Task Abends</td>
<td>Number of tasks that ended abnormally during the interval</td>
</tr>
<tr>
<td>Task Stalls</td>
<td>Number of tasks that timed out during the interval</td>
</tr>
<tr>
<td>Times Max Task</td>
<td>The number of times during the interval that a maximum tasks condition existed</td>
</tr>
<tr>
<td>System Mode CPU</td>
<td>Amount of time during the interval that the DC/UCF system spent performing system services on behalf of tasks. The value is reported as seconds.</td>
</tr>
<tr>
<td>User Mode CPU</td>
<td>Amount of time during the interval that user tasks spent in execution. The value is reported as seconds.</td>
</tr>
<tr>
<td>Pgms Called</td>
<td>Number of programs called by tasks during the interval; includes: LINKs, XCTLs Programs called by the DC/UCF system on behalf of the task</td>
</tr>
<tr>
<td>Pgms Loaded</td>
<td>Number of programs called during the interval that were not present in the program pool and that needed to be loaded</td>
</tr>
<tr>
<td>Get Stg Rqsts</td>
<td>Number of GET STORAGE (#GETSTG) requests issued during the interval</td>
</tr>
<tr>
<td>Free Stg Rqsts</td>
<td>Number of FREE STORAGE (#FREESTG) requests issued during the interval</td>
</tr>
<tr>
<td>DC Svce Rqsts</td>
<td>Number of requests for DC/UCF services issued during the interval</td>
</tr>
<tr>
<td>DB Svce Rqsts</td>
<td>Number of requests for database services issued during the interval</td>
</tr>
</tbody>
</table>

### PMIRPT32 Run unit statistics summary report

PMIRPT32 contains database statistics for each reported interval. The report shows one column of information for each interval.

### Sample report

<table>
<thead>
<tr>
<th>START TIME</th>
<th>R/U AT STRT</th>
<th>R/U AT END</th>
<th>NUM R/U ENDED</th>
<th>NUM R/U ENDED NORMAL</th>
<th>NUM DBMS CALLS</th>
<th>RECS CURR OF R/U</th>
<th>REC'S R/O STWD</th>
<th>PAGES R/O STWD</th>
<th>PAGES READ</th>
<th>PAGES WRITTEN</th>
<th>CALCS NO OFLOW</th>
<th>CALCS WITH OFLOW</th>
<th>VIA NO OFLOW</th>
<th>VIA NO OFLOW</th>
<th>VIA RECS WITH OFLOW</th>
<th>VIA RECS WITH OFLOW</th>
<th>FRAGS STORED</th>
</tr>
</thead>
<tbody>
<tr>
<td>14:58:16</td>
<td>0</td>
<td>11</td>
<td>43</td>
<td>32</td>
<td>727</td>
<td>571</td>
<td>202</td>
<td>2312</td>
<td>2078</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:00:00</td>
<td>11</td>
<td>11</td>
<td></td>
<td></td>
<td>7</td>
<td>13</td>
<td>2</td>
<td>18</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:10:00</td>
<td>11</td>
<td>11</td>
<td></td>
<td></td>
<td>182</td>
<td>49</td>
<td>7</td>
<td>42</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:20:00</td>
<td>11</td>
<td>11</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>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start Time</td>
<td>R/U At Strt</td>
<td>R/U At End</td>
<td>Num R/U Strtd</td>
<td>Num R/U Ended Normal</td>
<td>Num DBMS Calls</td>
<td>Recs Rqstd</td>
<td>Recs Curr of R/U</td>
<td>Pages Rqstd</td>
<td>Pages Read</td>
<td>Pages Written</td>
<td>CALC Recs No Oflow</td>
<td>CALC Recs With Oflow</td>
<td>VIA Recs No Oflow</td>
<td>VIA Recs With Oflow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<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>---------------------</td>
<td>--------------------</td>
<td>--------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:30:00</td>
<td>11</td>
<td>11</td>
<td>750</td>
<td>758</td>
<td>18078</td>
<td>5749</td>
<td>2771</td>
<td>4104</td>
<td>756</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:40:00</td>
<td>11</td>
<td>12</td>
<td>466</td>
<td>465</td>
<td>14885</td>
<td>11163</td>
<td>6462</td>
<td>8131</td>
<td>1379</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:50:00</td>
<td>12</td>
<td>11</td>
<td>75</td>
<td>73</td>
<td>7499</td>
<td>7333</td>
<td>6141</td>
<td>10931</td>
<td>8331</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:00:00</td>
<td>11</td>
<td>12</td>
<td>4</td>
<td>3</td>
<td>38</td>
<td>6</td>
<td>21</td>
<td>19</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:10:00</td>
<td>12</td>
<td>12</td>
<td>39</td>
<td>37</td>
<td>11860</td>
<td>15916</td>
<td>11393</td>
<td>12269</td>
<td>12056</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:20:00</td>
<td>12</td>
<td>12</td>
<td>47</td>
<td>47</td>
<td>2388</td>
<td>3867</td>
<td>1852</td>
<td>32</td>
<td>1997</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:30:00</td>
<td>12</td>
<td>11</td>
<td>99</td>
<td>99</td>
<td>16710</td>
<td>19701</td>
<td>15749</td>
<td>19559</td>
<td>19237</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:40:00</td>
<td>11</td>
<td>11</td>
<td>24</td>
<td>24</td>
<td>665</td>
<td>178</td>
<td>8</td>
<td>19</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:50:00</td>
<td>11</td>
<td>11</td>
<td>483</td>
<td>145</td>
<td>11</td>
<td>124</td>
<td>23</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PMIRPT32 fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (<em>hh:mm:ss</em>)</td>
</tr>
<tr>
<td>R/U At Strt</td>
<td>Number of run units active at the start of the interval</td>
</tr>
<tr>
<td>R/U At End</td>
<td>Number of run units active at the end of the interval</td>
</tr>
<tr>
<td>Num R/U Strtd</td>
<td>Number of run units started during the interval</td>
</tr>
<tr>
<td>Num R/U Ended Normal</td>
<td>Number of run units ended during the interval</td>
</tr>
<tr>
<td>Num DBMS Calls</td>
<td>Number of times DBMS was called</td>
</tr>
<tr>
<td>Recs Rqstd</td>
<td>Number of records retrieved from the database as a result of run unit processing requests</td>
</tr>
<tr>
<td>Recs Curr of R/U</td>
<td>Number of records that became current of the run unit during the interval as the result of FIND, STORE, or OBTAIN requests</td>
</tr>
<tr>
<td>Pages Rqstd</td>
<td>Number of pages requested by the DBMS (difference of Pages Rqstd and Pages Read in the number of pages found in the buffer)</td>
</tr>
<tr>
<td>Pages Read</td>
<td>Number of pages physically read on behalf of run units during the interval</td>
</tr>
<tr>
<td>Pages Written</td>
<td>Number of physical writes that occurred while this run unit was in control; because IDMSDBIO writes pages as they are placed in the buffer, physical writes can occur for a program READYed in retrieval mode</td>
</tr>
<tr>
<td>CALC Recs No Oflow</td>
<td>Number of new records stored during the interval that fit on the target page using the CALC location method</td>
</tr>
<tr>
<td>CALC Recs With Oflow</td>
<td>Number of new records stored during the interval using the CALC location method that were placed on a page other than the target page</td>
</tr>
<tr>
<td>VIA Recs No Oflow</td>
<td>Number of new records stored during the interval that fit on the target page when using the VIA location method</td>
</tr>
<tr>
<td>VIA Recs With Oflow</td>
<td>Number of new records stored during the interval using the VIA location method that were placed on a page other than the target page</td>
</tr>
</tbody>
</table>
PMIRPT38 Journal block full detail report

PMIRPT38 contains detailed information on the number of journal blocks written for a report interval.

Sample report

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>Blks Written</td>
<td>Number of journal blocks that were written from the buffer during the interval; each column indicates the number of blocks that were 0-10% full, 11-20% full, and so on</td>
</tr>
<tr>
<td>Pct of Total</td>
<td>Percent of the total number of blocks written that were 0-10% full, 11-20% full, and so on</td>
</tr>
<tr>
<td>Cumulative</td>
<td>Cumulative percentage of blocks written in order of percent full; for example, 60.3% of the journal blocks written were 30% full or less</td>
</tr>
<tr>
<td>Total Blks WRITTEN</td>
<td>Total number of journal blocks written for the interval</td>
</tr>
</tbody>
</table>

PMIRPT38 fields
PMIRPT40 Data sharing SYSPLEX detail report

PMIRPT40 contains detailed information on the use of SYSPLEX resources when exploiting data sharing.

Sample report

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Total</th>
<th>Blks</th>
<th>Writn</th>
</tr>
</thead>
</table>

PMIRPT40 fields
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>Starting time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
<tr>
<td>End Time</td>
<td>Ending time for the interval on a 24-hour clock (hh:mm:ss)</td>
</tr>
</tbody>
</table>

**XES lock statistics**

The following table describes the fields that contain information on the usage of XES lock structure `strname`.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>Resource Type</td>
<td>Type of resource for which XES lock requests were issued:</td>
</tr>
<tr>
<td>LmgrResource</td>
<td>Lock manager resources (e.g. DBKeys)</td>
</tr>
<tr>
<td>Phys.Page Lk</td>
<td>DBIO buffer page locks</td>
</tr>
<tr>
<td>GlobalDead Lk</td>
<td>Deadlock manager locks</td>
</tr>
<tr>
<td>LmgrProxy Lk</td>
<td>Lock manager proxy locks</td>
</tr>
<tr>
<td>EnqDeq. Lock</td>
<td>Locks for global ENQ/DEQ processing</td>
</tr>
<tr>
<td>AreaList Lk</td>
<td>Locks associated with keeping track of areas that are shared</td>
</tr>
<tr>
<td>FileList Lk</td>
<td>Locks associated with keeping track of files that are shared</td>
</tr>
<tr>
<td>Global Queue</td>
<td>Locks for global queue area processing</td>
</tr>
<tr>
<td>Obtains</td>
<td>Total number of obtains done for the resource type</td>
</tr>
<tr>
<td>Alters</td>
<td>Total number of alters done for the resource type</td>
</tr>
<tr>
<td>Releases</td>
<td>Total number of releases done for the resource type</td>
</tr>
<tr>
<td>Waits</td>
<td>Total number of waits</td>
</tr>
<tr>
<td>Cumulative Wait Time</td>
<td>Total amount of time spent waiting for XES lock requests</td>
</tr>
<tr>
<td>Average Wait Time</td>
<td>Average amount of time spent waiting for XES lock requests</td>
</tr>
</tbody>
</table>

**XES list statistics**

The following table describes the fields that contain information on the usage of XES list structure `strname`.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>List Name</td>
<td>The internal name of the list:</td>
</tr>
<tr>
<td>Area List</td>
<td>Keeps track of areas that are shared</td>
</tr>
<tr>
<td>File List</td>
<td>Keeps track of files that are shared</td>
</tr>
<tr>
<td>Queue List</td>
<td>Keeps track of global queues</td>
</tr>
<tr>
<td>Reads</td>
<td>Total number of reads done on the list</td>
</tr>
<tr>
<td>Writes</td>
<td>Total number of writes done on the list</td>
</tr>
<tr>
<td>Deletes</td>
<td>Total number of deletes done on the list</td>
</tr>
<tr>
<td>Waits</td>
<td>Total number of waits</td>
</tr>
<tr>
<td>Cumulative Wait Time</td>
<td>Total amount of time spent waiting for XES list requests</td>
</tr>
<tr>
<td>Average Wait Time</td>
<td>Average amount of time spent waiting for XES list requests</td>
</tr>
</tbody>
</table>
Group member statistics

The following table describes the fields that contain information on the usage of this DC system of XCF group grpname for member memname.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Type</td>
<td>The internal name of the message:</td>
</tr>
<tr>
<td>Reply Msg</td>
<td>A reply to one of the other message types</td>
</tr>
<tr>
<td>Test Msg</td>
<td>Message type used for testing purposes</td>
</tr>
<tr>
<td>Sync. Stamp</td>
<td>Message type used for invalidating the cache for SQL catalogs</td>
</tr>
<tr>
<td>Global Dead Lk</td>
<td>Deadlock manager messages</td>
</tr>
<tr>
<td>DCMTDCUFSEND</td>
<td>Messages sent on behalf of a broadcasted DCMT, DCUF or SEND</td>
</tr>
<tr>
<td>Area File Val</td>
<td>Message type used for informing data sharing members of shared files and areas</td>
</tr>
<tr>
<td>Queue Msg</td>
<td>Message type used for informing data sharing members of shared queues</td>
</tr>
<tr>
<td>Program Msg</td>
<td>Message type used for informing data sharing members of automatic program invalidation</td>
</tr>
</tbody>
</table>

| Sends   | Total number of sends done for the message type                           |
| Receives | Total number of receives done for the message type                         |

PMIRPT90 Machine-readable copy

Statistics extracted by Report 00, output to either a tape or disk.

When you run PMIRPT90, you must run it with PMIRPT00. Additionally, you can use the following task parameters with PMIRPT90:

- CV NUMBER
- DATE FORMAT
- REPORT FROM/THRU

PMIRPT99 Input processing summary report

PMIRPT99 contains information on:

- Interval selection parameters: For more information, see Request Reports (Interval Monitor) (see page 369).

- Input card processing

- Input record processing statistics:
  - Records read by PMIRPT00
    - Records selected by PMIRPT00
    - Records dropped by PMIRPT00
For example, this category includes the earliest record read, the latest record read, and the different record types read.

- **Processing of multipart records**: Task wait type and interval type records take up more than one DC/UCF log record.

**Sample report**

<table>
<thead>
<tr>
<th>REPORT NO. 99</th>
<th>CA, INC.</th>
<th>mm/dd/yy PAGE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA IDMS/PM nn.n volser</td>
<td>INPUT PROCESSING SUMMARY REPORT</td>
<td></td>
</tr>
<tr>
<td>DATE FORMAT: DMY</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

-------------------------------

**INPUT CARD PROCESSING**

- CARDS READ: 1
- CARDS PROCESSED: 1
- COMMENT CARDS: 0
- CARD ERRORS: 0

**INPUT RECORD PROCESSING STATISTICS**

**RECORDS READ BY PMIRPT00**

- # STAT RECS READ: 4,467
- # PMAM RECS READ: 4,179
- # PMIM RECS READ: 288
- EARLIEST REC READ: 07:54 ON 07/06/10 (10/158)
- LATEST REC READ: 08:10 ON 30/09/10 (99/274)

**BY RECORD TYPE**

- AREA WAITS: 284
- BUFFER WAITS: 12
- COMSLIB WAITS: 3
- INTERVAL STATS: 3
- INTERVAL WAITS: 6
- JOURNAL WAITS: 12
- LINE WAITS: 15
- PGMPOOL WAITS: 12
- RUNUNIT STATS: 3
- STGPOOL STATS: 9
- STG TYPE WAITS: 6
- DBGROUP WAITS: 3

**RECORDS SELECTED BY PMIRPT00**

<table>
<thead>
<tr>
<th>REPORT NO. 99</th>
<th>CA, INC.</th>
<th>mm/dd/yy PAGE 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA IDMS/PM nn.n volser</td>
<td>INPUT PROCESSING SUMMARY REPORT</td>
<td></td>
</tr>
<tr>
<td>DATE FORMAT: DMY</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**# PMIM RECS SELECTED**: 186

**EARLIEST REC SELECTD**: 07:54 ON 07/06/10 (10/158)

**LATEST REC SELECTD**: 08:10 ON 07/06/10 (10/158)

**BY RECORD TYPE**

- AREA WAITS: 102
- BUFFER WAITS: 12
- COMSLIB WAITS: 3
- INTERVAL STATS: 3
- INTERVAL WAITS: 6
- JOURNAL WAITS: 12
- LINE WAITS: 15
Application Monitor Batch Reports

You request Application Monitor reports using an CA Culprit job stream. The job control language you need to run the reports is shown in Preparing to Run Reports (see page 345). In the job stream, you supply:

- Selection Criteria Parameters -- for including and/or excluding specific information from the reports
- Report specification parameters -- for specifying the dictionary to use, formatting options, and the appropriate report names

You can request any or all of the reports in a single run.

General rules for parameter input

- Every parameter is optional.

- Include any or all of these parameters in a single run.

- Use a single line for each separate parameter.

- If you specify more than one parameter, all conditions that you specify must be met in order for you to select a task for reporting.

- Use columns 1 through 72. Input beyond column 72 is ignored. No error is flagged (unless a quoted description is truncated).

- An asterisk (*) in column 1 indicates a comment line.

- Specify either the 3-letter abbreviation or the whole word. For example, PROGRA is invalid. The syntax rules indicate (in uppercase characters) any other allowable abbreviations or synonyms.

- Blank lines are ignored but generate a warning message.
Request Reports (Application Monitor)

Selection criteria parameters, for Application Monitor reports, are used in CA Culprit JCL statements to include or exclude information from your Performance Monitor reports. The parameters apply to all of the reports you request in the same run. For example, if your selection criteria specifies reporting only for tasks within a certain time period, that time period is used for all of the reports in the run.

The following table describes how to code JCL statements when you want to define or omit the report parameters:

<table>
<thead>
<tr>
<th>System</th>
<th>Defining Report Parameters</th>
<th>Omitting Report Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS</td>
<td>Following the //SYS010 DD * statement</td>
<td>Use //SYS010 DD DUMMY</td>
</tr>
<tr>
<td>z/VSE</td>
<td>Following the /* in the EXEC CA Culprit step</td>
<td>Leave out the parameters</td>
</tr>
<tr>
<td>z/VM</td>
<td>In the SYS010 file</td>
<td>Use SYS010 DUMMY</td>
</tr>
</tbody>
</table>

This article describes the following information:

- Syntax (see page 408)
- Report selection parameters (see page 412)

Syntax

Application Monitor selection criteria syntax and parameter descriptions follow. Interpret the word task to mean either task or CA ADS dialog, as appropriate. You can omit leading zeros where syntax uses a number, unless otherwise noted.

```
►►►► ABOrt CODe ← abend-code ►►►►
►►►► ABOrt MESsage ← message-id ►►►►
►►►► ACTive STOrage ← storage-amount ►►►►
►►►► BILling GROup ← billing-group-code ►►►►
►►►► CPU time ← cpu-amount ►►►►
►►►► CV NUMBER ← dc/ucf-version-number ►►►►
►►►► DATe FORmat ← MDY ►►►►
                                       └─ DMY ── DYM ── YMD ── YDM ──
►►►► DB CALLs ← db-call-threshold ►►►►
►►►► DB LEVels ← ads-database-levels ►►►►
►►►► ELApsed TIMe ← elapsed-seconds ►►►►
►►►► IO ← physical-io-amount ►►►►
►►►► KEPT STOrage ← kept-storage-amount ►►►►
```
Parameters

- **ABOrt CODe abend-code**
  Selects only those tasks that abended with the (4-character) database/data communications task abend code specified. *Abend-code* is a 4-character database/data communications abend code. You can specify up to 20 codes. The abbreviation ABRT is permitted.

- **ABOrt MESSage message-id**
  Selects only those tasks that abended with the DC/UCF error message ID specified (6 digits, excluding the severity code suffix). *Message-id* is the six-digit DC/UCF message ID. You can select up to 20 IDs. The abbreviations ABRT and MSG are permitted.

- **ACTive STOrage storage-amount**
  Selects only those tasks that used more than the specified number of bytes of main memory from a storage pool during active execution. You can use the abbreviation STG.

- **BILLing GROup billing-group-code**
  Selects tasks by billing-group code. *Billing-group-code* is a 1- to 12-character billing-group code. Use single quotes if it contains embedded spaces. Use two quotation marks to indicate a quotation mark that is part of the description. Up to 20 codes are permitted. You can use the abbreviation GRP.

- **CPU time cpu-amount**
  Selects only those tasks that used more than *nnnn.nnnn* seconds of CPU time. *Cpu-amount* is a number between .0001 and 999999.9999.

- **CV NUMber dc/ucf-version-number**
  Selects only those tasks that ran under the specified DC/UCF system. *Dc/ucf-version-number* is a number between 0 and 9999. You can place multiple values on one line and you can use the abbreviation NBR. Up to 20 CV numbers are permitted.
- **DATE FORmat MDY/DMY/MYD/DYM/YMD/YDM**
  Specifies the date format that appears on the reports. The default is MDY. You can use the abbreviation FMT.

- **DB CALLs database-call-threshold**
  Selects only those tasks that issued more than the specified number of database calls. Synonyms you can use are DB and DBCALLS.

- **DB LEVELs ads-database-levels**
  Selects only those CA ADS dialogs that issued database calls from more than the specified number of application-thread levels. Non CA ADS tasks are dropped. You can use the abbreviation LVLS.

- **ELApsed TIME elapsed-seconds**
  Selects only those tasks with an elapsed time longer than the specified number of seconds. The elapsed time is a number between .0001 and 9999.9999 and is the internal CA IDMS response time, and it is measured from task initiation to task termination within the DC/UCF system.

- **IO physical-io-amount**
  Selects only those tasks that issued more than the specified number of physical disk I/Os.

- **KEPt STOrage kept-storage-amount**
  Selects only those tasks that kept more than the specified number of bytes of main memory from a storage pool after task termination (across a pseudo-converse). You can use the abbreviation STG.

- **LEVELs dialog-levels**
  Selects only those CA ADS dialogs that processed more than the specified number of levels in the application thread. Non CA ADS tasks are dropped. You can use the abbreviation LVLS.

- **LTErm logical-terminal-id**
  Selects only tasks initiated from the logical terminal specified. Logical-terminal-id is a 1- to 8-character logical terminal ID. Up to 50 IDs are permitted.

- **PROgram program-name**
  Selects only those tasks that execute the named program at the first level. Program-name is a 1- to 8-character program name. Up to 50 program names are permitted. You can use the abbreviations PROG and PGM.

- **PTErm physical-terminal-id**
  Selects only tasks initiated from the physical terminal specified. Physical-terminal-id is a 1- to 8-character physical terminal ID. Up to 50 IDs are permitted.

- **RBB number-of-rbbs**
  Selects only those CA ADS dialogs that obtained more than the specified number of record buffer blocks. Non CA ADS tasks are dropped.

- **REPort FROM/THRu**
  Selects intervals to be included in the report. If you want to report on the entire input file, don’t include this parameter. You can specify this parameter once per run, and you must specify at least one FROM or one THRU. The default is FROM 00:00 ON 00/001 THRU 24:00 ON 99/365. Regarding the time specification:
Specify the time as *hh:mm* or *hhmm* (00:00 through 24:00).

Times include the entire minute. For example, THRU 14:34 means up to 14:34:59.9999.

Times must include the leading 0. For example, 09:00 is valid, but 9:00 is not.

If you specify a time range, the FROM time must be earlier than the THRU time.

**Regarding the date specification:**

- Julian: *yy/ddd*
- Gregorian: as specified by DATE FORMAT
- The FROM date must be earlier or matching the THRU date.
- Slashes are optional in date specifications.

**SCAling FAc tor nnnnnn.nnnn**

Defines a scaling factor for report graphs. *nnnnnn.nnnn* is a numeric value that specifies the scaling factor (for example, .01 results in scaling of data in hundredths). The decimal point is not required and, if present, can be leading or trailing. Any more than 4 digits to the right of the decimal point are truncated. For example, 1.2345678 will be truncated to 1.2345. The default is 1.0. You can use the synonym SCALE FACTOR. About the values you can specify:

- 0 is invalid.
- The maximum is 999999.9999.

Examples of valid values follow:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>123456</td>
<td>.3456</td>
</tr>
<tr>
<td>1234.5678</td>
<td>45.</td>
</tr>
<tr>
<td>000000.01</td>
<td>0.3</td>
</tr>
</tbody>
</table>

**TASk task-code**

Selects only tasks with the task code or (for CA ADS) dialog name specified. *Task-code* is a 1- to 8-character task code or dialog name. Up to 50 task codes are permitted.

**TP REAd bytes-read**

Selects only those tasks that read in more than the specified number of bytes from the terminal.

**TP WRite bytes-written**

Selects only those tasks that wrote out more than the specified number of bytes to the terminal.

**TYPe task-type**

Selects only tasks of the type specified. Possible values for *task-type* are shown in the following table.

<table>
<thead>
<tr>
<th>Task-type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADS/O</td>
<td>DC task whose first-level program language is the CA ADS process language</td>
</tr>
<tr>
<td>ASSEM</td>
<td>DC task whose first-level language is Assembler</td>
</tr>
<tr>
<td>COBOL</td>
<td>DC task whose first-level language is COBOL</td>
</tr>
</tbody>
</table>
### Task-type Description

- **PL/I**: DC task whose first-level language is PL/I
- **CICS**: CICS task
- **TPMON**: Task initiated through a TP monitor other than CICS or a DC/UCF system
- **BATCH**: Batch ERUS
- **ERUS**: ERUS when PERFMON=NO is specified in the CA IDMS operating-system-specific SVC macro
- **SYSTM**: DC/UCF system internal task
- **UNDEF**: Undefined
- **SUBSC**: Subschema task
- **MAP**: Map task
- **TABLE**: Table task
- **DC/FE**: IDMS/DC Front End ERUS task

- **USEr user-id**
  Selects only those tasks invoked by the specified user. *User-id* is a 1- to 8-character user ID. Use quotes if it contains embedded spaces. Use two quotation marks to indicate a quotation mark that is part of *user-id*.
  Up to 20 user IDs are permitted. You can use the synonym USERID.

### Examples

The parameters below select only those tasks for which the first-level program was written in CA ADS or COBOL that ran on June 7, 2010, during prime time (between 9 a.m. and 5 p.m.), and that used more than 2.5 seconds of CPU time.

```
TYPE ADS/O
TYPE COBOL
FROM 09:00 ON 10158 THRU 17:00 ON 10158
CPU TIME 2.5000
```

The parameters below select only those executions of tasks CSFDURLJ and CSFDUMVJ that issued more than 30 database calls and that abended with DC/UCF task abend code D004 (indicating that the task took more CPU time than was allowed). No time or date parameters are specified, so the entire period represented by the input file is considered.

```
TASK CSFDURLJ
TASK CSFDUMVJ
ABORT CODE D004
DB CALLS 30
```

### Report selection parameters

Report selection parameters define:

- The dictionary that contains the report definitions
- Whether to print CA Culprit parameters
- Which reports to produce
Positioning report selection parameters

Position report selection parameters in the report-request JCL stream, using one line for each parameter:

<table>
<thead>
<tr>
<th>System</th>
<th>Position in JCL Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS</td>
<td>Following the //SYSIN DD * statement</td>
</tr>
<tr>
<td>z/VSE</td>
<td>Following the EXEC CA Culprit statement</td>
</tr>
<tr>
<td>z/VM</td>
<td>Following the DATABASE statement</td>
</tr>
</tbody>
</table>

Syntax and parameter descriptions for report selection parameters follow.

**Syntax**

```
DAtabase  DICTNAME= dictionary-name
          DICTNODE= nodename

PARAM= LIST
       NOLIST
       EJECT

=COPY 'report-name'
```

**Parameters**

- **DAtabase**
  Defines the data dictionary that contains the report definitions (DICTNAME option) or the node that controls the dictionary (DICTNODE option). Start this parameter in column 2.

- **PARAM=LIST/NOLIST/EJECT**
  Controls printing of the CA Culprit Sequential Input Parameter List:
  - LIST (default) prints all parameters
  - NOLIST prints no parameters
  - EJECT starts each new listing at the top of a new page

  Start this parameter in column 2.

- **=COPY 'report-name'**
  Requests the named report; begin =COPY in column 1; you can repeat the parameter any number of times. *Report-name* must be enclosed in quotes. Acceptable values for *report-name* are given in the following table.

<table>
<thead>
<tr>
<th>Value for Report-name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMARPT00</td>
<td>Performs housekeeping functions and extracts statistics for input to other reports; required, but not an output report</td>
</tr>
<tr>
<td>PMNAME</td>
<td>Supply the user site or company name to be printed in the heading of each report; required, but not an output report</td>
</tr>
<tr>
<td>PMARPT99</td>
<td>List an input processing summary based on the selection criteria specified</td>
</tr>
<tr>
<td>PMARPTnn</td>
<td></td>
</tr>
</tbody>
</table>
### Value for Report-name

Produce the report defined by the number (nn) specified:

- 01 Task Detail Report
- 02 Task Summary Report
- 03 CA ADS Dialog Detail Report
- 04 CA ADS Dialog Summary Report
- 05 User Detail Report
- 06 User Summary Report
- 07 Billing Group Detail Report
- 08 Billing Group Summary Report
- 09 Abnormal Termination Detail Report
- 10 Abnormal Termination Summary Report
- 11 LTERM Detail Report
- 12 LTERM Summary Report
- 13 PTERM Detail Report
- 14 PTERM Summary Report
- 15 System Detail Report
- 16 System Summary Report
- 17 Database Detail Report
- 18 Database Summary Report
- 19 DC Detail Report
- 20 DC Summary Report
- 21 Batch Job Accounting Summary Report
- 31 Task Wait Summary Report
- 36 Task Wait Detail Report
- 80 Load Balancing Report (by day and central version)
- 81 Load Balancing Report (by day)
- 82 Load Balancing Report (all central versions)
- 90 Machine-readable copy of the extracted statistics, output either to tape or disk
- 97 Summary Recap Report

### Examples

The following parameters select all printed reports. The CA Culprit report definitions are stored in the DICTCAS dictionary (DATABASE DICTNAME=DICTCAS). The report source (PARAM=NOLIST) is not printed.

```
DATABASE DICTNAME=DICTCAS
PARAM=NOLIST
=COPY 'PMARPT00'
=COPY 'PMNAME'
=COPY 'PMARPT99'
=COPY 'PMARPT01'
=COPY 'PMARPT02'
=COPY 'PMARPT03'
=COPY 'PMARPT04'
=COPY 'PMARPT05'
=COPY 'PMARPT06'
=COPY 'PMARPT07'
=COPY 'PMARPT08'
=COPY 'PMARPT09'
=COPY 'PMARPT10'
=COPY 'PMARPT11'
=COPY 'PMARPT12'
=COPY 'PMARPT13'
=COPY 'PMARPT14'
=COPY 'PMARPT15'
```
The parameters below request all summary reports, as well as an CA Culprit source listing for each report. The site uses only one dictionary, so there is no DATABASE parameter.

```
PARAM=LIST
=COPY 'PMARPT00'
=COPY 'PMARPT99'
=COPY 'PMARPT02'
=COPY 'PMARPT04'
=COPY 'PMARPT06'
=COPY 'PMARPT08'
=COPY 'PMARPT12'
=COPY 'PMARPT14'
=COPY 'PMARPT16'
=COPY 'PMARPT18'
=COPY 'PMARPT20'
=COPY 'PMARPT21'
```

List of Report Samples

This article describes the following reports and includes report samples:

- PMARPT01 Task detail report (see page 416)
- PMARPT02 Task summary report (see page 418)
- PMARPT03 CA ADS dialog detail report (see page 420)
- PMARPT04 CA ADS dialog summary report (see page 421)
- PMARPT05 User detail report (see page 421)
- PMARPT06 User summary report (see page 423)
- PMARPT07 Billing group detail report (see page 425)
- PMARPT08 Billing group summary report (see page 427)
- PMARPT09 Abnormal termination detail report (see page 428)
- PMARPT10 Abnormal termination summary report (see page 429)
- PMARPT11 LTERM detail report (see page 429)
- PMARPT12 LTERM summary report (see page 430)
- PMARPT13 PTERM detail report (see page 431)
- PMARPT14 PTERM summary report (see page 431)
- PMARPT15 System detail report (see page 432)
- PMARPT16 System summary report (see page 432)
- PMARPT17 Database detail report (see page 433)
- PMARPT18 Database summary report (see page 435)
- PMARPT19 DC statistics detail report (see page 435)
- PMARPT20 DC statistics summary report (see page 437)
Required reports

The following required reports do not have output:

- **PMARPT00**: Reads the input (archive) tape and formats it into global data fields; the data fields provide the input for all other reports.

- **PMNAME**: Reads the PMNAME module and inserts its contents into a global field called COMPANY-NAME; this produces the heading for each report.

Optional reports

The remaining optional reports for the Interval Monitor are described in numeric order. Each report description includes:

- An overview description

- A sample listing

- A description of the fields in the report

PMARPT01 Task detail report

PMARPT01 contains one detail line for every execution of each task reported. With CA ADS, it contains one detail line for every execution of each dialog. In the descriptions below, the word task should be interpreted as meaning either task or CA ADS dialog, as appropriate to the task type displayed.

Sample report

```
REPORT NO. 01
CA IDMS/PM nn.n<br>DC SYSTEM VERSION #: 71
TASK DETAIL REPORT
<----------- YOUR COMPANY NAME ----------->
DATA FROM: mm/dd/yy

CPU  WAIT  TP  TP  NUM  NUM  NUM  NUM  NUM  NUM  NUM  OF  OF  OF  OF  OF
TASK  VER  TASK  TASK C  START  STORAGE  STORAGE  TIME  TIME  READ  WRITE  OF  OF  OF  OF  OF
CODE  NUM  NUM  TYPE C  TIME  ACQUIRED  KEPT  (SECS)  (SECS)  LENGTH LENGTH I/O  DBCLS  LVLS  DBLVLS  BUFS
QUED   0  18 ASSEM 7:54:11 5760 0 .8927 4.5380 0 0 1002 26 -- -- -- --
CLOD   0  19 ASSEM 7:54:11 13056 0 .0552 .3059 0 0 10 94 -- -- -- --
FACTOTUM 1  20 ASSEM 7:57:40 0 0 .0011 .2076 0 27 0 27 -- -- -- --
S      0  22 ASSEM 7:57:43 9600 0 .0179 .0547 11 0 3 104 -- -- -- --
FACTOTUM 1  23 ASSEM 7:57:43 256 512 .0025 .0023 0 0 0 8 -- -- -- --
C      0  24 ASSEM 7:57:49 17280 768 .0141 .0787 15 0 1 24 -- -- -- --
FACTOTUM 1  25 ASSEM 7:57:50 256 512 .0025 .0023 0 0 0 12 -- -- -- --
DCMT   0  26 ASSEM 7:57:50 9600 768 .0224 .0941 29 0 1 6 -- -- -- --
```
### PMARPT01 fields

The following table describes the fields contained in PMARPT01.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Code</td>
<td>Task code or CA ADS dialog name</td>
</tr>
<tr>
<td>Ver Num</td>
<td>Version number of the level-1 program executed for the task defined above</td>
</tr>
<tr>
<td>Task Num</td>
<td>Sequential number assigned to the task at task initiation (also known as the task ID)</td>
</tr>
<tr>
<td>Task Type</td>
<td>Source language for the level-1 program for the task (ERUS for an external request unit)</td>
</tr>
<tr>
<td>CC</td>
<td>Completion code for the task: X if the task terminated abnormally; otherwise, the field is blank</td>
</tr>
<tr>
<td>Start Time</td>
<td>Time the task was initiated (hh:mm:ss)</td>
</tr>
<tr>
<td>Storage Active</td>
<td>Number of bytes of variable storage from a DC/UCF storage pool used by the task during execution</td>
</tr>
<tr>
<td>Storage Kept</td>
<td>Number of bytes of variable storage kept by the task at termination for pseudo-conversational processing; this does not include relocated storage for CA ADS</td>
</tr>
</tbody>
</table>
### Field | Description
--- | ---
CPU Time | Total CPU time for the task (ss.ssss)
Wait Time | Total wait time for the task (ss.ssss)
TP Read Length | Total number of bytes read from the terminal during task processing
TP Write Length | Total number of bytes written to the terminal during task processing
Num Of I/O | Number of physical I/Os performed by the task
Num of DBcs | Number of database calls issued by the task
Num of Lvls | Number of dialog levels in the CA ADS application structure; field is applicable to CA ADS only
Num of DBLvls | Number of dialog levels that issued database calls; this is the number of different levels, not the highest level number; field is applicable to CA ADS only
Num of Bufs | Number of record buffer blocks acquired for database record processing; field is applicable to CA ADS dialogs only

**PMARPT02 Task summary report**

PMARPT02 contains one summary line for each different task executed or, for CA ADS, for each different dialog. In the descriptions below, the word task should be interpreted as meaning either task or CA ADS dialog, as appropriate to the task type displayed.

**Sample report**
PMARPT02 fields

The following table describes the fields contained in PMARPT02.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Code</td>
<td>Task code or CA ADS dialog name</td>
</tr>
<tr>
<td>Ver Num</td>
<td>Version number of the level-1 program executed for the task defined above</td>
</tr>
<tr>
<td>Num Times</td>
<td>Number of times the task was executed</td>
</tr>
<tr>
<td>Exec</td>
<td>Number of times the task terminated abnormally</td>
</tr>
<tr>
<td>Task Type</td>
<td>Source language for the level-1 program for the task (ERUS for an external request unit)</td>
</tr>
<tr>
<td>Num Times</td>
<td>Number of times the task terminated abnormally</td>
</tr>
<tr>
<td>Abnd</td>
<td>Number of times the task terminated abnormally</td>
</tr>
<tr>
<td>Avg Storage</td>
<td>Average number of bytes of variable storage from a DC/UCF storage pool used by the task during execution</td>
</tr>
<tr>
<td>Active</td>
<td>Average number of bytes of variable storage from a DC/UCF storage pool used by the task during execution</td>
</tr>
<tr>
<td>Kept</td>
<td>Average number of bytes of variable storage kept by the task at termination for pseudo-conversational processing; this does not include relocated storage for CA ADS</td>
</tr>
<tr>
<td>Avg CPU Time</td>
<td>Average CPU time for the task (ss.ssss)</td>
</tr>
<tr>
<td>Avg Wait Time</td>
<td>Average wait time for the task (ss.ssss)</td>
</tr>
<tr>
<td>Avg TP Read Lngth</td>
<td>Average number of bytes read from the terminal during task processing</td>
</tr>
<tr>
<td>Avg TP Write Lngth</td>
<td>Average number of bytes written to the terminal during task processing</td>
</tr>
<tr>
<td>Avg Num of I/O</td>
<td>Average number of physical I/Os performed by the task</td>
</tr>
<tr>
<td>Avg Num of Dbcis</td>
<td>Average number of database calls issued by the task</td>
</tr>
</tbody>
</table>
PMARPT03 CA ADS dialog detail report

PMARPT03 contains one detail line for every execution of each CA ADS dialog. The fields found in PMARPT03 are identical to those in Report 01 (except column 1, which is Dialog Name, rather than Task Code). See PMARPT01: Task detail report (see page) for detailed field information. Task Type, shown in Report 01, does not apply here.

Sample report
PMARPT04 CA ADS dialog summary report

PMARPT04 contains one summary line for each different CA ADS dialog executed. The fields in PMARPT04 are identical to those in Report 02 (except column 1, which is Dialog Name, rather than Task Code). See PMARPT02: Task summary report (see page) for detailed field information. Task Type, shown in Report 02, does not apply here.

Sample report

PMARPT05 User detail report

PMARPT05 contains one detail line for every execution of each task, or each CA ADS dialog, executed by the user. The user is identified at the top of the report. The detail report is followed by a summary recap of user activity.
### User identification

The user identification at the top of the report varies, depending on the circumstances at the time the tasks shown on the report were executed. A prefix to the user identification indicates how the tasks were executed. The user identification itself corresponds to the prefix.

If no user was signed on under the DC/UCF system or no operator ID was available for CICS, an appropriate message replaces the ID. For example, in the DC/UCF system, the message is DC-NO USER ID AVAILABLE.

<table>
<thead>
<tr>
<th>PrefixTask executed under</th>
<th>User identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC DC/UCF</td>
<td>DC/UCF user ID</td>
</tr>
<tr>
<td>CICS CICS</td>
<td>CICS operator ID</td>
</tr>
<tr>
<td>TPM TP monitor other than DC/UCF or CICS</td>
<td>TP monitor ID</td>
</tr>
<tr>
<td>ON ERUS</td>
<td>None: BATCH displays instead of an ID</td>
</tr>
</tbody>
</table>

---

**Sample report**

**REPORT NO. 05**

**USER: DBCRUSER**

**CPU TIME**

<table>
<thead>
<tr>
<th>CPU</th>
<th>WAIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>TP</td>
</tr>
</tbody>
</table>

**TASK**

<table>
<thead>
<tr>
<th>CPU TIME</th>
<th>CPU TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>TP</td>
</tr>
</tbody>
</table>

**START STORAGE**

<table>
<thead>
<tr>
<th>STORAGE TYPE</th>
<th>STORAGE TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>C</td>
</tr>
</tbody>
</table>

**TOTAL DATABASE CALLS**

<table>
<thead>
<tr>
<th>TOTAL DATABASE CALLS</th>
<th>TOTAL DATABASE CALLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1199</td>
<td>1199</td>
</tr>
</tbody>
</table>

**TOTAL NUMBER OF TASKS**

<table>
<thead>
<tr>
<th>TOTAL NUMBER OF TASKS</th>
<th>TOTAL NUMBER OF TASKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1199</td>
<td>1199</td>
</tr>
</tbody>
</table>

**TOTAL PHYSICAL 1/0**

<table>
<thead>
<tr>
<th>TOTAL PHYSICAL 1/0</th>
<th>TOTAL PHYSICAL 1/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>193</td>
<td>193</td>
</tr>
</tbody>
</table>

**TOTAL DATABASE CALLS**

<table>
<thead>
<tr>
<th>TOTAL DATABASE CALLS</th>
<th>TOTAL DATABASE CALLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>5267</td>
<td>5267</td>
</tr>
</tbody>
</table>

**TOTAL CPU TIME (HH.MM.SS):** 0:00:13

**TOTAL CPU TIME**: 0:00:13

**TOTAL NUMBER OF TASKS**: 1199

**TOTAL PHYSICAL 1/0**: 193

**TOTAL DATABASE CALLS**: 5267

**AVG TP I/O READ LENGTH**: 51

**AVG TP I/O WRITE LENGTH**: 1718

**AVG STORAGE KEPT**: 5501

**AVG STORAGE USED**: 36187

**AVG STORAGE KEPT**: 5501

**AVG STORAGE USED**: 36187

---

User ID AVAILABLE.

The appropriate message replaces the ID. For example, in the DC/UCF system, the message is DC-NO USER ID AVAILABLE.

---

16-Jan-2018 422/898
PMARPT05 fields

The fields in PMARPT05 are identical to those in report 01. See PMARPT01: Task detail report (see page) for detailed field descriptions.

Descriptions of the additional Summary Recap fields at the end of PMARPT05 are shown in the following table. Interpret the word task to mean either task or CA ADS dialog, as appropriate. The fields in PMARPT05 are identical to those in Report 01. See PMARPT01: Task detail report (see page) for detailed field information.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Number of Tasks</td>
<td>Total tasks executed by the user</td>
</tr>
<tr>
<td>Total Physical I/O</td>
<td>Total number of physical I/Os performed by the tasks</td>
</tr>
<tr>
<td>Total Database Calls</td>
<td>Total number of database calls issued by the tasks</td>
</tr>
<tr>
<td>Total Abnormal Terminations</td>
<td>Total number of tasks that terminated abnormally</td>
</tr>
<tr>
<td>Total CPU Time</td>
<td>Total CPU time for the above tasks (hh:mm:ss)</td>
</tr>
<tr>
<td>Average Storage Used</td>
<td>Average number of bytes of variable storage from a DC/UCF storage pool used by the tasks during execution</td>
</tr>
<tr>
<td>Average Storage Kept</td>
<td>Average number of bytes of variable storage kept by the tasks at task termination for pseudo-conversational processing; this does not include relocated storage for CA ADS</td>
</tr>
<tr>
<td>Average TP I/O Read Length</td>
<td>Average number of bytes read from the terminal during task processing</td>
</tr>
<tr>
<td>Average TP I/O Write Length</td>
<td>Average number of bytes written to the terminal during task processing</td>
</tr>
</tbody>
</table>

PMARPT06 User summary report

PMARPT06 contains one summary line for each different task, or each different CA ADS dialog, executed by the user identified at the top of the report. The report is followed by a summary recap of user activity.

Sample report
### User identification

The user identification at the top of the report varies, depending on the circumstances at the time the tasks shown on the report were executed. A prefix to the user identification specifies how the tasks were executed. The user identification itself corresponds to the prefix:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Task executed under</th>
<th>User identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>DC/UCF</td>
<td>DC/UCF user ID</td>
</tr>
<tr>
<td>CICS</td>
<td>CICS</td>
<td>CICS operator ID</td>
</tr>
<tr>
<td>TPM</td>
<td>TP monitor other than DC/UCF or CICS ON</td>
<td>TP monitor ID</td>
</tr>
<tr>
<td>ERUS</td>
<td>Batch</td>
<td>None: BATCH displays instead of an ID</td>
</tr>
<tr>
<td></td>
<td>ERUS, if PERFMON=NO is specified in the CA IDMS operating-system-specific SVC macro</td>
<td>None: UNDEFINED displays instead of an ID</td>
</tr>
</tbody>
</table>

If no user was signed on under the DC/UCF system or no operator ID was available for CICS, an appropriate message replaces the ID. For example, in the DC/UCF system, the message is DC-NO USER ID AVAILABLE.

### PMARPT06 fields

The fields in PMARPT06 are identical to those in report 02. See PMARPT02: Task summary report (see page ) for detailed field descriptions.

Descriptions of the additional Summary Recap fields at the end of PMARPT06 are shown in the following table. Interpret the word task to mean either task or CA ADS dialog, as appropriate.
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Number of Tasks</td>
<td>Total tasks executed by the user</td>
</tr>
<tr>
<td>Total Physical I/O</td>
<td>Total number of physical I/Os performed by the tasks</td>
</tr>
<tr>
<td>Total Database Calls</td>
<td>Total number of database calls issued by the tasks</td>
</tr>
<tr>
<td>Total Abnormal Terminations</td>
<td>Total number of tasks that terminated abnormally</td>
</tr>
<tr>
<td>Total CPU Time</td>
<td>Total CPU time for the above tasks ((hh:mm:ss))</td>
</tr>
<tr>
<td>Average Storage Used</td>
<td>Average number of bytes of variable storage from a DC/UCF storage pool used by the tasks during execution</td>
</tr>
<tr>
<td>Average Storage Kept</td>
<td>Average number of bytes of variable storage kept by the tasks at task termination for pseudo-conversational processing; this does not include relocated storage for CA ADS</td>
</tr>
<tr>
<td>Average TP I/O Read</td>
<td>Average number of bytes read from the terminal during task processing</td>
</tr>
<tr>
<td>Average TP I/O Write</td>
<td>Average number of bytes written to the terminal during task processing</td>
</tr>
</tbody>
</table>

### PMARPT07 Billing group detail report

PMARPT07 contains one detail line for every execution of each task, or each CA ADS dialog, executed under the billing group code. The billing group code is shown at the top of the report. The detail report is followed by a summary recap of billing group activity.

### Sample report

<table>
<thead>
<tr>
<th>BILLING GROUP: UNDEFINED</th>
</tr>
</thead>
<tbody>
<tr>
<td>TASK CODE</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>QUED</td>
</tr>
<tr>
<td>CLOD</td>
</tr>
<tr>
<td>FACTOTUM</td>
</tr>
<tr>
<td>S</td>
</tr>
<tr>
<td>FACTOTUM</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>FACTOTUM</td>
</tr>
<tr>
<td>DOMT</td>
</tr>
<tr>
<td>FACTOTUM</td>
</tr>
<tr>
<td>DOMT</td>
</tr>
<tr>
<td>FACTOTUM</td>
</tr>
<tr>
<td>DOMT</td>
</tr>
<tr>
<td>FACTOTUM</td>
</tr>
</tbody>
</table>
### PMARPT07 fields

The fields in PMARPT07 are identical to those in report 01. See PMARPT01: Task detail report (see page) for detailed field descriptions.

Descriptions of the additional Summary Recap fields at the end of PMARPT07 are shown in the following table. Interpret the word task to mean either task or CA ADS dialog, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Number of Tasks</td>
<td>Total tasks executed by the user</td>
</tr>
<tr>
<td>Total Physical I/Os</td>
<td>Total number of physical I/Os performed by the tasks</td>
</tr>
<tr>
<td>Database Calls</td>
<td>Total number of database calls issued by the tasks</td>
</tr>
<tr>
<td>Total Abnormal Terminations</td>
<td>Total number of tasks that terminated abnormally</td>
</tr>
<tr>
<td>Total CPU Time</td>
<td>Total CPU time for the above tasks (hh:mm:ss)</td>
</tr>
<tr>
<td>Average Storage Used</td>
<td>Average number of bytes of variable storage from a DC/UCF storage pool used by the tasks during execution</td>
</tr>
<tr>
<td>Average Storage Kept</td>
<td>Average number of bytes of variable storage kept by the tasks at task termination for pseudo-conversational processing; this does not include relocated storage for CA ADS</td>
</tr>
</tbody>
</table>
PMARPT08 Billing group summary report

PMARPT08 contains one summary line for each different task, or each different CA ADS dialog, executed under the billing group. The billing group is shown at the top of the report. The report is followed by a summary recap of billing group activity.

Sample report

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average TP I</td>
<td>Average number of bytes read from the terminal during task processing</td>
</tr>
<tr>
<td>Length</td>
<td></td>
</tr>
<tr>
<td>Average TP I</td>
<td>Average number of bytes written to the terminal during task processing</td>
</tr>
<tr>
<td>Length</td>
<td></td>
</tr>
</tbody>
</table>

PMARPT08 Billing group summary report

REPORT NO. 08 CA DMS/PM nn.n volser <---------- YOUR COMPANY NAME ----------> DATA FROM: mm/dd/yy

Task Code   Version   Times  Type  Times  Active  Kept  CPU   Wait  TP   TP   OF  OF  OF  OF  OF
Task   Ver   Num    Task   Num    Storage  Storage  Time  Time  READ  WRITE  DBCLS  LVL  DBLVL  DBFS
B  12 AAS/0  30720  17190  .0153  .3391  6  727  7  92  0  0  1
BYE  1 ASSEM  9856   0      .0051  .5961  0  0    2  13  2  2  2
CAP  7 AAS/0  41179  13769  .0240  .5578  7  761  11  152 0  0  1
CAR  13 AAS/0 41945  16423  .0196  .3977  6  949  18  115 0  0  1
CAS  11 AAS/0 33218  11537  .0495  .5644  5  1023 11  166 0  0  1
CASCAS 11 AAS/0 32576  11776  .0262  .7107  5  788  14  145 0  0  1
CCC  7 AAS/0  54254  18341  .0229  .3467  7  755  18  194 0  0  1
CGL  13 AAS/0 33398  17388  .0196  .2833  7  794  12  107 0  0  1
CLIST 1 ASSEM 18176  640   .0096  2.0227 20  0   4  29 2  2  2
CLOD 1 ASSEM 25088  0      .0247  2.2841 0  0    5  221 0  0  1
COE  13 AAS/0 24930  17290  .0126  .1365  5  837  4  58  1  0  1
CPFR 11 AAS/0 42519  16547  .0198  .3653  6  838 18  125 0  0  1
CPRO 11 AAS/0 45452  16593  .0172  .1852  6  896  7  107 0  0  1
CPMS  12 AAS/0 36597  17739  .0142  .1443  5  887  4  74  1  0  1
CPVR  9 AAS/0  35584  15986  .0142  .1114  7  864  4  79  1  0  1
DCMT 9 AAS/0 13184  15013  .0063  .4919  19  162  2  11  -1 -1 1
DCUT 8 AAS/0 19752  3552   .0031  .1663  22  0   2  6 -1 -1 -1
DFACTTUM 78 ASSEM 1971  9849   .0095  .0444  29  84  0  1  -1 -1 -1
IDO  30 ASSEM 2  56588  68629 1.0281  43.6643 35 839 1234 1124 2  2  2
IN  13 AAS/0 29588  15931  .0118  .1399  5  798  4  56  0  0  1
MPS  10 AAS/0 52659  15962  .2224  .3438  9  1013 8  142 0  0  1
MRP  10 AAS/0 37914  15616  .0166  .1704  6  819  5  86  0  0  1
OLP  6 AAS/0 13568  11392  .5986  13.5025 10 1147 745 3699 2  2  2
OLQ  25 ASSEM 2  61322  8443 1.3810  7.3972 57 736 201 230 2  2  2
OPER 9 AAS/0 12064  15881  .0958  519.6225 38 7363 2  7  -1 -1 -1
PMAM 1 ASSEM 16512  6144  .0133  .7543  10  683  7  17  0  0  1
PMIM 9 ASSEM 19172  15317  .0191  .4763  10 1153  4  28  -1 -1 -1
PMIM 2 ASSEM 34368  5760  .0321  52.2188 36 2724 7  54  -1 -1 -1
PMNDRVR 1 187 ASSEM 27220 35579  .0043  .2874  3  1714 0  11  -1 -1 -1
QUED 1 ASSEM 6656  0  1.0723  43.5960 0  0  2008 39 2  2  2
RHDCSRTS 4 ASSEM 11776  0  .0593  2.3894 0  0  8  32 2  2  2
S  3 ASSEM 18752  0  .0107  .6712  20  0  11  38 0  0  1
SDEL 1 UNDEF 19328  0  .0070  2.4336 0  0  2  50 0  0  1
SFC 11 AAS/0 33699  17081  .0159  .2957  4  819  7  97  0  0  1
SIGMON 1 ASSEM 11520  768  .0159  3.7168 46 96  14  69  -1 -1 -1
USGADFL 1 UNDEF 17536  0  .0179  44.3700 0  0  2  44 0  0  1
USGAFIX 1 UNDEF 17536  0  .0157  2.7638 0  0  1  44 0  0  1

*** SUMMARY RECAP *** FOR BILLING GROUP: UNDEFINED FOR DC SYSTEM VERSION #: 56 ON 6/19/99
TOTAL NUMBER OF TASKS: 573
AVERAGE STORAGE USED: 28735
TOTAL PHYSICAL I/O: 50331
AVERAGE STORAGE KEPT: 23964
TOTAL DATABASE CALLS: 87529
AVERAGE TP I/O READ LENGTH: 14
TOTAL ABNORMAL TERMINATIONS: 4
AVERAGE TP I/O WRITE LENGTH: 1038
TOTAL CPU TIME (MM:SS): 0:80:46

Sample report followed by a summary recap of billing group activity.
PMARPT08 fields

The fields in PMARPT08 are identical to those in report 02. See PMARPT02: Task summary report (see page) for detailed field descriptions.

Descriptions of the additional Summary Recap fields at the end of PMARPT08 are shown in the following table. Interpret the word task to mean either task or CA ADS dialog, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Number of Tasks</td>
<td>Total tasks executed by the user</td>
</tr>
<tr>
<td>Total Physical I/O</td>
<td>Total number of physical I/Os performed by the tasks</td>
</tr>
<tr>
<td>Total Database Calls</td>
<td>Total number of database calls issued by the tasks</td>
</tr>
<tr>
<td>Total Abnormal Terminations</td>
<td>Total number of tasks that terminated abnormally</td>
</tr>
<tr>
<td>Total CPU Time</td>
<td>Total CPU time for the above tasks (hh:mm:ss)</td>
</tr>
<tr>
<td>Average Storage Used</td>
<td>Average number of bytes of variable storage from a DC/UCF storage pool used by the tasks during execution</td>
</tr>
<tr>
<td>Average Storage Kept</td>
<td>Average number of bytes of variable storage kept by the tasks at task termination for pseudo-conversational processing; this does not include relocated storage for CA ADS</td>
</tr>
<tr>
<td>Average TP I/O Read Length</td>
<td>Average number of bytes read from the terminal during task processing</td>
</tr>
<tr>
<td>Average TP I/O Write Length</td>
<td>Average number of bytes written to the terminal during task processing</td>
</tr>
</tbody>
</table>

PMARPT09 Abnormal termination detail report

PMARPT09 contains one detail line for every execution of each task or CA ADS dialog that terminated abnormally.

Sample report
PMAEPT09 fields

The fields in PMAEPT09 are similar to those in Report 01. See PMAEPT01: Task detail report (see page) for detailed field information. The CC (completion code), TP Read Length, and TP Write Length fields shown in Report 01, do not apply here.

The following table describes fields unique to PMAEPT09. Interpret the word task to mean either task or CA ADS dialog, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abend Code</td>
<td>Four-character task abend code for the task. A task abend can be issued either from within the task or from the DC-UCF system. This is only applicable if a task abend caused the abnormal termination.</td>
</tr>
<tr>
<td>Msg Id</td>
<td>Six-character ID of the message that indicates the abnormal status of the executing task.</td>
</tr>
<tr>
<td>Sevr Code</td>
<td>Severity code associated with the message ID, in the range 0 through 9.</td>
</tr>
</tbody>
</table>

PMAEPT10 Abnormal termination summary report

PMAEPT10 contains one summary line for each task or CA ADS dialog that terminated abnormally.

The fields in PMAEPT10 are identical to those in Report 02. See PMAEPT02: Task summary report (see page) for detailed field information.

Sample report

<table>
<thead>
<tr>
<th>TASK CODE</th>
<th>VER</th>
<th>NUM TIMES</th>
<th>TASK TYPE</th>
<th>CPU</th>
<th>WAIT</th>
<th>TP READ</th>
<th>TP WRITE</th>
<th>NUM OF</th>
<th>NUM OF</th>
<th>NUM OF</th>
<th>NUM OF</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDD</td>
<td>0</td>
<td>30 ASSEM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OLQ</td>
<td>0</td>
<td>25 ASSEM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PMAEPT11 LTERM detail report

PMAEPT11 contains one detail line for each task or CA ADS dialog invoked from the logical terminal identified at the top of the report.

The fields in PMAEPT11 are identical to those in Report 01. See PMAEPT01: Task detail report (see page) for detailed field information.

Sample report
PMARPT12 LTERM summary report

PMARPT12 contains one summary line for each task or CA ADS dialog invoked from the logical terminal. The summary line is identified at the top of the report.

The fields in PMARPT12 are identical to those in Report 02. See PMARPT02: Task summary report (see page) for detailed field information.

**Sample report**

<table>
<thead>
<tr>
<th>TASK</th>
<th>NUM TASK</th>
<th>CODE</th>
<th>EXEC TIME</th>
<th>STORAGE</th>
<th>CPU TIME</th>
<th>WAIT TIME</th>
<th>TIME</th>
<th>READ TIME</th>
<th>WRITE TIME</th>
<th>DBCLS</th>
<th>LVLS</th>
<th>DBLVLS</th>
<th>BUFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>1 ASSEM</td>
<td>0</td>
<td>7:54:11</td>
<td>9856</td>
<td>0</td>
<td>0.0951</td>
<td>0.5961</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>13</td>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>BYE</td>
<td>0 ASSEM</td>
<td>0</td>
<td>0.0303</td>
<td>9856</td>
<td>0</td>
<td>0.8414</td>
<td>0.0173</td>
<td>17</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>17</td>
<td>0</td>
</tr>
<tr>
<td>DCMT</td>
<td>0 ASSEM</td>
<td>0</td>
<td>0.0404</td>
<td>10624</td>
<td>0</td>
<td>0.3184</td>
<td>0.7192</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>13</td>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>FACTOTUM</td>
<td>1 ASSEM</td>
<td>17</td>
<td>0.0007</td>
<td>1114</td>
<td>30232</td>
<td>0.0113</td>
<td>0.1713</td>
<td>131</td>
<td>124</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>IDD</td>
<td>0 ASSEM</td>
<td>27</td>
<td>57864</td>
<td>27518</td>
<td>1.1287</td>
<td>47.4342</td>
<td>36.873</td>
<td>1355</td>
<td>1229</td>
<td>0</td>
<td>0</td>
<td>1355</td>
<td>1229</td>
</tr>
<tr>
<td>OPER</td>
<td>0 ASSEM</td>
<td>2</td>
<td>16000</td>
<td>47296</td>
<td>0.3883</td>
<td>150.8583</td>
<td>86.5178</td>
<td>9</td>
<td>20</td>
<td>9</td>
<td>20</td>
<td>9</td>
<td>20</td>
</tr>
<tr>
<td>PRM</td>
<td>0 ASSEM</td>
<td>1</td>
<td>16512</td>
<td>6144</td>
<td>0.1313</td>
<td>0.7543</td>
<td>10.683</td>
<td>7</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>17</td>
</tr>
<tr>
<td>PRM</td>
<td>0 ASSEM</td>
<td>2</td>
<td>17408</td>
<td>10624</td>
<td>0.0708</td>
<td>0.1799</td>
<td>18.1135</td>
<td>4</td>
<td>14</td>
<td>4</td>
<td>14</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>PMR</td>
<td>0 ASSEM</td>
<td>1</td>
<td>26112</td>
<td>512</td>
<td>0.1127</td>
<td>0.2524</td>
<td>0.2795</td>
<td>11326</td>
<td>11326</td>
<td>0</td>
<td>0</td>
<td>11326</td>
<td>11326</td>
</tr>
<tr>
<td>PMNR</td>
<td>1 ASSEM</td>
<td>145</td>
<td>28458</td>
<td>40575</td>
<td>0.0388</td>
<td>0.1878</td>
<td>0.1878</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S</td>
<td>0 ASSEM</td>
<td>1</td>
<td>10752</td>
<td>768</td>
<td>0.0110</td>
<td>1.2497</td>
<td>0.1249</td>
<td>23</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
</tr>
</tbody>
</table>

**PMARPT12 LTERM summary report**

PMARPT12 contains one summary line for each task or CA ADS dialog invoked from the logical terminal. The summary line is identified at the top of the report.

The fields in PMARPT12 are identical to those in Report 02. See PMARPT02: Task summary report (see page) for detailed field information.
PMARPT13 PTERM detail report

PMARPT13 contains one detail line for each task or CA ADS dialog invoked from the physical terminal. The detail line is identified at the top of the report.

The fields in PMARPT13 are identical to those in Report 01. See PMARPT01: Task detail report (see page ) for detailed field information.

Sample report

PMARPT14 PTERM summary report

PMARPT14 contains one summary line for each task or CA ADS dialog invoked from the physical terminal. The summary line is identified at the top of the report.

The fields in PMARPT14 are identical to those in Report 02. See PMARPT02: Task summary report (see page ) for detailed field information.

Sample report
**PMARPT15 System detail report**

PMARPT15 contains one detail line for each DC/UCF system internal or driver task (DBRC, MASTER, line driver, print driver, and so on).

The fields in PMARPT15 are identical to those in Report 01. See PMARPT01: Task detail report for detailed field information.

**Sample report**

**PMARPT16 System summary report**

PMARPT16 contains one summary line for each DC/UCF system internal or driver task (DBRC, MASTER, line driver, print driver, and so on).
The fields in PMARPT16 are identical to those in Report 02. See PMARPT02: Task summary report (see page ) for detailed field information.

Sample report

PMARPT17 Database detail report

PMARPT17 contains one line for each task showing database statistics.

Sample report
### PMARPT17 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Code</td>
<td>Task code or CA ADS dialog name</td>
</tr>
<tr>
<td>Ver Num</td>
<td>Version number of the level-1 program executed for the task</td>
</tr>
<tr>
<td>Task Num</td>
<td>Sequential number assigned to the task at task initiation (also known as the task ID)</td>
</tr>
<tr>
<td>Task Type</td>
<td>Source language for the level-1 program for the task (ERUS for an external request unit)</td>
</tr>
<tr>
<td>Recs Curr of R/U</td>
<td>Number of records that became current of run unit as a result of FIND, STORE, or OBTAIN requests</td>
</tr>
<tr>
<td>Recs Rqstd</td>
<td>Number of records retrieved from the database as a result of processing requests issued by the run unit</td>
</tr>
<tr>
<td>Pages Rqstd</td>
<td>Number of pages requested by the DBMS for the run unit</td>
</tr>
<tr>
<td>Pages Read</td>
<td>Number of pages physically read on behalf of the run unit</td>
</tr>
<tr>
<td>Pages Written</td>
<td>Number of physical writes that occurred while the task was in control</td>
</tr>
<tr>
<td>CALC Recs No Oflow</td>
<td>Number of records stored using the CALC location mode that were stored on the target page</td>
</tr>
<tr>
<td>CALC Recs With Oflow</td>
<td>Number of records stored using the CALC location mode that were stored on a page other than the target page</td>
</tr>
<tr>
<td>VIA Recs No Oflow</td>
<td>Number of records stored using the VIA location mode that were stored on the target page</td>
</tr>
<tr>
<td>VIA Recs With Oflow</td>
<td>Number of records stored using the VIA location mode that were stored on a page other than the target page</td>
</tr>
<tr>
<td>Frags Stored</td>
<td>Number of record fragments stored</td>
</tr>
<tr>
<td>Select Locks Of R/U</td>
<td>Number of select locks held by the task</td>
</tr>
<tr>
<td>Update Locks Of R/U</td>
<td>Number of update locks held by the task</td>
</tr>
</tbody>
</table>
PMARPT18 Database summary report

PMARPT18 contains one summary line for each task type.

The fields in this report are the averages, sorted by task, of the fields in PMARPT17. See PMARPT17: Database detail report (see page 2) for detailed field information.

Sample report

REPORT NO. 18
CA INC.
DATA FROM: mm/dd/yy
PAGE 1

PMARPT19 DC statistics detail report

PMARPT19 contains one line for each task showing DC/UCF system statistics.
PMARPT19 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Code</td>
<td>Task code or CA ADS dialog name</td>
</tr>
<tr>
<td>Ver Num</td>
<td>Version number of the level-1 program executed for the task or CA ADS dialog</td>
</tr>
<tr>
<td>Task Num</td>
<td>Sequential number assigned to the task at task initiation (also known as the task ID)</td>
</tr>
</tbody>
</table>

Sample report

```
REPORT NO. 19  CA, INC.  CA IDMS/PM nn.n  volser  DC SYSTEM VERSION #: 71

DC STATISTICS DETAIL REPORT

TASK CODE  VER NUM  TASK DC PRYT PGM CALLED PGMG LOADED  PGMG SPACE GET STORAGE FREE STORAGE GET STORAGE PUT DELETE GET PUT DELETE

QUED 0 18 250 1 1 0 14 11 5760 0 0 0 0 0 0 0
CLOD 0 19 250 6 2 6704 44 41 13056 0 0 0 0 0 0 0
FACTOTUM 1 20 251 0 0 0 5 2 0 0 0 0 0 0 0
FACTOTUM 1 21 251 0 0 0 3 1 0 0 0 0 0 0 0
FACTOTUM 1 22 251 1 1 0 16 13 9600 0 0 0 0 0 0 0
FACTOTUM 1 23 251 0 0 0 3 3 256 512 0 0 0 1 0 0
FACTOTUM 1 24 251 0 0 0 3 3 255 512 0 0 0 1 0 0
DCMT 0 26 225 2 2 24360 14 11 9600 0 0 0 0 0 0 0
FACTOTUM 1 27 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 28 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 29 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 30 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
DCMT 0 31 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 32 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
CLOD 0 33 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 34 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 35 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
DCMT 0 36 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 37 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
DCMT 0 38 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 39 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 40 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
CLOD 0 41 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 42 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
CLOD 0 43 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 44 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 45 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
DCMT 0 46 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 47 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 48 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 49 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
DCMT 0 50 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 51 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 52 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
DCMT 0 53 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 54 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 55 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
DCMT 0 56 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 57 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
DCMT 0 58 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 59 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 60 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
DCMT 0 61 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 62 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
DCMT 0 63 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 64 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
DCMT 0 65 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0
FACTOTUM 1 66 251 3 0 17672 7 5 12288 768 0 0 0 0 0 0

CA IDMS - 19.0

16-Jan-2018  436/898
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC Prty</td>
<td>DC/UCF priority assigned to the task</td>
</tr>
<tr>
<td>Pgms Called</td>
<td>Number of programs called by the task; includes: LINKs XCTLs Programs called by the system on behalf of the task</td>
</tr>
<tr>
<td>Pgms Loaded</td>
<td>Number of programs called that were not present in the program pool and that needed to be loaded</td>
</tr>
<tr>
<td>Pgm Space Used</td>
<td>Amount of program-pool space used by the task</td>
</tr>
<tr>
<td>Get Storage Rqsts</td>
<td>Number of GET STORAGE (#GETSTG) requests issued by or on behalf of the task</td>
</tr>
<tr>
<td>Free Storage Rqsts</td>
<td>Number of FREE STORAGE (#FREESTG) requests issued by or on behalf of the task</td>
</tr>
<tr>
<td>Storage Active</td>
<td>High-water mark of storage used by the task; includes all types of storage</td>
</tr>
<tr>
<td>Storage Kept</td>
<td>Amount of USER KEPT or SHARED KEPT storage maintained by the DC/UCF system on behalf of the task; such storage can be held across a pseudo-converse; this does not include relocated storage for CA ADS</td>
</tr>
<tr>
<td>Get Scratch Rqsts</td>
<td>Number of GET SCRATCH requests issued by or on behalf of the task</td>
</tr>
<tr>
<td>Put Scratch Rqsts</td>
<td>Number of PUT SCRATCH requests issued by or on behalf of the task</td>
</tr>
<tr>
<td>Delete Scratch Rqsts</td>
<td>Number of DELETE SCRATCH requests issued by or on behalf of the task</td>
</tr>
<tr>
<td>Get Queue Rqsts</td>
<td>Number of GET QUEUE requests issued by or on behalf of the task</td>
</tr>
<tr>
<td>Put Queue Rqsts</td>
<td>Number of PUT QUEUE requests issued by or on behalf of the task</td>
</tr>
<tr>
<td>Delete Queue Rqsts</td>
<td>Number of DELETE QUEUE requests issued by or on behalf of the task</td>
</tr>
</tbody>
</table>

**PMARPT20 DC statistics summary report**

PMARPT20 contains one line for each task showing DC/UCF system statistics.

The fields contained in this report are averages, by task and task priority, for the fields in PMARPT19. See **PMARPT19: DC statistics detail report (see page)** for detailed field information.
### PMARPT21 Batch job accounting summary report

PMARPT21 contains one summary line of batch job accounting information for each batch ERUS task reported.

The batch job accounting information on a single report line is associated with an executed task. The TASK NUM for that task can be used to cross-reference to Report 01 to see additional task statistics.

Sample report

---

**16-Jan-2018**

PMARPT21 Batch job accounting summary report

PMARPT21 contains one summary line of batch job accounting information for each batch ERUS task reported.

The batch job accounting information on a single report line is associated with an executed task. The TASK NUM for that task can be used to cross-reference to Report 01 to see additional task statistics.

Sample report

---

**16-Jan-2018**

PMARPT21 Batch job accounting summary report

PMARPT21 contains one summary line of batch job accounting information for each batch ERUS task reported.

The batch job accounting information on a single report line is associated with an executed task. The TASK NUM for that task can be used to cross-reference to Report 01 to see additional task statistics.

Sample report

---

**16-Jan-2018**

PMARPT21 Batch job accounting summary report

PMARPT21 contains one summary line of batch job accounting information for each batch ERUS task reported.

The batch job accounting information on a single report line is associated with an executed task. The TASK NUM for that task can be used to cross-reference to Report 01 to see additional task statistics.

Sample report

---

**16-Jan-2018**

PMARPT21 Batch job accounting summary report

PMARPT21 contains one summary line of batch job accounting information for each batch ERUS task reported.

The batch job accounting information on a single report line is associated with an executed task. The TASK NUM for that task can be used to cross-reference to Report 01 to see additional task statistics.

Sample report

---

**16-Jan-2018**

PMARPT21 Batch job accounting summary report

PMARPT21 contains one summary line of batch job accounting information for each batch ERUS task reported.

The batch job accounting information on a single report line is associated with an executed task. The TASK NUM for that task can be used to cross-reference to Report 01 to see additional task statistics.

Sample report

---

**16-Jan-2018**

PMARPT21 Batch job accounting summary report

PMARPT21 contains one summary line of batch job accounting information for each batch ERUS task reported.

The batch job accounting information on a single report line is associated with an executed task. The TASK NUM for that task can be used to cross-reference to Report 01 to see additional task statistics.

Sample report

---

**16-Jan-2018**

PMARPT21 Batch job accounting summary report

PMARPT21 contains one summary line of batch job accounting information for each batch ERUS task reported.

The batch job accounting information on a single report line is associated with an executed task. The TASK NUM for that task can be used to cross-reference to Report 01 to see additional task statistics.

Sample report

---

**16-Jan-2018**

PMARPT21 Batch job accounting summary report

PMARPT21 contains one summary line of batch job accounting information for each batch ERUS task reported.

The batch job accounting information on a single report line is associated with an executed task. The TASK NUM for that task can be used to cross-reference to Report 01 to see additional task statistics.

Sample report

---

**16-Jan-2018**

PMARPT21 Batch job accounting summary report

PMARPT21 contains one summary line of batch job accounting information for each batch ERUS task reported.

The batch job accounting information on a single report line is associated with an executed task. The TASK NUM for that task can be used to cross-reference to Report 01 to see additional task statistics.

Sample report

---

**16-Jan-2018**

PMARPT21 Batch job accounting summary report

PMARPT21 contains one summary line of batch job accounting information for each batch ERUS task reported.

The batch job accounting information on a single report line is associated with an executed task. The TASK NUM for that task can be used to cross-reference to Report 01 to see additional task statistics.

Sample report

---

**16-Jan-2018**

PMARPT21 Batch job accounting summary report

PMARPT21 contains one summary line of batch job accounting information for each batch ERUS task reported.

The batch job accounting information on a single report line is associated with an executed task. The TASK NUM for that task can be used to cross-reference to Report 01 to see additional task statistics.

Sample report

---

**16-Jan-2018**

PMARPT21 Batch job accounting summary report

PMARPT21 contains one summary line of batch job accounting information for each batch ERUS task reported.

The batch job accounting information on a single report line is associated with an executed task. The TASK NUM for that task can be used to cross-reference to Report 01 to see additional task statistics.

Sample report

---

**16-Jan-2018**

PMARPT21 Batch job accounting summary report

PMARPT21 contains one summary line of batch job accounting information for each batch ERUS task reported.

The batch job accounting information on a single report line is associated with an executed task. The TASK NUM for that task can be used to cross-reference to Report 01 to see additional task statistics.

Sample report
### PMARPT21 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Code</td>
<td>Task code for batch ERUS</td>
</tr>
<tr>
<td>Task Num</td>
<td>Sequential number assigned to the task at task initiation (also known as the task ID)</td>
</tr>
<tr>
<td>Task Start Date</td>
<td>Date the task was initiated (mm/dd/yy)</td>
</tr>
<tr>
<td>Task Start Time</td>
<td>Time the task was initiated (hh:mm:ss)</td>
</tr>
<tr>
<td>Task End Date</td>
<td>Date the task was ended (mm/dd/yy)</td>
</tr>
<tr>
<td>Task End Time</td>
<td>Time the task was ended (hh:mm:ss)</td>
</tr>
<tr>
<td>Jobname</td>
<td>Specifies the name of the job</td>
</tr>
<tr>
<td>Job Number</td>
<td>Specifies the job ID within the address space</td>
</tr>
<tr>
<td>Accounting Info</td>
<td>Account number, plus any other accounting information that was specified</td>
</tr>
</tbody>
</table>

### PMARPT31 Task wait summary report

PMARPT31 contains wait information for each task execution.

#### Sample report

<table>
<thead>
<tr>
<th>Task Code</th>
<th>Task Start Date</th>
<th>Task Start Time</th>
<th>Task End Date</th>
<th>Task End Time</th>
<th>Jobname</th>
<th>Job Number</th>
<th>Accounting Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDMSDDDL</td>
<td>8/21/17</td>
<td>09:33:07</td>
<td>8/21/17</td>
<td>09:33:07</td>
<td>JOB00619</td>
<td>IDMSDDDL</td>
<td></td>
</tr>
<tr>
<td>IDMSBCF</td>
<td>8/22/17</td>
<td>10:17:16</td>
<td>8/22/17</td>
<td>10:17:16</td>
<td>JOB00634</td>
<td>IDMSBCF</td>
<td></td>
</tr>
<tr>
<td>EMPLOAD1</td>
<td>8/24/17</td>
<td>16:24:20</td>
<td>8/24/17</td>
<td>16:24:20</td>
<td>EMPLOAD1</td>
<td>EMPLOAD1</td>
<td></td>
</tr>
<tr>
<td>IDMSDDDL</td>
<td>8/27/17</td>
<td>08:47:31</td>
<td>8/21/17</td>
<td>08:47:32</td>
<td>JOB00793</td>
<td>IDMSDDDL</td>
<td></td>
</tr>
</tbody>
</table>

#### PMARPT31 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Number of CPU units used for task execution</td>
</tr>
<tr>
<td>DBIO</td>
<td>Number of DBIO units used for task execution</td>
</tr>
<tr>
<td>AVG DBIO</td>
<td>Average time spent by DBIO units for task execution</td>
</tr>
<tr>
<td>I/O</td>
<td>Number of I/O requests used for task execution</td>
</tr>
<tr>
<td>AVG I/O</td>
<td>Average time spent by I/O requests for task execution</td>
</tr>
<tr>
<td>OTHER</td>
<td>Number of other units used for task execution</td>
</tr>
<tr>
<td>AVG OTHER</td>
<td>Average time spent by other units for task execution</td>
</tr>
</tbody>
</table>
### PMARPT31 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Code</td>
<td>Task code or CA ADS dialog name</td>
</tr>
<tr>
<td>Task Num</td>
<td>Sequential number assigned to the task at task initiation (also known as the task ID)</td>
</tr>
<tr>
<td>Start Time</td>
<td>Time the task was initiated (hh:mm:ss)</td>
</tr>
<tr>
<td>CPU Time</td>
<td>Total CPU time for the task (ss.ssss)</td>
</tr>
<tr>
<td>DBIO Waits</td>
<td>Number of waits for database reads and writes</td>
</tr>
<tr>
<td>DBIO Wait Time</td>
<td>Amount of time spent waiting for database reads and writes (ssss.ttt)</td>
</tr>
<tr>
<td>Avg DBIO Wait Time</td>
<td>Average time spent waiting for a database read or write (ssss.tttt)</td>
</tr>
<tr>
<td>Othr I/O Waits</td>
<td>Number of waits for I/O other than database reads and writes; typically journal I/O</td>
</tr>
<tr>
<td>Othr I/O Wait Time</td>
<td>Amount of time spent waiting for I/O other than database reads and writes (ssss.ttt)</td>
</tr>
<tr>
<td></td>
<td>Average time spent waiting for I/O other than database reads and writes (ssss.tttt)</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Avg Othr I/O</td>
<td>Time Number of waits for resources other than I/O; should be investigated further using PMARPT36</td>
</tr>
<tr>
<td>Othr Waits</td>
<td>Number of waits for resources other than I/O; should be investigated further using PMARPT36</td>
</tr>
<tr>
<td>Othr Wait Time</td>
<td>Amount of time spent waiting for resources other than I/O (ssss.tttt)</td>
</tr>
<tr>
<td>Avg Othr Wait Time</td>
<td>Average amount of time spent waiting for resources other than I/O (ssss.tttt)</td>
</tr>
<tr>
<td>Total Waits</td>
<td>Total of all waits</td>
</tr>
<tr>
<td>Total Wait Time</td>
<td>Total amount of time spent waiting (ssss.tttt)</td>
</tr>
</tbody>
</table>

**PMARPT36 Task wait detail report**

PMARPT36 contains one page of detailed wait statistics per task execution.

⚠️ **Note:** To minimize output, always run this report with explicit selection criteria.

---

**Sample report**

<table>
<thead>
<tr>
<th>REPORT NO. 36</th>
<th>CA, INC.</th>
<th>mm/dd/yy PAGE</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA IDMS/PM mm.n volser TASK=WAIT DETAIL REPORT DC SYSTEM VERSION #: 72 &lt;---------------- YOUR COMPANY NAME --------------&gt; DATA FROM: mm/dd/yy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TASK: DBCRUPD TASKID: 277 START TIME: 5:36:56</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WAITS</td>
<td>WAIT TIME</td>
<td>AVG TIME</td>
<td>HIGHEST TIME</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>DBIO READ</td>
<td>.2762</td>
<td>.0276</td>
<td>.0956</td>
</tr>
<tr>
<td>JOURNAL READ</td>
<td>.5240</td>
<td>.0066</td>
<td>.0126</td>
</tr>
<tr>
<td>SCRATCH READ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOG READ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBIO WRITE</td>
<td>.0009</td>
<td>.0009</td>
<td>.0009</td>
</tr>
<tr>
<td>JOURNAL WRITE</td>
<td>.0140</td>
<td>.0079</td>
<td>.0122</td>
</tr>
<tr>
<td>SCRATCH WRITE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOG WRITE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOS PRIOR IO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOURNAL BUFFER</td>
<td>.3338</td>
<td>.1113</td>
<td>.2613</td>
</tr>
<tr>
<td>SCR SINGLE THRD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOG SINGLE THRD</td>
<td>.0053</td>
<td>.0053</td>
<td>.0053</td>
</tr>
<tr>
<td>DB KEY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOADER SNGL THRD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QUEUE WRITE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXTERNAL RU</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AREA ACCESS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDS WRITES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TP READS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OTHER EXTRNL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

16-Jan-2018 441/898
### PMARPT36 fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Task code or CA ADS dialog name</td>
</tr>
<tr>
<td>Taskid</td>
<td>Sequential number assigned to the task at task initiation (also known as the task ID)</td>
</tr>
<tr>
<td>Start Time</td>
<td>Time the task was initiated (hh:mm:ss)</td>
</tr>
<tr>
<td>DBIO Read</td>
<td>Waits, wait time, average wait time, and highest wait time for database reads performed by or on behalf of the task</td>
</tr>
<tr>
<td>Journal Read</td>
<td>Waits, wait time, average wait time, and highest wait time for journal reads performed by or on behalf of the task (usually for rollback)</td>
</tr>
<tr>
<td>Scratch Read</td>
<td>Waits, wait time, average wait time, and highest wait time for scratch area reads performed by or on behalf of the task</td>
</tr>
<tr>
<td>Log Read</td>
<td>Waits, wait time, average wait time, and highest wait time for log area reads performed by or on behalf of the task</td>
</tr>
<tr>
<td>DBIO Write</td>
<td>Waits, wait time, average wait time, and highest wait time for database writes performed by or on behalf of the task</td>
</tr>
<tr>
<td>Journal Write</td>
<td>Waits, wait time, average wait time, and highest wait time for journal writes performed by or on behalf of the task</td>
</tr>
<tr>
<td>Scratch Write</td>
<td>Waits, wait time, average wait time, and highest wait time for scratch area writes performed by or on behalf of the task</td>
</tr>
<tr>
<td>Log Write</td>
<td>Waits, wait time, average wait time, and highest wait time for log area writes performed by or on behalf of the task</td>
</tr>
<tr>
<td>DOS Prior IO</td>
<td>Waits, wait time, average wait time, and highest wait time for prior I/O to complete (z/VSE only)</td>
</tr>
<tr>
<td>Journal Buffer</td>
<td>Waits, wait time, average wait time, and highest wait time because of full journal buffer (can indicate too few pages assigned to journal buffer)</td>
</tr>
<tr>
<td>Scratch Single Thread</td>
<td>Waits, wait time, average wait time, and highest wait time for the scratch manager to finish single-threaded processing</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Log</td>
<td>Waits, wait time, average wait time, and highest wait time for the log to finish single-threaded processing (if happening because of task or systems snaps, isolate and correct)</td>
</tr>
<tr>
<td>SingleThrd</td>
<td></td>
</tr>
<tr>
<td>DB</td>
<td>Waits, wait time, average wait time, and highest wait time for database buffers</td>
</tr>
<tr>
<td>Buffer</td>
<td></td>
</tr>
<tr>
<td>Pgm</td>
<td>Waits, wait time, average wait time, and highest wait time for programs to be loaded by or on behalf of the task</td>
</tr>
<tr>
<td>Loads</td>
<td></td>
</tr>
<tr>
<td>Queue</td>
<td>Waits, wait time, average wait time, and highest wait time for queue area reads performed by or on behalf of the task</td>
</tr>
<tr>
<td>Read</td>
<td></td>
</tr>
<tr>
<td>Log</td>
<td>Waits, wait time, average wait time, and highest wait time because log either full or being unloaded (frequent waits can indicate a problem with the log)</td>
</tr>
<tr>
<td>Full</td>
<td></td>
</tr>
<tr>
<td>DBkeys</td>
<td>Waits, wait time, average wait time, and highest wait time for db-key waits (consistently high numbers can indicate a problem). Details on DBkeys are shown just behind all wait types.</td>
</tr>
<tr>
<td>Loader</td>
<td>Waits, wait time, average wait time, and highest wait time for the loader to finish single-threaded processing</td>
</tr>
<tr>
<td>SnglThrd</td>
<td></td>
</tr>
<tr>
<td>Queue</td>
<td>Waits, wait time, average wait time, and highest wait time for queue writes issued by or on behalf of the task</td>
</tr>
<tr>
<td>Write</td>
<td></td>
</tr>
<tr>
<td>ExternRU</td>
<td>Waits, wait time, average wait time, and highest wait time for external request units</td>
</tr>
<tr>
<td>AreaAccess</td>
<td>Waits, wait time, average wait time, and highest wait time for access to an area (check usage mode in this and other concurrently running programs)</td>
</tr>
<tr>
<td>DDS</td>
<td>Waits, wait time, average wait time, and highest wait time for DDS I/O issued by or on behalf of the task</td>
</tr>
<tr>
<td>Writes</td>
<td></td>
</tr>
<tr>
<td>TP</td>
<td>Waits, wait time, average wait time, and highest wait time for terminal writes</td>
</tr>
<tr>
<td>Reads</td>
<td></td>
</tr>
<tr>
<td>OtherExtrnl</td>
<td>Waits, wait time, average wait time, and highest wait time for other meaningful external wait types, including the ICE, LRE, and user ECBs</td>
</tr>
<tr>
<td>StoragePool</td>
<td>Waits, wait time, average wait time, and highest wait time to acquire storage for the task</td>
</tr>
<tr>
<td>Checkuser</td>
<td>Waits, wait time, average wait time, and highest wait time for an available check user subtask (z/OS batch external run units only)</td>
</tr>
<tr>
<td>TPWrites</td>
<td></td>
</tr>
<tr>
<td>OtherIntrnl</td>
<td>Waits, wait time, average wait time, and highest wait time for other meaningful internal wait types, including ENQUEUE, DEQUEUE, LTE, PDE, and TCE</td>
</tr>
<tr>
<td>ProgramPool</td>
<td>Waits, wait time, average wait time, and highest wait time for access to a program or reentrant pool</td>
</tr>
<tr>
<td>DBGroUp</td>
<td>Waits, wait time, average wait time, and highest wait time for a request issued from a front-end CV that want to start a database session, to get an answer from a back-end CV that volunteers to service the request</td>
</tr>
<tr>
<td>SharedCache</td>
<td>Waits, wait time, average wait time, and highest wait time for a shared cache from the Coupling Facility</td>
</tr>
</tbody>
</table>
If waits on DBKeys occur, details about the DBKeys are shown.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>The DBKey's area name.</td>
</tr>
<tr>
<td>PgGrp</td>
<td>The DBKey's page group.</td>
</tr>
<tr>
<td>DBKey-Page</td>
<td>The DBKey's page number.</td>
</tr>
<tr>
<td>Line</td>
<td>The DBKey's line index.</td>
</tr>
<tr>
<td>Holder</td>
<td>The name of the holder of the DBKey lock.</td>
</tr>
<tr>
<td>Type</td>
<td>The type (TASK, LTE, or DDS) of the Holder field.</td>
</tr>
<tr>
<td>Hold.TskID</td>
<td>The task ID of the holder of the DBKey lock.</td>
</tr>
</tbody>
</table>

**PMARPT80 Load balancing report (by day and central version)**

PMARPT80 contains one set of summary lines for each hour reported, for both batch and online processing. You get one report for each day of processing. Each set of lines includes the following three statistics:

- The number of tasks or CA ADS dialogs executed
- The amount of CPU used
- The number of physical disk I/Os issued

The relative lengths of the lines for different time periods reflect the relative loads for those periods.

If the report-generation job includes task selection parameters, this report can not reflect load balancing information accurately. Only data for those tasks that meet the selection criteria are considered in preparing the report.

**Sample report**

<table>
<thead>
<tr>
<th>TIME</th>
<th>RESOURCE</th>
<th>VALUE</th>
<th>DATA FOR 06/19/10</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATCH 12:00</td>
<td>TASKS</td>
<td>0 .</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>.0000</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>I/O</td>
<td>0 .</td>
<td>.</td>
</tr>
<tr>
<td>ONLINE 12:00</td>
<td>TASKS</td>
<td>0 .</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>.0000</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>I/O</td>
<td>0 .</td>
<td>.</td>
</tr>
<tr>
<td>BATCH 13:00</td>
<td>TASKS</td>
<td>0 .</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>.0000</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>I/O</td>
<td>0 .</td>
<td>.</td>
</tr>
<tr>
<td>ONLINE 13:00</td>
<td>TASKS</td>
<td>0 .</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>.0000</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>I/O</td>
<td>0 .</td>
<td>.</td>
</tr>
<tr>
<td>BATCH 14:00</td>
<td>TASKS</td>
<td>0 .</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>.0000</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>I/O</td>
<td>0 .</td>
<td>.</td>
</tr>
<tr>
<td>ONLINE 14:00</td>
<td>TASKS</td>
<td>19 .TTT</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>1.1863</td>
<td>.C</td>
</tr>
<tr>
<td></td>
<td>I/O</td>
<td>2,053 .II</td>
<td>.</td>
</tr>
<tr>
<td>BATCH 15:00</td>
<td>TASKS</td>
<td>0 .</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>.0000</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>I/O</td>
<td>0 .</td>
<td>.</td>
</tr>
</tbody>
</table>
| ONLINE 15:00 | TASKS | 287 | .TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
The following table describes the fields in PMARPT80. Interpret the word "task" to mean either task or CA ADS dialog, as appropriate.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tasks</td>
<td>Representation of the number of tasks or CA ADS dialogs executed</td>
</tr>
<tr>
<td>CPU</td>
<td>Representation of the total CPU time for all tasks in seconds</td>
</tr>
<tr>
<td>I/O</td>
<td>Representation of the number of physical disk I/Os for all tasks</td>
</tr>
</tbody>
</table>

PMARPT81 Load balancing (by CV)

PMARPT81 contains one set of summary lines for each hour reported, for both batch and online processing, sorted by central version. All days are compressed into one 24-hour graph.

The statistics are identical to those for Report 80. See PMARPT80: Load balancing report (by day and central version) (see page) for detailed field information.

PMARPT82 Load balancing (All CVs)

PMARPT82 contains one set of summary lines for each hour reported, for both batch and online processing, for all central versions. All days are compressed into one 24-hour graph.

The statistics are identical to those for Report 80. See PMARPT80: Load balancing report (by day and central version) (see page) for detailed field information.

PMARPT90 Machine-readable copy

Statistics extracted by Report 00 are output to either tape or disk.

When you run PMARPT90, you must run it with PMARPT00. Additionally, you can use the following parameters with PMARPT90:

- CV NUMBER
- DATE FORMAT
- REPORT FROM/THRU
PMARPT97 Summary recap report

PMARPT97 contains summary statistics for all tasks and CA ADS dialogs reported. This report is sorted by day and by central version.

Sample report

<table>
<thead>
<tr>
<th>REPORT NO. 97</th>
<th>CA, INC.</th>
<th>mm/dd/yy PAGE</th>
<th>1</th>
</tr>
</thead>
</table>
| CA IDMS/PM nn.n | volser | SUMMARY RECAP REPORT | <--------- YOUR COMPANY NAME ---------->
| TOTALS FOR DC SYSTEM VERSION 56 ON 4/11/11 |
| 0 | 183 TOTAL TASKS | 37 DC/UCF | 24 FACTOTUM | 0 BATCH | 0 CICS |
| 0 | 0 TASKS NOT SELECTED | 0 DC/UCF | 0 FACTOTUM | 0 BATCH | 0 CICS |
| 0 | 183 TASKS SELECTED | 37 DC/UCF | 24 FACTOTUM | 0 BATCH | 0 CICS |

PMARPT97 fields

The following table describes the fields in PMARPT97. Interpret the word tasks to mean either task or CA ADS dialog, as appropriate.

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of tasks (total)</td>
<td>Total number of tasks executed</td>
</tr>
<tr>
<td>System information (total)</td>
<td>DC/UCF system information, including total and average CPU time</td>
</tr>
</tbody>
</table>
### PMARPT99 Input processing summary report

PMARPT99 provides the following information:

- **Task selection parameters**: For more information, see the [Request Reports (Interval Monitor)](see page 369).

- **Input parameter processing**

- **Input record processing statistics**:
  - Records read by PMARPT00
  - Records selected by PMARPT00
  - Records dropped by PMARPT00

  For example, this category includes the earliest record read and the latest record read.

### Sample report

#### Information

<table>
<thead>
<tr>
<th>Description</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of tasks selected and not selected.</td>
<td>Task selection information</td>
</tr>
<tr>
<td>DC/UCF system information, including total and average CPU time</td>
<td>System information (for selected tasks)</td>
</tr>
<tr>
<td>Daily totals and averages for all database and DC/UCF statistics</td>
<td>Database/data communications statistics (for selected tasks)</td>
</tr>
<tr>
<td>The total system mode CPU time for selected tasks consumed on a CP because zIIP is unavailable.</td>
<td>Total zIIP on CP CPU</td>
</tr>
<tr>
<td>The average system mode CPU time for selected tasks consumed on a CP because zIIP is unavailable.</td>
<td>Average zIIP on CP CPU</td>
</tr>
<tr>
<td>The total system mode CPU time for selected tasks consumed on a zIIP.</td>
<td>Total zIIP on zIIP CPU</td>
</tr>
<tr>
<td>The average system mode CPU time for selected tasks consumed on a zIIP.</td>
<td>Average zIIP on zIIP CPU</td>
</tr>
<tr>
<td>The total system mode CPU time for selected tasks consumed on a zIIP or CP eligible as SRB, including zIIP on CP and zIIP on zIIP.</td>
<td>Total Enclave CPU</td>
</tr>
<tr>
<td>The average system mode CPU time for selected tasks consumed on a zIIP or CP eligible as SRB, including zIIP on CP and zIIP on zIIP.</td>
<td>Average Enclave CPU</td>
</tr>
<tr>
<td>The total CPU time for selected tasks consumed on CP, including user mode and system mode CPU time.</td>
<td>Total TCB CPU</td>
</tr>
<tr>
<td>The average CPU time for selected tasks consumed on CP, including user mode and system mode CPU time.</td>
<td>Average TCB CPU</td>
</tr>
</tbody>
</table>
INPUT CARD PROCESSING

CARDS READ: 1
CARDS PROCESSED: 1
COMMENT CARDS: 0
CARD ERRORS: 0

INPUT RECORD PROCESSING STATISTICS

RECORDS READ BY PMARPT00

# STAT RECS READ: 4,467
# PMAM RECS READ: 4,179
# PMIM RECS READ: 288
EARLIEST REC READ: 07:54 ON 30/09/99 (99/274)
LATEST REC READ: 08:20 ON 30/09/99 (99/274)

BY RECORD TYPE
    TASK STATS 1,393
    TASK WAIT 0
    DBKEY 0

RECORDS SELECTED BY PMARPT00

# PMAM RECS SELECTED: 1,393
EARLIEST REC SELECTD: 07:54 ON 30/09/99 (99/274)
LATEST REC SELECTD: 08:20 ON 30/09/99 (99/274)

BY RECORD TYPE
    TASK STATS 1,393
    TASK WAIT 0
    DBKEY 0

RECORDS DROPPED BY PMARPT00

# PMAM RECS DROPPED: 0
# PMIM RECS DROPPED: 288

PROCESSING OF MULTIPART RECORDS

#PMTASDS SEQ# 1 1,393
#PMTASDS SEQ# 2 1,393
#PMTASDS SEQ# 3 1,393
#PMTAWS SEQ# 1 0
#PMTAWS SEQ# 2 0
Online Query (OLQ) Reports

In this section CA OLQ (Online Query) is a query tool and report writer that accesses data stored in a CA IDMS/DB database.

You can use CA OLQ to:

- **Query a CA IDMS/DB database.** For example, if you want to know the phone number of an employee, you can use CA OLQ to retrieve the phone number from the database.

- **Create reports.** For example, if you want to list the names, phone numbers, and sales quotas for a group of salespeople, you can use CA OLQ to create this report.

For more information, see the following topics:

- What are CA OLQ features? (see page 449)
- Sign on and Sign off (see page 451)
- How to Make a Report from a Table (see page 456)
- How to Make a Report From More Than One Table (see page 472)
- How to Format Your Report (see page 480)
- How to Create Report Totals and Subtotals (see page 510)
- How To Format Reports Containing Calculations (see page 528)
- How to Save a Set of Commands as a Qfile (see page 540)
- How to Make a Report from Database Records (see page 570)
- How to Save a Table From a Report (see page 581)
- How to Use CA OLQ in Batch Mode (see page 590)
- How to Print Your Report (see page 595)
- Introducing the OLQ SELECT statement (see page 597)
- Retrieving Information from a Table (see page 608)
- Summarizing Information (see page 623)
- Joining Tables or Database Records (see page 626)
- Sample Tables and Database (see page 641)
- Answers (see page 645)

What are CA OLQ features?

CA OLQ offers the following features:

- Menu facility (see page 450)
- CA OLQ command mode (see page 451)
- Two processing modes (see page 451)
- Two access modes (see page 451)
Menu facility

Using the CA OLQ menu facility you can learn how to create reports quickly. The menus are easy to use, and provide most of the power of syntax-driven CA OLQ.

Create reports

You can report on:

- SQL tables
- ASF tables
- Database records

Format your report

You can enhance the appearance of your report by:

- Sorting report rows
- Modifying column and page headers
- Specifying external pictures for columns
- Changing the spacing between columns
- Using code tables to translate data stored in the database into an alternative external representation

Perform computations

You can create the following types of report calculations:

- Computed columns
  - Break processing and group subtotals
  - Report totals
  - Calculations including aggregate and built-in functions

Save a report

Once you create a report, you can save it online for a specified period of time.

Create a table from a report

You can create a table by saving your CA OLQ report as a table. The rows and columns in the table correspond to those you have defined in the report.
Define predefined routines

You can save the set of CA OLQ commands used to create a report in a routine. This routine can be re-executed to create reports that reflect the changing nature of data in the database.

Print a report

You can route your CA OLQ report to output devices defined to your system.

Execute OLQ commands using the OLQ batch facility

You can use the CA OLQ batch facility to create reports that query very large amounts of data, or to execute during non-peak hours.

CA OLQ command mode

CA OLQ also offers syntax mode for very specific queries, or for people who prefer to use syntax.

For information on how to use CA OLQ command mode, refer to the CA OLQ Reference Section.

Two processing modes

You can create a report in either an online or a batch environment.

Two access modes

CA OLQ provides you with the ability to access both SQL and ASF tables, database and logical records, and sequential files (batch only). Setting the access switch to either idms or olq tells CA OLQ what kind of table you want to access.

Sign on and Sign off

Signing on

You sign on to CA OLQ from CA IDMS/DC or CA IDMS UCF. When CA IDMS/DC or CA IDMS UCF prompts you with ENTER NEXT TASK CODE, specify olq menu options.

To go directly to a specific screen, name the screen when you sign on. For example, to go straight to the Session Options screen when you sign on, type the following at the CA IDMS/DC or CA IDMS UCF system prompt:

V12 ENTER NEXT TASK CODE:
olq menu options

Signing off
To sign off from CA OLQ:

- Press [PF3] from any menu facility screen.
- Type QUIT or BYE in the command line.

In this section, you will learn how to do the following in CA OLQ:

- Getting Help (see page 452)
- Using Menu Facility (see page 453)
- Using PF Keys (see page 455)

## Getting Help

The CA OLQ help system provides quick answers about how to use CA OLQ.

For example, if you're working on the Column Select screen and have a question about how to specify selection criteria, you can press [PF1] to get more information.

The following help screen shows ways to specify retrieval selection criteria.

```plaintext
CA OLQ. Release nn.n  *** Help ***
->           Page 10 of 20
127000 Press the ENTER key to go to the next screen

HELP FOR *** COLUMN (FIELD) SELECT ***
------------------------------------------------------------------
Retrieving all rows If you don't enter any selection criteria on the screen, CA OLQ will retrieve all rows from the data tables or records.

What to type under SELECTION CRITERIA

Each column has its own SELECTION CRITERIA entry. If you want to retrieve rows based on the value in a certain column, fill in the SELECTION CRITERIA entry of that column. For example:

<table>
<thead>
<tr>
<th>SELECTION CRITERIA</th>
</tr>
</thead>
<tbody>
<tr>
<td>X 02 COLUMN1</td>
</tr>
<tr>
<td>X 02 COLUMN2</td>
</tr>
<tr>
<td>X 02 COLUMN3 eq 2000</td>
</tr>
<tr>
<td>X 02 COLUMN4 gt 5 * COLUMN1</td>
</tr>
</tbody>
</table>

------------------------------------------------------------------
```

<table>
<thead>
<tr>
<th>Type of Help</th>
<th>PF key</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>How to use CA OLQ in general</td>
<td>[PF 2]</td>
<td>General information on how to use the CA OLQ menu facility. Topics include CA OLQ command-line commands, PF-key assignments, and overviews of how to report on tables and records.</td>
</tr>
<tr>
<td>How to use a specific menu facility screen</td>
<td>[PF 1]</td>
<td>How to use each screen, including how to use each of the fields on the screen.</td>
</tr>
<tr>
<td>What the message on your screen means</td>
<td>[PF 4]</td>
<td>Explains the messages that you receive on the message line of the screen.</td>
</tr>
</tbody>
</table>
Using Menu Facility

Screen components

Each CA OLQ screen has:

- **The screen name** listed in the top right corner.

- **A page field** under the screen name. The page field shows you which page of the current screen or report you are on. You can type over this field to jump to a specific page.

- **A command line** marked by the -> prompt. You can enter commands to move you to other screens or to perform other CA OLQ functions.

- **A message line** right below the command line. CA OLQ messages either tell you what to do next or signal that you have tried to perform an invalid function.

- **A list of PF keys** along the bottom of the screen. The PF key assignments correspond to those established at system generation.

```
CA, Inc.
--->
Page 1 Of 3
```

---

### How to use the screens

**To select a screen option**, type any character (except a blank, underscore, or d) on the line next to the option listing and press [Enter].

For example, to select the EMPLOYEE record from the Record Select screen, type a character on the line next to the EMPLOYEE option and press [Enter].
Enter records:

-and/or-

Select records:

- COVERAGE
- DENTAL-CLAIM
- DEPARTMENT

x EMPLOYEE
- EMPOSITION
- EXPERTISE
- HOSPITAL-CLAIM
- INSURANCE-PLAN
- JOB
- NON-HOSP-CLAIM
- OFFICE
- SKILL
- STRUCTURE

1=HELP 3=QUIT 4=MESSAGE 6=MENU PA2=REFRESH

To get to another menu screen, type the command for that screen in the command line and press [Enter].

For example, to get to the Menu screen, type MENU in the command line of any menu facility screen and press [Enter].

To specify syntax on your screen, enter the syntax statement next to the prompt on the screen.

For example, to specify a COMPUTE statement, enter the statement next to the Compute prompt.
Using PF Keys

PF keys are used to:

- **Scroll through pages of a screen** ([PF7] and [PF8])
- **Display the current report** ([PF5])
- **Access a specific screen** (for example, [PF6] for the Menu screen)
- **Page right and left** through the default sequence of the Report Format screens ([PF10] and [PF11])
- **Access the help system** ([PF1], [PF2], [PF4])
- **Terminate a CA OLQ session** ([PF3])

For example, CA OLQ has a set of screens that you use to add formatting enhancements to your report. These screens are set up so that you can scroll through them by using [PF10] and [PF11].

After you have created your report, you can enhance its appearance by using the Report Format screens. Usually, you start with the Report Format - Sort screen and use [PF11] to scroll right to additional formatting screens.

**Default Paging Sequence of the Report Format screens**: You can use PF keys to page between the CA OLQ report formatting screens.

![Diagram of the default paging sequence of the Report Format screens]

IDMSDB--Using PF Keys

PF key values are assigned at system generation. The values given in this discussion are the default system-generation values. If your site has other PF key assignments, they will be reflected in the help system.
How to Make a Report from a Table

In this section In this section, you create a report that retrieves data from a table. The following report displays data stored in the EMPLOYEE table.

```plaintext
CA OLQ Release nn.n
*** Display Report ***
->
125004 press the ENTER key for DISPLAY/FORMAT ACTIVITY selections

EMPLOYEE REPORT

mm/dd/yy

EMP-LAST-NAME | DEPT-ID | SALARY-AMOUNT | PROJECT
---------------|--------|---------------|--------
BANK          | 4000   | 80000.00      | TESTING
ANGELO        | 4000   | 18000.00      | PLANNING
MCDUGALL      | 4000   | 18000.00      | PLANNING
PENMAN        | 4000   | 39000.00      | PLANNING
JACKSON       | 4000   | 34000.00      | PLANNING
ZEDI          | 4000   | 37000.00      | EVALUATE

END OF REPORT
```

What is a table?

A table is a systematic arrangement of data in **rows** and **columns**:

- Each row contains related information on a particular item. For example, data about an employee named Jackson.
- Each column represents a category of information. For example, each employee's last name.

In other words, columns are **attributes** associated with each row. Each row contains one entry for each column.

Figure 3-1 on page 3-4 shows a table containing information on employees.

The EMPLOYEE table:

```plaintext
EMPLOYEE Table

<table>
<thead>
<tr>
<th>EMP-ID</th>
<th>EMP-LAST-NAME</th>
<th>EMP-FIRST-NAME</th>
<th>START-YEAR</th>
<th>DEPT-HEAD-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>0075</td>
<td>Lanzarotta</td>
<td>Jennifer</td>
<td>78</td>
<td>0003</td>
</tr>
<tr>
<td>3302</td>
<td>Elopoulos</td>
<td>Bart</td>
<td>67</td>
<td>0004</td>
</tr>
<tr>
<td>3871</td>
<td>Mahoney</td>
<td>Reginald</td>
<td>90</td>
<td>0007</td>
</tr>
<tr>
<td>4230</td>
<td>Ho</td>
<td>Duc</td>
<td>91</td>
<td>0011</td>
</tr>
<tr>
<td>6264</td>
<td>Ortega</td>
<td>Daniel</td>
<td>91</td>
<td>0013</td>
</tr>
<tr>
<td>6348</td>
<td>Jones</td>
<td>Edith</td>
<td>85</td>
<td>0015</td>
</tr>
<tr>
<td>7170</td>
<td>Poznanski</td>
<td>Anthea</td>
<td>88</td>
<td>0030</td>
</tr>
<tr>
<td>8939</td>
<td>Sahu</td>
<td>Ankur</td>
<td>77</td>
<td>0321</td>
</tr>
<tr>
<td>8957</td>
<td>Sternbach</td>
<td>William</td>
<td>82</td>
<td>0349</td>
</tr>
</tbody>
</table>
```
The EMP-ID column represents one category of information on the EMPLOYEE table. Each row represents one data occurrence, which includes all the information on one employee.

For more information on tables, refer to CA IDMS SQL Self-Training Section.

Key Terms

Column
A vertical division in a table. A column represents a category of information, for example, employee last name.

Table
A presentation of data as a series of rows and columns.

- ASF tables refers to tables associated with the IDMSR schema.
- SQL tables refers to tables associated with an SQL schema.

Row
A horizontal division in a table. A row represents one data occurrence, for example, information on each employee.

Selection criteria
An expression that specifies which rows of a table are to be selected for processing.

Creating Your Report From an SQL Table

Contents

- Step 1 -- Select the type of table (see page 458)
- Step 2 -- Select a table (see page 458)
- Step 3 -- Choose columns and specify selection criteria (see page 459)
- Step 4 -- Specifying additional selection criteria (see page 461)
- Step 5 -- Retrieve the data (see page 462)
- Step 6 -- Display your report (see page 464)

In this example, you create a report from an SQL table using the following steps:

1. Be certain the access switch is set to idms
2. Select a table
3. Indicate which columns you want to appear in your report and specify selection criteria that determine which rows are retrieved for your report
4. Specify any additional selection criteria for rows to be retrieved for your report

5. Retrieve the data for your report

6. Display your report

**Step 1 -- Select the type of report**

When you sign on to CA OLQ, select the Session Options screen:

```
V12 ENTER NEXT TASK CODE: olq menu options
```

For this example, select Y to access SQL tables:

```
Access IDMS SQL tables: Y (Y/N)
```

User options: Help Option Change Current option Alternate option
- Report Processing Options -
  _ _ NOFiller Filler
  _ _ FULL SParse
  _ _ HEAder NOHeader

- Column Processing Options -
  _ _ OLOheader NOOlqheader
  _ _ PICture NOPIcture
  _ _ CODetable NOCODetable

1=HELP 3=QUIT 4=MESSAGE 6=MENU 8=FWD

For more information on the access switch, refer to the CA OLQ Reference Section.

**Step 2 -- Select a table**

For this example, select **Choose tables**.

```
Select Pfkey Description Command/ Screen Name Show/ Help
  --- Choose tables TABLE

  --- Retrieval Activity ---
  _ Choose records from selected subschema RECORD
  _ Choose columns for report COLUMN
  _ Retrieve data to build report RETrieve
```
Specify `SELECT` and select the `EMPLOYEE` table.

**Step 3 -- Choose columns and specify selection criteria**

When you create a report that retrieves data from a table, you do not have to display the whole table. You can restrict the scope of your report in three ways:

- By choosing which columns you want to display
- By specifying selection criteria to restrict which rows are retrieved for your report
- By specifying that you want to display only unique rows on your report

**Choosing columns**

You don't have to display all of the columns in your table. By using the Column Select screen, you can select only those columns that you want to include in your report.

For example, if you just wanted to list employees' names and department numbers, you would only select those columns:
Specifying selection criteria

Selection criteria are logical expressions that you use to tell CA OLQ which rows of data to retrieve for your report. You specify your selection criteria in the Selection criteria field of the Column Select screen.

For example, if you only wanted to list the names of those employees in department 4000, you would specify the following:

```
Columns Currently Selected: 0 Selection Criteria Distinct Y Y/N
EMPDEMO.EMPLOYEE
  _ 02 EMP_ID 0415
  _ 02 EMP_LAST_NAME 0415
  _ 02 EMP_FIRST_NAME 0415
  _ 02 EMP_STREET 0415
  _ 02 EMP_CITY 0415
  _ 02 EMP_STATE 0415
  _ 02 EMP_ZIP_FIRST_FIVE 0415
  _ 02 EMP_ZIP_LAST_FOUR 0415
  _ 02 EMP_PHONE 0415
  _ 02 EMP_SALARY_AMOUNT 0415
  _ 02 EMP_START_YEAR 0415
  _ 02 EMP_TERMAINATION_DATE 0415
  _ 02 EMP_DEPT_ID eq 4000
```

Specifying only unique rows

You do not have to include duplicate rows (detail lines) on your report. By using the Distinct Y/N field on the Column Select screen, you can display a report that contains only unique rows.

Select lastname, state, phone, salary, start year, termination date, and department ID. Specify that you want to display only unique rows. Also, specify that you want to view only the list of employees in department 4000.

```
CA OLQ Release nn.n *** Column Select ***
->
124000 Select columns, specify selection criteria and press the ENTER key.

Columns Currently Selected: 0 Selection Criteria Distinct N Y/N
EMPDEMO.EMPLOYEE
  _ 02 EMP_ID 0415
  _ 02 EMP_LAST_NAME 0415
  _ 02 EMP_FIRST_NAME 0415
  _ 02 EMP_STREET 0415
  _ 02 EMP_CITY 0415
  _ 02 EMP_STATE 0415
  _ 02 EMP_ZIP_FIRST_FIVE 0415
  _ 02 EMP_ZIP_LAST_FOUR 0415
  _ 02 EMP_PHONE 0415
  _ 02 EMP_SALARY_AMOUNT 0415
  _ 02 EMP_START_YEAR 0415
  _ 02 EMP_TERMAINATION_DATE 0415
  _ 02 EMP_DEPT_ID eq 4000
```
Step 4 -- Specifying additional selection criteria

Additional selection criteria

The Additional selection criteria field and the Selection Criteria screen, give you more room to add selection criteria for the rows of data you want in your report.

In this example, you want to report on those employees whose:

- Department ID is 4000 (already specified above)
- Phone exchange (first 3 digits) is '329'
- Last name begins with a 'C' or an 'S'
- State of residence is Massachusetts or New Hampshire
- Annual salary is greater than $40,000 and less than $100,000
- Start date with the company was after 1988
- Status is active (still an active employee)

Begin entering the additional selection criteria on this screen. Then specify Y to go to the Selection Criteria Screen for more room to enter selection criteria.

Finish entering the additional selection criteria.
With the Selection Criteria screen you can also enter:

- Exists predicates
- Quantified predicates (ANY, SOME, ALL)
- Nested select predicates
- Multiple criteria for the same command
- Group by, order by, and so on

**Step 5 -- Retrieve the data**

When you are retrieving the data for your report, two CA OLQ screens help you monitor what’s going on.

**The Retrieval Interrupted screen**

This screen indicates that your report will contain more records than the current interrupt count allows.

⚠️ **Note:** This screen is here just to show you an example. The numbers on the screen do not reflect the numbers you will encounter in this example.

Specify Yes to continue data retrieval.
Retrieval interrupted due to excessive data base accesses.

Number of whole rows........... 90
Total number of records read.... 100
Total number of records selected.. 90
Number of data errors .......... 0

Continue execution  x Yes
X No

Current interrupt interval is 100 data base accesses.

What is an interrupt count?

The interrupt count indicates how many records are retrieved from the database at a time. It serves as a testing feature when you are designing your report, enabling you to create the report format that you want without retrieving more data than you need to.

Changing the interrupt count

The interrupt count is set at system generation. You can change this value by using the **Interrupt Count** option on the Retrieval Interrupted screen, or on the Session Options screen.

What to do

If the number of records that meet the criteria for your report exceeds the interrupt count, the Retrieval Interrupted screen asks you whether you want to continue data retrieval:

- **If you want to continue retrieval** select the **Yes** function on the screen. CA OLQ continues to retrieve more records until the interrupt count is met again, or until all of the records that meet the criteria for your report have been retrieved.

- **If you want to terminate retrieval** select the **No** function on the screen. CA OLQ create a report that contains only those records that have been retrieved up to this point.

The Retrieval Completed screen

This screen indicates that retrieval is completed for your report. From here you can either display your report or select one of the report formatting or print options from the menu on the screen.

In this example, only eight records are retrieved for your report. Since this is less than the current interrupt count, you bypass the Retrieval Interrupted screen and go straight to the Retrieval Completed screen.

```
CA OLQ Release nn.n              *** Retrieval Completed ***
- > 130000 Select activity and press the ENTER key.

Number of whole rows............ 8  Total displayable cols 20
```
Step 6 -- Display your report

Your report displays information on the employees in department 4000 with the additional selection criteria you specified.

Press [PF11] to see the right half of the report.
Creating Your Report From an ASF Table

Contents

- Step 1 -- Select the type of table (see page 465)
- Step 2 -- Select a table (see page 466)
- Step 3 -- Choose columns and specify selection criteria (see page 467)
- Step 4 -- Specifying additional selection criteria (see page 468)
- Step 5 -- Retrieve the data (see page 470)
- Step 6 -- Display your report (see page 471)

In this example, you create a report from a table using the following steps:

1. Be certain that the access switch is set to **olq**.

2. Select a table.

3. Indicate which columns you want to appear in your report and specify selection criteria that determine which rows are retrieved for your report.

4. Retrieve the data for your report.

5. Display your report.

**Step 1 -- Select the type of table**

When you sign on to CA OLQ, select the Session Options screen:

```
V12 ENTER NEXT TASK CODE:
olq menu options
```

For this example, select **N** to access ASF tables:

```
CA OLQ Release nn.n
-> *** Session Options ***
107017 CA OLQ Release nn.n
107019 Copyright(C) 2003 CA,Inc.
Current interrupt count: 100
Access IDMS SQL tables: N (Y/N)
User options: 
Help Change Current option Alternate option
-> Report Processing Options <-
  NOFiller FIller
Page Columns Spread: (L-Left,E-Even,M-Max,nn)
```
Step 2 -- Select a table

For this example, select Choose tables.

Specify SELECT and select the EMPLOYEE table.
Step 3 -- Choose columns and specify selection criteria

When you create a report that retrieves data from a table, you do not have to display the whole table. You can restrict the scope of your report in three ways:

- By choosing which columns you want to display
- By specifying selection criteria to restrict which rows are retrieved for your report
- By specifying that you want to display only unique rows on your report

Choosing columns

You don't have to display all of the columns in your table. By using the Column Select screen, you can select only those columns that you want to include in your report.

For example, if you just wanted to list employees' names and department numbers, you would only select those columns:

<table>
<thead>
<tr>
<th>Columns Currently Selected: 0</th>
<th>Selection Criteria</th>
<th>Distinct N Y/N</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x 02 EMP-ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x 02 EMP-LAST-NAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02 EMP-FIRST-NAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02 START-YEAR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x 02 DEPT-HEAD-ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x 02 DEPT-ID</td>
<td>eq 4000</td>
<td></td>
</tr>
<tr>
<td>02 SALARY-AMOUNT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02 PROJECT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02 OFFICE-CODE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifying selection criteria

Selection criteria are logical expressions that you use to tell CA OLQ which rows of data to retrieve for your report. You specify your selection criteria in the Selection criteria field of the Column Select screen.

For example, if you only wanted to list the names of those employees in department 4000, you would specify the following:

<table>
<thead>
<tr>
<th>Columns Currently Selected: 0</th>
<th>Selection Criteria</th>
<th>Distinct N Y/N</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x 02 EMP-ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x 02 EMP-LAST-NAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02 EMP-FIRST-NAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02 START-YEAR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02 DEPT-HEAD-ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x 02 DEPT-ID</td>
<td>eq 4000</td>
<td></td>
</tr>
<tr>
<td>02 SALARY-AMOUNT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02 PROJECT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02 OFFICE-CODE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
More information on how to specify selection criteria is given in Section 11, "How to Make a Report from Database Records".

**Specifying only unique rows**

You do not have to include duplicate rows (detail lines) on your report. By using the **Distinct Y/N** field on the Column Select screen, you can display a report that contains only unique rows.

Select last name, start year, department head, department ID, salary amount, and project. Specify that you want to display only unique rows. Also, specify that you want to view only the list of employees in department 4000.

```
CA OLQ Release nn.n*** Column Select ***
-> *** Page 1 of 1

124000 Select columns, specify selection criteria and press the ENTER key.

Columns Currently Selected: 0 Selection Criteria Distinct N Y/N

  EMPLOYEE
  EMP-ID
  EMP-LAST-NAME
  EMP-FIRST-NAME
  START-YEAR
  DEPT-HEAD-ID
  DEPT-ID eq 4000
  SALARY-AMOUNT
  PROJECT
  OFFICE-CODE

Additional Selection Criteria:

Proceed to Selection Criteria Screen? N Y/N

1=HELP 3=QUIT 4=MESSAGE 6=MENU PA2=REFRESH
```

**Step 4 -- Specifying additional selection criteria**

**Additional selection criteria**

The **Additional selection criteria** field and the **Selection Criteria** screen, give you more room to add selection criteria for the rows of data you want in your report.

In this example, you want to report on those employees whose:

- Department ID is 4000 (already specified above)
- Project is Evaluation
- Manager is Ms. Sanchez (DEPT-HEAD-ID = 0111)
- Last name begins with a 'C' or an 'S'
- Annual salary is greater than $40,000 and less than $100,000
- Start date with the company was after 1988
Begin entering the additional selection criteria on this screen. Then specify Y to go to the Selection Criteria Screen for more room to enter selection criteria.

```
CA OLQ Release nn.n *** Column Select ***
->

124000 Select columns, specify selection criteria and press the ENTER key.

Columns Currently Selected: 0 Selection Criteria Distinct N Y/N
  EMPLOYEE
  02 EMP-ID
  s 02 EMP-LAST-NAME
  02 EMP-FIRST-NAME
  02 START-YEAR
  02 DEPT-HEAD-ID
  02 DEPT-ID eq 4000
  02 SALARY-AMOUNT
  02 PROJECT
  02 OFFICE-CODE

Additional Selection Criteria: PROJECT EQ 'EVALUATION'

Proceed to Selection Criteria Screen? Y Y/N
1=HELP 3=QUIT 4=MESSAGE 6=MENU PA2=REFRESH
```

Finish entering the additional selection criteria.

```
CA OLQ Release nn.n ***Selection Criteria***
->

146000 Type in selection criteria, and press the ENTER key.
Please Enter Additional Selection Criteria:
PROJECT EQ 'EVALUATION' AND (DEPT-ID EQ 0111) AND (EMP-LAST-NAME EQ 'C' OR EMP-LAST-NAME EQ 'S') AND (SALARY-AMOUNT GT 40000 AND SALARY-AMOUNT LT 100000) AND (START-YEAR GT 88)

1=HELP 3=QUIT 4=MESSAGE 6=MENU PA2=REFRESH
```

With the Selection Criteria screen you can also enter:

- Logical record keywords
- Criteria expressions for subscripted fields
Step 5 -- Retrieve the data

When you are retrieving the data for your report, two CA OLQ screens help you monitor what's going on.

The Retrieval Interrupted screen

This screen indicates that your report will contain more records than the current interrupt count allows.

⚠️ Note: This screen is here just to show you an example. The numbers on the screen do not reflect the numbers you will encounter in this example.

Specify Yes to continue data retrieval.

<table>
<thead>
<tr>
<th>CA OLQ Release nn.n</th>
<th>*** Retrieval Interrupted ***</th>
</tr>
</thead>
<tbody>
<tr>
<td>-&gt; 131000</td>
<td>Select YES or NO and press the ENTER key</td>
</tr>
<tr>
<td>Retrieval interrupted due to excessive data base accesses.</td>
<td></td>
</tr>
<tr>
<td>Number of whole rows. . . . . . . . . 46</td>
<td></td>
</tr>
<tr>
<td>Total number of records read. . . . . 100</td>
<td></td>
</tr>
<tr>
<td>Total number of records selected. . . . . 46</td>
<td></td>
</tr>
<tr>
<td>Number of data errors . . . . . . . . . . . . 0</td>
<td></td>
</tr>
<tr>
<td>Continue execution x Yes</td>
<td></td>
</tr>
<tr>
<td>Current interrupt interval is 100 data base accesses.</td>
<td></td>
</tr>
<tr>
<td>1=HELP 3=QUIT 4=MESSAGE</td>
<td></td>
</tr>
</tbody>
</table>

What is an interrupt count?

The interrupt count indicates how many records are retrieved from the database at a time. It serves as a testing feature when you are designing your report, enabling you to create the report format that you want without retrieving more data than you need to.

Changing the interrupt count

The interrupt count is set at system generation. You can change this value by using the Interrupt Count option on the Retrieval Interrupted screen, or on the Session Options screen.

What to do
If the number of records that meet the criteria for your report exceeds the interrupt count, the Retrieval Interrupted screen asks you whether you want to continue data retrieval:

- **If you want to continue retrieval** select the Yes function on the screen. CA OLQ will continue to retrieve more records until the interrupt count is met again, or until all of the records that meet the criteria for your report have been retrieved.

- **If you want to terminate retrieval** select the No function on the screen. CA OLQ will create a report that contains only those records that have been retrieved up to this point.

**The Retrieval Completed screen**

This screen indicates that retrieval is completed for your report. From here you can either display your report or select one of the report formatting or print options from the menu on the screen.

In this example, only six records are retrieved for your report. Since this is less than the current interrupt count, you bypass the Retrieval Interrupted screen and go straight to the Retrieval Completed screen.

---

**Step 6 -- Display your report**

Your report displays information on the employees in department 4000 with the additional selection criteria you specified.

---

**CA OLQ Release nn.n *** Display Report ***

Line 1

125004 press the ENTER key for DISPLAY/FORMAT ACTIVITY selections

<table>
<thead>
<tr>
<th>EMP-LAST-NAME</th>
<th>START-YEAR</th>
<th>DEPT-HEAD-ID</th>
<th>DEPT-ID</th>
<th>SALARY-AMOUNT</th>
<th>PROJECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALLAHAN</td>
<td>89</td>
<td>0111</td>
<td>4000</td>
<td>54000.00</td>
<td>EVALUATE</td>
</tr>
<tr>
<td>CARTWRIGHT</td>
<td>91</td>
<td>0111</td>
<td>4000</td>
<td>80000.00</td>
<td>EVALUATE</td>
</tr>
<tr>
<td>CEDI</td>
<td>90</td>
<td>0111</td>
<td>4000</td>
<td>78000.00</td>
<td>EVALUATE</td>
</tr>
<tr>
<td>SAVINO</td>
<td>91</td>
<td>0111</td>
<td>4000</td>
<td>78000.00</td>
<td>EVALUATE</td>
</tr>
</tbody>
</table>
How to Make a Report From More Than One Table

In this section, you create a report that combines information from two tables, the DEPARTMENT table and the EMPLOYEE table.

Join
A relational operation through which two or more tables are combined. Tables are joined based on columns that the tables have in common.

Join criteria
A logical expression that compares like columns in two or more tables.

Project
A relational operation through which only particular columns of a table are accessed.

Select
A relational operation through which only particular rows of a table are accessed.

Selection criteria
An expression that specifies which rows of a table are to be selected for processing.
How to Combine Data From More Than One Table

Relational operations

When you combine data from two or more tables, you need to specify information that relates them somehow. Three relational operations, select, project, and join, can be used to define and access tables:

- **Select** enables you to choose which **rows** you want to include in your report.
- **Project** enables you to choose which **columns** you want to display.
- **Join** enables you to combine **two or more tables** on the basis of common values.

In the preceding section, you applied two of these relational operations:

- You **selected** which rows you want displayed in your report by specifying selection criteria.
  
  ```sql
  select * from department where dept-id = 4000
  ```

<table>
<thead>
<tr>
<th>DEPT-NAME</th>
<th>DEPT-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Operations</td>
<td>3200</td>
</tr>
<tr>
<td>Public Relations</td>
<td>4000</td>
</tr>
<tr>
<td>Thermoregulation</td>
<td>5200</td>
</tr>
</tbody>
</table>

  **Selecting rows:**

- You **projected** which columns you want displayed in your report.

  ```sql
  select emp-last-name, dept-id from employee
  ```

<table>
<thead>
<tr>
<th>EMP-ID</th>
<th>EMP-LAST-NAME</th>
<th>EMP-FIRST-NAME</th>
<th>START-YEAR</th>
<th>DEPT-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>0235</td>
<td>Alanza</td>
<td>Mariana</td>
<td>87</td>
<td>3200</td>
</tr>
<tr>
<td>1450</td>
<td>Enrique</td>
<td>Uriel</td>
<td>76</td>
<td>4000</td>
</tr>
<tr>
<td>9011</td>
<td>Roberts</td>
<td>Ellen</td>
<td>90</td>
<td>4000</td>
</tr>
<tr>
<td>6732</td>
<td>Ditka</td>
<td>Robert</td>
<td>81</td>
<td>4000</td>
</tr>
<tr>
<td>7648</td>
<td>Carr</td>
<td>Eustace</td>
<td>84</td>
<td>5200</td>
</tr>
<tr>
<td>5552</td>
<td>Chiu</td>
<td>Li</td>
<td>86</td>
<td>5200</td>
</tr>
</tbody>
</table>

  **EMP/DEPT-ID**

<table>
<thead>
<tr>
<th>EMP-LAST-NAME</th>
<th>DEPT-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Projecting Columns:

In this section you specify join criteria to combine data from the EMPLOYEE and DEPARTMENT tables into a report.

What are join criteria?

Join criteria are logical expressions that equate a column in one table with equivalent columns in additional tables. You must specify join criteria if you are reporting on more than one table at a time.

Joining tells CA OLQ which columns the tables have in common. By comparing the values in these columns, CA OLQ can match the rows in the tables and retrieve only those rows that the tables share.

Join

```
select depart.dept-name, department.dept-id, employee.emp-last-name
from employee, department where department.dept-id = employee.dept-id
```

<table>
<thead>
<tr>
<th>DEPT-ID</th>
<th>DEPT-NAME</th>
<th>EMPLOYEES BY DEPARTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>3200</td>
<td>Computer Operations</td>
<td></td>
</tr>
<tr>
<td>4000</td>
<td>Public Relations</td>
<td></td>
</tr>
<tr>
<td>5200</td>
<td>Thermoregulation</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DEPT-ID</th>
<th>DEPT-NAME</th>
<th>EMPLOYEE</th>
</tr>
</thead>
<tbody>
<tr>
<td>4000</td>
<td>Public Relations</td>
<td>Enrique</td>
</tr>
<tr>
<td>4000</td>
<td>Public Relations</td>
<td>Ditka</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DEPT-ID</th>
<th>DEPT-NAME</th>
<th>EMP-LAST-NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>3200</td>
<td>Computer Operations</td>
<td>Alanza</td>
</tr>
<tr>
<td>4000</td>
<td>Public Relations</td>
<td>Enrique</td>
</tr>
<tr>
<td>4000</td>
<td>Public Relations</td>
<td>Ditka</td>
</tr>
<tr>
<td>5200</td>
<td>Thermoregulation</td>
<td>Carr</td>
</tr>
</tbody>
</table>

Joining the EMPLOYEE and DEPARTMENT tables:

How do you specify join criteria?

To join two tables, find a column that the two tables have in common. The columns do not have to have the same name, but they should:

- Have the same internal picture
- Have the same data type (for example, numeric)
- Contain some data that matches (in order to display data from both tables)

**Where do you specify join criteria?**

You specify join criteria in one of two places -- on the Column Select screen, or on the Selection Criteria screen:

- **On the Column Select screen, in the Selection criteria field next to the join column:** For example, these join criteria join TABLE1 and TABLE2 based on common values of the ID and NUMBER fields:

  ```
  Columns Currently Selected: 0  Selection Criteria  Distinct N Y/N
  TABLE1
  X 03 ID
  X 03 NAME
  X 03 PHONE
  TABLE2
  - 03 NUMBER  eq ID
  X 03 SALARY
  Additional Selection Criteria:
  
  Proceed to Selection Criteria Screen? N Y/N
  ```

  (Note that you could also specify the join criteria, *eq ID*, next to the TABLE1 ID column.)

- **On the Column Select screen, under Additional Selection Criteria:**

  ```
  Columns Currently Selected: 0  Selection Criteria  Distinct N Y/N
  TABLE1
  X 03 ID
  X 03 NAME
  X 03 PHONE
  TABLE2
  - 03 NUMBER  eq id
  X 03 SALARY
  Additional Selection Criteria:
  name eq 'george'
  
  Proceed to Selection Criteria Screen? N Y/N
  ```

- **On the Selection Criteria screen, under Additional Selection Criteria:**

  ```
  146000 Type in selection criteria, and press the ENTER key.  
  Please Enter Additional Selection Criteria:  
  id eq number and (phone matches '617******') and (salary gt 30000 and salary lt 50000) 
  ```

**Example**

You can join the EMPLOYEE and JOB tables because they both have columns that represent employee ID numbers (column ID in EMPLOYEE and column EMP-NUMBER in JOB). To join them, specify the following:

```
Columns Currently Selected: 0  Selection Criteria  Distinct N Y/N
EMPLOYEE
 X 03 ID
 X 03 NAME
 ```
Note that you do not have to select the EMP-NUMBER column from JOB. If you did, the report would list the employee ID twice, once for ID and once for EMP-NUMBER.

What if columns have the same name?

If your common columns have the same name, you must include the table name in the join criteria. The table names and the column names are separated by a period (.).

For example, if both EMPLOYEE and JOB contain the ID field, specify:

<table>
<thead>
<tr>
<th>Columns Currently Selected: 0</th>
<th>Selection Criteria</th>
<th>Distinct N Y/N</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X 03 ID</td>
<td>eq employee.id</td>
<td></td>
</tr>
<tr>
<td>X 03 NAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X 03 PHONE</td>
<td>eq employee.id</td>
<td></td>
</tr>
<tr>
<td>JOB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X 03 ID</td>
<td>eq employee.id</td>
<td></td>
</tr>
</tbody>
</table>

What happens if you don't specify join criteria?

When the access mode is set to olq, CA OLQ automatically prevents you from leaving out your join criteria. If you select two tables and don't specify how to join them, CA OLQ issues a message reminding you to do so.

Creating your report

Contents

- Step 1 -- Select the type of table (see page 477)
- Step 2 -- Select tables (see page 477)
- Step 3 -- Select, project, and join (see page 478)
- Step 4 -- Retrieve the data (see page 479)
- Step 5 -- Display your report (see page 479)

In this example, you create a report from two tables using the following steps:

1. Decide what kind of table you want to report on, a ASF table or an SQL table.
2. Choose two tables.
3. Specify selection criteria that select which rows of data are retrieved for your report.
4. Project which columns you want to appear in your report.
5. Specify join criteria that relate the two tables based on a common column.

6. Retrieve the data for your report.

7. Display your report.

**Step 1 -- Select the type of table**

When you sign on to CA OLQ, select the Session Options screen:

```
V12 ENTER NEXT TASK CODE:
olq menu options
```

For this example, select Y to access SQL tables:

```
CA OLQ Release nn.n

Access IDMS SQL tables: N (Y/N)
```

**Step 2 -- Select tables**

When you sign on to CA OLQ, the first screen you see is the Menu screen. Select tables.

For this example, select **Choose tables**.
Specify **SELECT** and select the DEPARTMENT and EMPLOYEE tables.

**Step 3 -- Select, project, and join**

In this step, you perform all three relational operations: select, project, and join:

- **Select** by restricting the rows of the report to those employees hired before 1990.
- **Project** by displaying only the DEPT-NAME, EMP-ID, EMP-LAST-NAME, and START-YEAR columns.
- **Join** by linking the DEPARTMENT and EMPLOYEE tables based on common values of the DEPT-ID field.

Select which columns you want to list in your report, specify that you want to include only those employees hired before 1990, and join the two tables based on common values in the DEPT-ID fields.
Step 4 -- Retrieve the data

Depending on how high the interrupt count at your site is set, CA OLQ may bypass the Retrieval Interrupted screen and proceed directly to the Retrieval Completed screen.

Step 5 -- Display your report

Your report displays information from both the EMPLOYEE and DEPARTMENT tables.

EMPLOYEE/DEPARTMENT REPORT

<table>
<thead>
<tr>
<th>EMP-ID</th>
<th>EMP-LAST-NAME</th>
<th>START-YEAR</th>
<th>DEPT-NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>0007</td>
<td>BANK</td>
<td>78</td>
<td>PUBLIC RELATIONS</td>
</tr>
<tr>
<td>0120</td>
<td>ANGELO</td>
<td>79</td>
<td>PUBLIC RELATIONS</td>
</tr>
<tr>
<td>0149</td>
<td>PENMAN</td>
<td>77</td>
<td>PUBLIC RELATIONS</td>
</tr>
<tr>
<td>0158</td>
<td>JACKSON</td>
<td>77</td>
<td>PUBLIC RELATIONS</td>
</tr>
<tr>
<td>0329</td>
<td>FINN</td>
<td>79</td>
<td>THERMOREGULATION</td>
</tr>
<tr>
<td>0349</td>
<td>WILCO</td>
<td>79</td>
<td>THERMOREGULATION</td>
</tr>
</tbody>
</table>
How to Format Your Report

Using CA OLQ, you can format your report by:

- Sorting your report rows
- Changing the external picture of your data columns
- Modifying how data columns are displayed
- Changing your column headers
- Making page headers and footers

You can perform the following steps:
- Format Report Key Terms (see page 481)
- Creating a Report (see page 482)
- Sorting Your Report Rows (see page 491)
- Editing Your Report Values (see page 493)
- Changing Your Column Relative Positions (see page 498)
- Changing Your Column Pictures (see page 500)
- Changing Column Headers (see page 503)
- Making Page Headers and Footers (see page 506)
Format Report Key Terms

Code table

A table that is defined in the data dictionary and that contains corresponding pairs of values. For example:

<table>
<thead>
<tr>
<th>Internal Code</th>
<th>Displayed Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Alabama</td>
</tr>
<tr>
<td>02</td>
<td>Alaska</td>
</tr>
<tr>
<td>03</td>
<td>Arizona</td>
</tr>
</tbody>
</table>

Internal codes are used to efficiently store the data in the database. External values are used to display data in programs or reports.

Column alignment

A way to specify how report data columns align under the column headers. Options are left-justified, right-justified, and centered.

Column header

A header at the top of each column of report data.

Display sequence

Determines the order in which report columns are displayed.

External picture

A code that defines the way your column value is formatted in your report. The external picture is used to add punctuation (for example, commas) and special characters (for example, dollar signs) to your column data display.

Page header

A title at the top of each page of your report.

Page footer

A title at the bottom of each page of your report.

Sort

A way to order report rows. CA OLQ sorts the rows in your report based on the value of the sort fields that you specify. Rows can be sorted in ascending or descending order.

Sparse/Full option
A CA OLQ editing feature that determines how column values that repeat in consecutive rows are displayed:

- **Sparse** displays only the first of a repeating set of column values.
- **Full** displays all occurrences of the repeating value.

## Creating a Report

### Contents

- Step 1 -- Select a subschema (see page 482)
- Step 2 -- Select records (see page 483)
- Step 3 -- Choose columns (see page 483)
- Step 4 -- Retrieve the data (see page 484)
- Step 5 -- Display your report (see page 485)
- Step 1 - Create a Current Report (see page 485)
- Step 2 - Modify Your Report Headers (see page 488)
- Step 3 - Change Your Report Column Pictures (see page 489)
- Step 4 - Sort Your Report Rows and Specify Group Calculations (see page 490)

Before you can apply any report formatting features, you must have a current report. In this example you build a current report with records from a subschema using the following steps:

⚠️ **Note:** Before you begin, make certain the access switch is set to `olq`.

For more information on setting the access switch, see 3.3.1, "Step 1 -- Select the type of table".

1. Select a subschema.
2. Select the records you want to include in your report.
3. Select which columns you want to display.
4. Retrieve the data from the database and display your report.

### Step 1 -- Select a subschema

Select the EMPSS01 subschema.

```
CA OLQ Release nn.n
-> 121000 Select a subschema and press the ENTER key

Dictionary name . : ASFDICT
Database name . : 

*** Signon Database View ***
Page 1 OF 1

*** Signon Database View ***
Page 1 OF 1
```

<table>
<thead>
<tr>
<th>CA OLQ Release nn.n</th>
<th>*** Signon Database View ***</th>
</tr>
</thead>
<tbody>
<tr>
<td>121000</td>
<td>Select a subschema and press the ENTER key</td>
</tr>
<tr>
<td>Dictionary name . : ASFDICT</td>
<td></td>
</tr>
<tr>
<td>Database name . :</td>
<td></td>
</tr>
</tbody>
</table>

16-Jan-2018 482/898
Step 2 -- Select records

Select the DEPARTMENT, EMPLOYEE, and EMPOSITION records.

Step 3 -- Choose columns

Step 4 -- Retrieve the data

Specify Yes to continue data retrieval.

---

CA OLQ Release nn.n  
-  
131000 Select YES or NO and press the ENTER key  
  
Retrieval interrupted due to excessive data base accesses.

Number of whole rows .        46  
Total number of records read . 100  
Total number of records selected. 90  
Number of data errors .        0  

Continue execution  x Yes  
_  No  

Current interrupt interval is  100 data base accesses.

1=HELP  3=QUIT  4=MESSAGE  
---

Depending on how high the interrupt count at your site is set, CA OLQ may bypass the Retrieval Interrupted screen and proceed directly to the Retrieval Completed screen.

---

CA OLQ Release nn.n  
-  
130000 Select activity and press the ENTER key.  
  
Number of whole rows .        68  
Total displayable cols .       20  
Total number of records read . 143  
Formatted line length .        372  
Total number of records selected. 133  
Number of data errors .        0  

Select Option Command/Screen Name  
X  Display report  DISPLAY  
_-  Save report  SAVE  
_  Choose the sort sequence of report  SORT  
_  Change column headers  HEADER  
_  Change page header and footer  PAGE HEADER  
_  Change display format of data ($,commas)  PICTURE  
_  Format columns (Alignment, sparse)  EDIT  
_  Specify summary computations (Totals)  GROUP BY  

Step 5 -- Display your report

Your report displays the data as it is retrieved from the database.

```
CA OLQ Release nn.n
*** Display Report ***
Page 1 Line 1
125000 Press the ENTER key to go to the next page of the report.

DEPARTMENT/EMPLOYEE/EMPOSITION REPORT

DEPT-ID-0410  EMP-LAST-NAME-0415  SALARY-AMOUNT-0420

   6666  HENDON         240000.00
   6666  PAPAZEUS       100000.00
   6666  PAPAZEUS       90000.00
   6666  RUPEE          80000.00
   6666  RUPEE          76000.00
   6666  WILDER         90000.00
   2000  BLOOMER        150000.00
   2000  HUTTON         440000.00
   2000  JENSON         820000.00
   2000  KIMBALL        450000.00
   2000  KING           145000.00
   2000  NICEMAN        140000.00

1=HELP      3=QUIT      4=MESSAGE      6=MENU      8=FWD      10=LEFT      11=RIGHT
```

Step 1 - Create a Current Report

When you save a report CA OLQ takes the set of commands you used to build your current report and saves them in the data dictionary. In this step, you build a current report.

This report uses the EMPLOYEE and EMPOSITION records from the sample employee database. These records reside in the EMPSS01 subschema.

Start on the Signon Database View screen. To get there, type `subschema` (or sub) in the command line of any screen.

Select the EMPSS01 subschema.

```
CA OLQ Release nn.n
*** Signon Database View ***
Page 3 OF 4
121000 Select a subschema and press the ENTER key

Dictionary name . : TSTDICT          Dictionary node name . : 
Database name . :                Database node name . : 

Specify Subschema :    of Schema . . :      Version :
     -or-
Select subschema:    Description:
 s EMPSS01 OF EMPSCHM VER 100  DEPARTMENT AND EMPLOYEE INFORMATION
 s FINANC01 OF EMPSCHM VER 100  4087
 s SALES01 OF EMPSCHM VER 100  SALES QUOTAS
```
Select the EMPLOYEE and EMPOSITION records.


Select Yes to continue data retrieval.
131000 Select YES or NO and press the ENTER key

Retrieval interrupted due to excessive data base accesses.

Number of whole rows . . . . . . . . 54
Total number of records read . . . 100
Total number of records selected . . 99
Number of data errors . . . . . . . . 0

Continue execution  X Yes
X No

Current interrupt interval is 100 data base accesses.

1=HELP  3=QUIT  4=MESSAGE

130000 Select activity and press the ENTER key

This is how your report looks before you add any formatting features.

 EMPLOYEE/EMPOSITION REPORT
 mm/dd/yy
 EMP-LAST-NAME-0415  SS-NUMBER-0415  SALARY-GRADE-0420  SALARY-AMOUNT-0420

LINGER  092345812  33  38500.00
TERNER  045672222  11  13000.00
BROWN  019556712  44  42500.00
CHARLES  019556712  43  38000.00
PENMAN  014593186  33  39000.00
DUNCAN  010673343  72  85000.00

1=HELP  3=QUIT  4=MESSAGE  6=MENU
Step 2 - Modify Your Report Headers

In this step, you modify your report headers. Start on the Report Format - Header screen. To get there, type `header` in the command line.

Change your report headers to make them more legible.

```
CA OLQ Release nn.n                      *** Report Format - Header ***
-->                                                      Page 1 of 1
134000 Specify column headers and press the ENTER key

Underline character:  -   Disp                  Header
  Seq
EMPLOYEE
  X  EMP-LAST-NAME-0415     1 name
  X  SS-NUMBER-0415        2 social security number
EMPOSITION
  X  SALARY-GRADE-0420    3 salary grade
  X  SALARY-AMOUNT-0420   4 salary

Compute:
1=HELP    3=QUIT    4=MESSAGE    5=DISPLAY    6=MENU    10=SORT    11=EDIT

CA OLQ displays your new report headers.
```

```
CA OLQ Release nn.n                      *** Display Report ***
-->                                                      Page 1 Line 1
125000 Press the ENTER key to go to the next page of the report.

EMPLOYEE/EMPOSITION REPORT
mm/dd/yy

<table>
<thead>
<tr>
<th>NAME</th>
<th>SOCIAL SECURITY NUMBER</th>
<th>SALARY GRADE</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINGER</td>
<td>092345812</td>
<td>33</td>
<td>38500.00</td>
</tr>
<tr>
<td>TERNER</td>
<td>045672222</td>
<td>11</td>
<td>13000.00</td>
</tr>
<tr>
<td>BROWN</td>
<td>019556712</td>
<td>44</td>
<td>42500.00</td>
</tr>
<tr>
<td>CHARLES</td>
<td>019556712</td>
<td>43</td>
<td>38000.00</td>
</tr>
<tr>
<td>PENMAN</td>
<td>014593186</td>
<td>33</td>
<td>39000.00</td>
</tr>
<tr>
<td>DUNCAN</td>
<td>018673343</td>
<td>72</td>
<td>85000.00</td>
</tr>
<tr>
<td>EVERETT</td>
<td>018673343</td>
<td>71</td>
<td>75000.00</td>
</tr>
<tr>
<td>LITERATA</td>
<td>023567831</td>
<td>43</td>
<td>37500.00</td>
</tr>
<tr>
<td>WILCO</td>
<td>029661234</td>
<td>72</td>
<td>80000.00</td>
</tr>
<tr>
<td>HEAROWITZ</td>
<td>031896154</td>
<td>42</td>
<td>33000.00</td>
</tr>
</tbody>
</table>
```
Step 3 - Change Your Report Column Pictures

In this step, you modify the display of the social security and salary columns to make them more legible. Start on the Report Format - Picture screen. To get there, type picture in the command line.

Change the picture for SS-NUMBER-0415 to include hyphens.

CA OLQ displays your new picture.

```
CA OLQ Release nn.n
*** Report Format - Picture ***
->
137000 Specify pictures and press the ENTER key

<table>
<thead>
<tr>
<th>Seq</th>
<th>$</th>
<th>Picture</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE</td>
<td></td>
<td>X(15)</td>
</tr>
<tr>
<td>X EMP-LAST-NAME-0415</td>
<td>1</td>
<td>X(15)</td>
</tr>
<tr>
<td>X SS-NUMBER-0415</td>
<td>2</td>
<td>99-999-9999</td>
</tr>
<tr>
<td>EMPOSITION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X SALARY-GRADE-0420</td>
<td>3</td>
<td>99</td>
</tr>
<tr>
<td>X SALARY-AMOUNT-0420</td>
<td>4</td>
<td>-ZZZZZ9.99</td>
</tr>
</tbody>
</table>

Compute:

1=HELP    3=QUIT    4=MESSAGE    5=DISPLAY    6=MENU    10=EDIT    11=GROUP BY

CA OLQ displays your new picture.

CA OLQ Release nn.n
*** Display Report ***
->
125000 Press the ENTER key to go to the next page of the report.

EMPLOYEE/EMPOSITION REPORT

<table>
<thead>
<tr>
<th>NAME</th>
<th>SOCIAL SECURITY NUMBER</th>
<th>SALARY GRADE</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINGER</td>
<td>09-234-5812</td>
<td>33</td>
<td>38500.00</td>
</tr>
<tr>
<td>TERNER</td>
<td>04-567-2222</td>
<td>11</td>
<td>13000.00</td>
</tr>
<tr>
<td>BROWN</td>
<td>01-955-6712</td>
<td>44</td>
<td>42500.00</td>
</tr>
<tr>
<td>CHARLES</td>
<td>01-955-6712</td>
<td>43</td>
<td>38000.00</td>
</tr>
<tr>
<td>PENMAN</td>
<td>01-459-3186</td>
<td>33</td>
<td>39000.00</td>
</tr>
<tr>
<td>DUNCAN</td>
<td>01-067-3343</td>
<td>72</td>
<td>85000.00</td>
</tr>
<tr>
<td>EVERETT</td>
<td>01-067-3343</td>
<td>71</td>
<td>75000.00</td>
</tr>
<tr>
<td>LITERATA</td>
<td>02-356-7831</td>
<td>43</td>
<td>37500.00</td>
</tr>
<tr>
<td>WILCO</td>
<td>11-100-0823</td>
<td>72</td>
<td>80000.00</td>
</tr>
<tr>
<td>HEAROWITZ</td>
<td>03-189-6154</td>
<td>42</td>
<td>33000.00</td>
</tr>
</tbody>
</table>

1=HELP    3=QUIT    4=MESSAGE    5=DISPLAY    6=MENU    10=EDIT    11=GROUP BY
```
Step 4 - Sort Your Report Rows and Specify Group Calculations

In this step, you sort your report rows by salary grade, and compute the average salary in each salary grade. Start on the Report Format - Sort screen. To get there, type sort in the command line.

Sort your rows in ascending order by salary grade; group your report rows by salary grade. Sort within salary grade by salary amount.

CA OLQ displays the average salary for each salary grade group.
Sorting Your Report Rows

Contents

- Step 1 - Specify your sort criteria (see page 491)
- Step 2 - Display your report (see page 492)
- Step 1 - Sort your report (see page 492)
- Step 2 - Display your report (see page 493)

In this example, you use the Report Format - Sort screen to sort your report rows. To get there, type `sort` in the command line.

Step 1 - Specify your sort criteria

To sort the rows in ascending order by department ID, specify 1 in the Sort Priority column and a in the Order (A/D) column next to DEPT-ID-0410. And to sort each department's employees in ascending order by employee name, specify 2 in the Sort Priority column and an a in the Order (A/D) column next to EMP-LAST-NAME-0415.
Step 2 - Display your report

The rows are sorted in ascending order by department ID, and within each department by employee last name.

<table>
<thead>
<tr>
<th>DEPT-ID-0410</th>
<th>EMP-LAST-NAME-0415</th>
<th>SALARY-AMOUNT-0420</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>FITZHUGH</td>
<td>13000.00</td>
</tr>
<tr>
<td>1000</td>
<td>JOHNSON</td>
<td>13500.00</td>
</tr>
<tr>
<td>1000</td>
<td>ORGRATZI</td>
<td>39000.00</td>
</tr>
<tr>
<td>1000</td>
<td>PEOPLES</td>
<td>80000.00</td>
</tr>
<tr>
<td>2000</td>
<td>BLOOMER</td>
<td>15000.00</td>
</tr>
<tr>
<td>2000</td>
<td>HUTTON</td>
<td>44000.00</td>
</tr>
<tr>
<td>2000</td>
<td>JENSON</td>
<td>82000.00</td>
</tr>
<tr>
<td>2000</td>
<td>KIMBALL</td>
<td>45000.00</td>
</tr>
<tr>
<td>2000</td>
<td>KING</td>
<td>14500.00</td>
</tr>
<tr>
<td>2000</td>
<td>NICEMAN</td>
<td>14000.00</td>
</tr>
<tr>
<td>3100</td>
<td>DOUGH</td>
<td>33000.00</td>
</tr>
<tr>
<td>3100</td>
<td>GALLWAY</td>
<td>33000.00</td>
</tr>
</tbody>
</table>

Step 1 - Sort your report

Start on the Report Format - Sort screen. To get there, type SORT in the command line.

Use Disp Seq to change the column order to reflect the sort priority. Use Sort Priority to specify that you want to sort the report rows by DEPT-ID (level 1), within each department by PROJECT (level 2), and within each project by SALARY-AMOUNT (level 3). Use Order (A/D) to specify that you want all of the sorts to be in ascending order. Press [PF5] to display your report.
Display lines: Detail X and/or Summary X Group by all _

Compute:
1=HELP 3=QUIT 4=MESSAGE 5=DISPLAY 6=MENU 11=HEADER

Step 2 - Display your report

The rows are sorted by DEPT-ID, within each department by PROJECT, and within each project by SALARY-AMOUNT.

<table>
<thead>
<tr>
<th>DEPT-ID</th>
<th>PROJECT</th>
<th>SALARY-AMOUNT</th>
<th>EMP-LAST-NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>4000</td>
<td>EVALUATE</td>
<td>37000.00</td>
<td>ZEDI</td>
</tr>
<tr>
<td>PLANNING</td>
<td></td>
<td>18000.00</td>
<td>MCDougall</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25000.00</td>
<td>ANGELO</td>
</tr>
<tr>
<td>34000.00</td>
<td>JACKSON</td>
<td>18000.00</td>
<td>PENMAN</td>
</tr>
<tr>
<td>39000.00</td>
<td>JACKSON</td>
<td>25000.00</td>
<td>FINN</td>
</tr>
<tr>
<td>45000.00</td>
<td>JACKSON</td>
<td>39000.00</td>
<td>WILCO</td>
</tr>
<tr>
<td>5200</td>
<td>EVALUATE</td>
<td>31000.00</td>
<td>KASPAR</td>
</tr>
<tr>
<td>TESTING</td>
<td></td>
<td>33000.00</td>
<td>TIME</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45000.00</td>
<td>FINN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>80000.00</td>
<td>WILCO</td>
</tr>
</tbody>
</table>

END OF REPORT

1=HELP 3=QUIT 4=MESSAGE 6=MENU 10=LEFT 11=RIGHT

Editing Your Report Values

Contents
- Step 1 - Specify your edit criteria (see page 495)
- Step 2 - Display your report (see page 496)
- Additional editing features (see page 496)

In this example, you use the Report Format - Edit screen to:

- Suppress the display of repeating values in a given column
- Align the data under a column header (left-justified, center, right-justified)
- Change the order in which your columns are displayed
Compute:

1=HELP  3=QUIT  4=MESSAGE  5=DISPLAY  6=MENU  10=HEADER  11=PICTURE

Start on the Report Format - Edit screen. To get there, type EDIT in the command line of any screen and press [Enter].

**Suppressing repeating values**

When a column displays the same value for many consecutive rows, the report can become hard to read. To suppress repeating values in a column, use the **Sparse** option on the Report Format - Edit screen:

If you don't use Sparse: If you use Sparse:

**DEPT EMPLOYEE PHONE**

<table>
<thead>
<tr>
<th>DEPT</th>
<th>EMPLOYEE</th>
<th>PHONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>003</td>
<td>Sam</td>
<td>1250</td>
</tr>
<tr>
<td>003</td>
<td>Elmer</td>
<td>3000</td>
</tr>
<tr>
<td>003</td>
<td>Leo</td>
<td>4110</td>
</tr>
<tr>
<td>005</td>
<td>Jack</td>
<td>3092</td>
</tr>
<tr>
<td>005</td>
<td>Gus</td>
<td>5555</td>
</tr>
</tbody>
</table>

To restore a column to full display, type a space under **Sparse** next to that column.

**Sparse/Full**

Another way to suppress repeating values is to use the Sparse and Full options on the Session Options screen.

- **Sparse** displays only the first of a repeating set of column values.
- **Full** displays all occurrences of the repeating value.

To get to the Session Options screen, type OPTIONS in the command line and press [Enter].

**Note:** When you select the sparse option on the Report Format - Edit screen, the display of repeating values is suppressed only for the column you specify.
When you specify the Sparse/Full option on the **Session Options screen**, the display of repeating values is suppressed throughout the entire report.

**Aligning columns**

You can modify how the columns of data align under their headers by using the **Align** field. For example:

Using LEFT: Using CENTER: Using RIGHT:

```
BONUS-PERCENT BONUS-PERCENT BONUS-PERCENT
-------------- ────────────── ──────────────
03 03 03
02 02 02
06 06 06
```

To specify how you want data to align under the column headings, type LEFT, CENTER, or RIGHT under **Align** next to that column.

**Changing the column order**

Use the **Display Sequence** field to specify the order in which columns are to appear. For each column, type a number to indicate the new position for that column. For example:

```
<table>
<thead>
<tr>
<th>Disp Seq</th>
<th>NAME</th>
<th>ID</th>
<th>PHONE</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Step 1 - Specify your edit criteria**

Select **Sparse** next to DEPT-ID-0410. And to center a column’s data under its header, specify **center** for that column under **Align**.

```
Compute:
1=HELP   3=QUIT   4=MESSAGE   5=DISPLAY   6=MENU   10=HEADER   11=PICTURE
```
Step 2 - Display your report

Each department ID number is listed only once. Data in the DEPT-ID-0410 and SALARY-AMOUNT-0420 are centered under their headers.

DEPARTMENT/EMPLOYEE/EMPOSITION REPORT

<table>
<thead>
<tr>
<th>DEPT-ID-0410</th>
<th>EMP-LAST-NAME-0415</th>
<th>SALARY-AMOUNT-0420</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>FITZHUGH</td>
<td>13000.00</td>
</tr>
<tr>
<td></td>
<td>JOHNSON</td>
<td>13500.00</td>
</tr>
<tr>
<td></td>
<td>ORGRATZI</td>
<td>39000.00</td>
</tr>
<tr>
<td></td>
<td>PEOPLES</td>
<td>80000.00</td>
</tr>
<tr>
<td>2000</td>
<td>BLOOMER</td>
<td>15000.00</td>
</tr>
<tr>
<td></td>
<td>HUTTON</td>
<td>44000.00</td>
</tr>
<tr>
<td></td>
<td>JENSON</td>
<td>82000.00</td>
</tr>
<tr>
<td></td>
<td>KIMBALL</td>
<td>45000.00</td>
</tr>
<tr>
<td></td>
<td>KING</td>
<td>14500.00</td>
</tr>
<tr>
<td></td>
<td>NICEMAN</td>
<td>14000.00</td>
</tr>
<tr>
<td>3100</td>
<td>DOUGH</td>
<td>33000.00</td>
</tr>
<tr>
<td></td>
<td>GALLWAY</td>
<td>33000.00</td>
</tr>
</tbody>
</table>

Additional editing features

Using the Report Format - Edit screen, you can also:

- Display a column in hexadecimal notation
- Specify a code table used to edit a column
- Remove columns from the report (or bring back columns that you have removed)
- Assign fixed column positions in your report

Using hexadecimal notation

When you see the sign (@) in a column value, means that invalid data has been retrieved from the database. To see the actual value of the bad data, you can display the column in hexadecimal form. Once you have done this, report the value of the column to your support staff.

To display a column in hexadecimal form, type any character under Hex next to that column:

Normal Display Hexadecimal Form JOE@ X'D1D6C5AC'

To change the column from hexadecimal format back to normal display, type a space under hex next to that column.

Editing with code tables

Code tables are tables of values that have been defined in the data dictionary. CA OLQ uses these tables to translate internal codes from the database into a more meaningful external expression of the data.
Here's how the Credit Rating code table looks in the dictionary:

<table>
<thead>
<tr>
<th>Encoded Value</th>
<th>Decoded Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Credit O.K.</td>
</tr>
<tr>
<td>02</td>
<td>Rejected</td>
</tr>
<tr>
<td>03</td>
<td>Review</td>
</tr>
<tr>
<td>etc.</td>
<td>etc.</td>
</tr>
</tbody>
</table>

What to do

If your report contains a column of encoded values, you can tell CA OLQ to edit that column with a code table. Type the name of the appropriate code table under **Code Table** next to that column. You may also need to type a version number for that code table in the **Ver** entry. If you are unsure of the code table name (or version number) ask your support staff or database administrator (DBA).

Built-in code tables for columns

Some columns have code tables built into them. CA OLQ indicates these columns by displaying the following under **Code Table**:

**DICT**

If you want to use a different code table, you can type the name of the other code table over **DICT**.

If you don't want to use any code table, type spaces over **DICT**.

Removing columns from your report

To remove a column from your report, type a space or the letter d (for delete) in front of that column. For example, to remove the PHONE and BONUS columns from your report, specify:

X NAME
X ID
d PHONE
X SALARY
d BONUS

Bringing a column back

To redisplay a column that you have removed, type any character (other than the letter d) in front of that column.

⚠️ **Note:** You cannot redisplay computed columns. Once you have removed them from the report, they cannot be restored.

Fixing columns in place

You can also use the display sequence field to fix one or more columns in position on the Display Report screen. A fixed column always remains on the screen when you scroll right or left.

For example, you can fix the display of EMP-ID so that it always displays at the left margin of your report, no matter how many times you page right or left.
### DEPARTMENT/EMPLOYEE/EMPPOSITION Report

<table>
<thead>
<tr>
<th>EMP-ID</th>
<th>EMP-FIRST-NAME</th>
<th>NAME</th>
<th>EMP-CITY</th>
<th>SS-NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>0030</td>
<td>HENRIETTA</td>
<td>HENDON</td>
<td>WELLESLEY</td>
<td>011334444</td>
</tr>
<tr>
<td>0471</td>
<td>IAN</td>
<td>VALIODIS</td>
<td>SOMERVILLE</td>
<td>022887770</td>
</tr>
<tr>
<td>0001</td>
<td>JUAN</td>
<td>PAPAZEUS</td>
<td>NORTHBORO</td>
<td>022887770</td>
</tr>
<tr>
<td>0001</td>
<td>JOHN</td>
<td>RUPEE</td>
<td>METHUEN</td>
<td>013445656</td>
</tr>
<tr>
<td>0472</td>
<td>ROBBY</td>
<td>WILDER</td>
<td>SOUTHBORO</td>
<td>038779010</td>
</tr>
<tr>
<td>0069</td>
<td>JUNE</td>
<td>BLOOMER</td>
<td>LEXINGTON</td>
<td>039557818</td>
</tr>
<tr>
<td>0100</td>
<td>EDWARD</td>
<td>HUTTON</td>
<td>MELROSE</td>
<td>011223333</td>
</tr>
<tr>
<td>0011</td>
<td>RUPERT</td>
<td>JENSON</td>
<td>MELROSE</td>
<td>022347891</td>
</tr>
<tr>
<td>0067</td>
<td>MARIANNE</td>
<td>KIMBALL</td>
<td>LITTLETON</td>
<td>022778878</td>
</tr>
<tr>
<td>0106</td>
<td>DORIS</td>
<td>KING</td>
<td>MELROSE</td>
<td>067845516</td>
</tr>
<tr>
<td>0101</td>
<td>BRIAN</td>
<td>NICEMAN</td>
<td>MELROSE</td>
<td>033456110</td>
</tr>
</tbody>
</table>

#### Fixed Column Positions

- The EMP-ID column always appears at the left margin of the report.

### Salary Report

<table>
<thead>
<tr>
<th>EMP-ID</th>
<th>START-YEAR</th>
<th>SALARY-GRADE</th>
<th>SALARY</th>
<th>BONUS-PERCENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0030</td>
<td>73</td>
<td>93</td>
<td>$240,000.00</td>
<td>.010</td>
</tr>
<tr>
<td>0471</td>
<td>78</td>
<td>82</td>
<td>$100,000.00</td>
<td>.010</td>
</tr>
<tr>
<td>0471</td>
<td>78</td>
<td>72</td>
<td>$90,000.00</td>
<td>.010</td>
</tr>
<tr>
<td>0001</td>
<td>75</td>
<td>81</td>
<td>$80,000.00</td>
<td>.010</td>
</tr>
<tr>
<td>0001</td>
<td>75</td>
<td>72</td>
<td>$76,000.00</td>
<td>.010</td>
</tr>
<tr>
<td>0472</td>
<td>79</td>
<td>81</td>
<td>$90,000.00</td>
<td>.010</td>
</tr>
<tr>
<td>0069</td>
<td>80</td>
<td>13</td>
<td>$15,000.00</td>
<td>.004</td>
</tr>
<tr>
<td>0100</td>
<td>77</td>
<td>62</td>
<td>$44,000.00</td>
<td>.007</td>
</tr>
<tr>
<td>0011</td>
<td>80</td>
<td>72</td>
<td>$82,000.00</td>
<td>.010</td>
</tr>
<tr>
<td>0067</td>
<td>78</td>
<td>62</td>
<td>$45,000.00</td>
<td>.007</td>
</tr>
<tr>
<td>0106</td>
<td>80</td>
<td>12</td>
<td>$14,500.00</td>
<td>.004</td>
</tr>
<tr>
<td>0101</td>
<td>80</td>
<td>12</td>
<td>$14,000.00</td>
<td>.004</td>
</tr>
</tbody>
</table>

### Changing Your Column Relative Positions

You can change how your columns are placed relative to one another by using the `Page columns spread` option on the Session Options screen.

#### Session Options

- Current interrupt count: 100
- Current underline character:
User options:

Page columns spread: (L-Left,E-Even,M-Max,nn)

<table>
<thead>
<tr>
<th>Option</th>
<th>Current option</th>
<th>Alternate option</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>NOFiller</td>
<td>Filler</td>
</tr>
<tr>
<td>E</td>
<td>FULL</td>
<td>SPArse</td>
</tr>
<tr>
<td>E</td>
<td>HEAder</td>
<td>NOHeader</td>
</tr>
</tbody>
</table>

-> Report Processing Options <-

<table>
<thead>
<tr>
<th>Option</th>
<th>Current option</th>
<th>Alternate option</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>OLQheader</td>
<td>NOOlqheader</td>
</tr>
<tr>
<td>E</td>
<td>PICTURE</td>
<td>NOPicture</td>
</tr>
<tr>
<td>E</td>
<td>CODetable</td>
<td>NOCODetable</td>
</tr>
</tbody>
</table>

1=HELP 3=QUIT 4=MESSAGE 6=MENU 8=FWD

Even

To evenly distribute the space between your report columns, specify E (for even):

Employee/Emposition Report

```
<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOUGLAS</td>
<td>KAHALLY</td>
<td>20000.00</td>
</tr>
<tr>
<td>THEMIS</td>
<td>PAPAZEUS</td>
<td>100000.00</td>
</tr>
<tr>
<td>HARRY</td>
<td>ARM</td>
<td>46000.00</td>
</tr>
<tr>
<td>DORIS</td>
<td>KING</td>
<td>14500.00</td>
</tr>
<tr>
<td>BETH</td>
<td>CLOUD</td>
<td>52750.00</td>
</tr>
<tr>
<td>HENRIETTA</td>
<td>HENDON</td>
<td>240000.00</td>
</tr>
</tbody>
</table>
```

Left

To group your columns to the left of the page, specify L (for left):

Employee/Emposition Report

```
<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOUGLAS</td>
<td>KAHALLY</td>
<td>20000.00</td>
</tr>
<tr>
<td>THEMIS</td>
<td>PAPAZEUS</td>
<td>100000.00</td>
</tr>
<tr>
<td>HARRY</td>
<td>ARM</td>
<td>46000.00</td>
</tr>
<tr>
<td>DORIS</td>
<td>KING</td>
<td>14500.00</td>
</tr>
<tr>
<td>BETH</td>
<td>CLOUD</td>
<td>52750.00</td>
</tr>
<tr>
<td>HENRIETTA</td>
<td>HENDON</td>
<td>240000.00</td>
</tr>
</tbody>
</table>
```

Max

To display the maximum amount of space between your report columns, specify M (for max):

Employee/Emposition Report

```
<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOUGLAS</td>
<td>KAHALLY</td>
<td>20000.00</td>
</tr>
<tr>
<td>THEMIS</td>
<td>PAPAZEUS</td>
<td>100000.00</td>
</tr>
<tr>
<td>HARRY</td>
<td>ARM</td>
<td>46000.00</td>
</tr>
</tbody>
</table>
```
### Number of spaces

To display a specific number of spaces between your report columns, specify that number. For example, if you specify a 4, CA OLQ spaces the columns four spaces apart:

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOUGLAS</td>
<td>KAHALLY</td>
<td>20000</td>
</tr>
<tr>
<td>THEMIS</td>
<td>PAPAZEUS</td>
<td>100000</td>
</tr>
<tr>
<td>HARRY</td>
<td>ARM</td>
<td>46000</td>
</tr>
<tr>
<td>DORIS</td>
<td>KING</td>
<td>14500</td>
</tr>
<tr>
<td>BETH</td>
<td>CLOUD</td>
<td>52750</td>
</tr>
<tr>
<td>HENRIETTA</td>
<td>HENDON</td>
<td>240000</td>
</tr>
</tbody>
</table>

### Changing Your Column Pictures

#### Contents
- Step 1 - Modify a column picture (see page 502)
- Step 2 - Display your report (see page 503)

In this example, you use the Report Format - Picture screen to change the format of data in your report columns.

```
137002 The command you have specified is invalid for this screen

CA OLQ Release nn.n *** Report Format - Picture ***
Page 1 of 1

DEPARTMENT
X DEPT-ID-0410 1 _ _ _ 9999
EMPLOYEE
X EMP-LAST-NAME-0415 2 _ _ _ X(15)
EMPOSITION
X SALARY-AMOUNT-0420 3 _ _ _ -ZZZZZ9.99
```

Compute:

1=HELP 3=QUIT 4=MESSAGE 5=DISPLAY 6=MENU 10=EDIT 11=GROUP BY

#### What is a picture?

A picture is a code that represents how the data in a column is displayed. You can format your column data by defining a picture for that column.
For example, when numeric data appears on the report, you create different pictures to display it in different ways:

- With a dollar sign: $500
- Or without: 500
- With commas: 65,000
- Or without: 65000
- With leading zeros: 001229
- Or without: 1229

You can also combine these formats:

$65,000  001,229

**How do you specify a picture?**

To specify a column picture, use either of the following fields (but not both):

- **Select Options** -- If you want a fast, easy way to format numeric data.
- **Alter Picture** -- If you require more complex formatting for your data.

**Select Options**

These three columns let you do basic formatting of numeric data. In most cases, these columns are all you need to define pictures for your report columns. You can choose one or more of these options:

- $ displays a floating dollar sign
- , displays a comma between every three digits (left of the decimal)
- 0 displays leading zeros

**Alter picture**

This option lets you format a column by changing its picture. To change a picture, type over the default value that appears in the Alter Picture entry. For example, suppose you have a column whose picture currently looks like this:

Alter Picture
9999.99

To display a fixed dollar sign in this column, you could change the picture to this:

Alter Picture
$9999.99

**Picture symbols**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>What it Does</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Stands for a single alphanumeric character</td>
</tr>
<tr>
<td>A</td>
<td>Stands for a single alphabetic character</td>
</tr>
<tr>
<td>9</td>
<td>Stands for a single numeric character</td>
</tr>
<tr>
<td>Z</td>
<td>Stands for a numeric character and suppresses leading zeros</td>
</tr>
<tr>
<td>$</td>
<td>Stands for a numeric character and displays a floating dollar sign</td>
</tr>
<tr>
<td>.</td>
<td>Displays the decimal point in that position</td>
</tr>
</tbody>
</table>
+ Stands for a numeric character and displays + for positive values
-
- Stands for a numeric character and displays - for negative values
B Displays a blank character (a space) in that position
* Requests check protection. Leading zeros are displayed as asterisks.

**Examples**

<table>
<thead>
<tr>
<th>If you want your column to look like ...</th>
<th>and the data is stored in the database as ...</th>
<th>then specify this picture ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>123400M</td>
<td>123400M</td>
<td>X(7)</td>
</tr>
<tr>
<td>1 23400 M</td>
<td>123400M</td>
<td>XBXXXXXXXXB</td>
</tr>
<tr>
<td>JOHN</td>
<td>JOHNSON</td>
<td>A(4)</td>
</tr>
<tr>
<td>TWO WORDS</td>
<td>TWOWORDS</td>
<td>A(3)BA(5)</td>
</tr>
<tr>
<td>2350000</td>
<td>2350000</td>
<td>9(7)</td>
</tr>
<tr>
<td>2350000.00</td>
<td>2350000</td>
<td>9(7).99</td>
</tr>
<tr>
<td>$2,350,000.00</td>
<td>2350000</td>
<td>$$$,$$$,$$$9.99</td>
</tr>
<tr>
<td>23/50/000</td>
<td>2350000</td>
<td>99/99/999</td>
</tr>
<tr>
<td>120</td>
<td>00120</td>
<td>ZZZZZZ</td>
</tr>
<tr>
<td>+9876</td>
<td>9876</td>
<td>+++99</td>
</tr>
</tbody>
</table>

**Step 1 - Modify a column picture**

To add a leading dollar sign, select the $ option for SALARY-AMOUNT-0420. To insert commas, select the , option for SALARY-AMOUNT-0420.

```
CA OLQ Release nn.n
-> Specify pictures and press the ENTER key
137000

*** Report Format - Picture ***
1 of 1

<table>
<thead>
<tr>
<th>Seq</th>
<th>DEPARTMENT</th>
<th>Select</th>
<th>Options</th>
<th>Picture</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>DEPT-ID-0410</td>
</tr>
<tr>
<td>1</td>
<td>X</td>
<td></td>
<td></td>
<td>9999</td>
</tr>
<tr>
<td>X</td>
<td>EMPLOYEE</td>
<td></td>
<td></td>
<td>EMP-LAST-NAME-0415</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td></td>
<td></td>
<td>X(15)</td>
</tr>
<tr>
<td>X</td>
<td>EMPOSITION</td>
<td></td>
<td></td>
<td>EMP-SALARY-AMOUNT-0420</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>X</td>
<td></td>
<td>-ZZZZZ9.99</td>
</tr>
</tbody>
</table>
```

Compute:

1=HELP  3=QUIT  4=MESSAGE  5=DISPLAY  6=MENU  10=EDIT  11=GROUP BY
Step 2 - Display your report

125000 Press the ENTER key to go to the next page of the report.

DEPARTMENT/EMPLOYEE/EMPOSITION REPORT

DEPT-ID-0410  EMP-LAST-NAME-0415  SALARY-AMOUNT-0420

1000  FITZHUGH  $13,000.00
      JOHNSON  $13,500.00
      ORGRATZI $39,000.00
      PEOPLES  $60,000.00

2000  BLOOMER  $15,000.00
      HUTTON  $44,000.00
      JENSON  $82,000.00
      KIMBALL $45,000.00
      KING    $14,500.00
      NICEMAN $14,000.00

3100  DOUGH    $33,000.00
      GALLWAY $33,000.00

- 1 -

1=HELP   3=QUIT   4=MESSAGE   6=MENU   8=FWD   10=LEFT   11=RIGHT

Changing Column Headers

Contents
- Step 1 - Change your column headers (see page 505)
- Step 2 - Display your report (see page 505)

In this step, you use the Report Format - Header screen to:

- Change the text of the column headers that appear on the report
- Underline column headers

To get to the Report Format - Header screen, type header in the command line of any screen.

Underline character:  Disp
Seq  Header

DEPARTMENT           X DEPT-ID-0410  1 DEPT-ID-0410
EMPLOYEE

X EMP-LAST-NAME-0415  2 EMP-LAST-NAME-0415
EMPOSITION

X SALARY-AMOUNT-0420  3 SALARY-AMOUNT-0420
Default column headers

When you build a report, CA OLQ automatically creates column headers for you. These default headers come from the names of the columns in the database or from CA OLQ headers defined in the dictionary.

Changing the defaults

To change the default column headers, type over the text in the Header entry. You can change any or all of the headers in the report. Each header you type can be from 1 to 37 characters long.

Typing multiword headers

You can type headers that contain more than one word. To make a multiword header appear on a single line, type quotes around it. Otherwise, each word of the header appears on a separate line of the report. For example:

Input:

Last Name "Last Name"

Output:

Last Name Last Name
Jones Jones
Smith Smith
Whipple Whipple

Session Options That Affect Report Headers:

<table>
<thead>
<tr>
<th>Session Option</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>CA OLQ displays column headers on your report</td>
</tr>
<tr>
<td>Noheader</td>
<td>CA OLQ displays the report without headers (even if you've created new headers on the Report Format - Header screen)</td>
</tr>
<tr>
<td>Olqheader</td>
<td>CA OLQ uses any headers you've typed on the Report Format - Header screen (or that have been created in the data dictionary)</td>
</tr>
<tr>
<td>Noolqheader</td>
<td>CA OLQ ignores your headers and uses the column names from the database as the headers on the report</td>
</tr>
</tbody>
</table>

To get to the Session Options screen, type options on the command line and press [Enter].

Underlining column headers
Underlining column headers can make them easier to read. CA OLQ uses the dash (-) character to draw these lines.

To change the underline character, type the character you want to use next to **Underline Character**. If you don’t want underlining on your report, type a space here.

**Step 1 - Change your column headers**

Change your column headers to make them more legible.

```
CA OLQ Release nn.n                         *** Report Format - Header ***
->                                               Page     1 of     1
134000 Specify column headers and press the ENTER key

Underline character:                       Disp         Header

DEPARTMENT
X     DEPT-ID-0410          1 'Department Id'
EMPLOYEE
X     EMP-LAST-NAME-0415    2 Name
EMPOSITION
X     SALARY-AMOUNT-0420   3 Salary

Compute:
1=HELP    3=QUIT    4=MESSAGE    5=DISPLAY    6=MENU    10=SORT    11=EDIT
```

**Step 2 - Display your report**

```
CA OLQ Release nn.n                         *** Display Report ***
->                                               Page     1 Line     1
125000 Press the ENTER key to go to the next page of the report.

DEPARTMENT/EMPLOYEE/EMPOSITION REPORT

<table>
<thead>
<tr>
<th>DEPARTMENT ID</th>
<th>NAME</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>FITZHUGH</td>
<td>$13,000.00</td>
</tr>
<tr>
<td></td>
<td>JOHNSON</td>
<td>$13,500.00</td>
</tr>
<tr>
<td></td>
<td>ORGRATZI</td>
<td>$39,000.00</td>
</tr>
<tr>
<td></td>
<td>PEOPLES</td>
<td>$80,000.00</td>
</tr>
<tr>
<td>2000</td>
<td>BLOOMER</td>
<td>$15,000.00</td>
</tr>
<tr>
<td></td>
<td>HUTTON</td>
<td>$44,000.00</td>
</tr>
<tr>
<td></td>
<td>JENSON</td>
<td>$82,000.00</td>
</tr>
<tr>
<td></td>
<td>KIMBALL</td>
<td>$45,000.00</td>
</tr>
<tr>
<td></td>
<td>KING</td>
<td>$14,500.00</td>
</tr>
<tr>
<td></td>
<td>NICEMAN</td>
<td>$14,000.00</td>
</tr>
<tr>
<td>3100</td>
<td>DOUGH</td>
<td>$33,000.00</td>
</tr>
<tr>
<td></td>
<td>GALLWAY</td>
<td>$33,000.00</td>
</tr>
</tbody>
</table>

- 1 -
```

1=HELP    3=QUIT    4=MESSAGE    6=MENU    8=FWD    10=LEFT    11=RIGHT
Making Page Headers and Footers

Contents

- Step 1 - Specify page headers and footers (see page 509)
- Step 2 - Display your report (see page 510)

This example shows you how to use the Page Header/Footer screen to:

- Specify page header or footer text
- Include variable values (for example, the report date)
- Specify the alignment of page header or footer elements (right-justified, left-justified, or centered)
- Skip lines before and after the page header or footer
- Put more than one heading element on a line

To get to the Page Header/Footer screen, type page in the command line and press [Enter].

CA OLQ Release nn.n

-> *** Page Header/Footer ***
152000 Specify page header(s), footer(s) and press the ENTER key

Format for $DATE: MM/DD/YY
Use variables: $DATE, $TIME, $PAGE, $LINE, $USER...
Skip lines before heading: 0 Skip lines after heading: 1

Line  Page heading text Align
1    DEPARTMENT/EMPLOYEE/EMPOSITION REPORT CENTER
2    $DATE CENTER
3    CENTER
4    CENTER
5    CENTER

Skip lines before footing: 1 Skip lines after footing: 0

Line  Page footing text Align
1    - $PAGE - CENTER
2    CENTER
3    CENTER
4    CENTER
5    CENTER

1=HELP  3=QUIT  4=MESSAGE  5=DISPLAY  6=MENU

Default page headers and footers

When you build a new report, CA OLQ automatically adds a standard page header and page footer to each page of the report. The page header includes the names of the tables or records used and the current date:

EMPLOYEE REPORT

mm/dd/yy

The page footer displays the number of each report page:

- 1 -
Note: You specify page headers and footers in the same way. In this discussion, the examples will illustrate only page headers.

Specifying text

To change the text in your header, type your text over the default text under **Page heading text**:

**Input:**

<table>
<thead>
<tr>
<th>Line</th>
<th>Page heading text</th>
<th>Align</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Widget Inventory</td>
<td>center</td>
</tr>
<tr>
<td>2</td>
<td>$date</td>
<td>center</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>center</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>center</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>center</td>
</tr>
</tbody>
</table>

**Output:**

Widget Inventory

mm/dd/yy

Using variables

CA OLQ provides variables that you can include in your headers. These variables are placeholders for which CA OLQ substitutes real values (such as a page number).

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$DATE</td>
<td>Displays the current date using the user-specified date format</td>
<td>10/24/99</td>
</tr>
<tr>
<td>$EDATE</td>
<td>Displays the current date (European format)</td>
<td>24/10/99</td>
</tr>
<tr>
<td>$JDATE</td>
<td>Displays the current date (Julian format)</td>
<td>99297</td>
</tr>
<tr>
<td>$TIME</td>
<td>Displays the current time</td>
<td>10:30:59</td>
</tr>
<tr>
<td>$PAGE</td>
<td>Displays the current page number</td>
<td>- 12 -</td>
</tr>
<tr>
<td>$USER</td>
<td>Displays the user ID (of the report’s creator)</td>
<td>JFK</td>
</tr>
<tr>
<td>$LINE</td>
<td>Displays the current line number</td>
<td>33</td>
</tr>
</tbody>
</table>

Changing text alignment

To change how your header is aligned, specify LEFT, RIGHT, or CENTER under **Align**:

- **Left** left-justifies the heading text.
- **Right** right-justifies the heading text.
- **Center** centers the heading text.

For example:

**Input:**

<table>
<thead>
<tr>
<th>Line</th>
<th>Page footing text</th>
<th>Align</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Leaving blank lines before your text

To add to the top margin of your report, enter a number next to **Skip lines before heading**. If you type 0, CA OLQ will leave no blank lines before it prints the page header. To leave one or more blank lines, type a number from 1 through 9.

Leaving blank lines after your text

To provide space after your page header, enter a number next to **Skip lines after heading**. If you type 0, CA OLQ will leave no blank lines after it prints the page header. To leave one or more blank lines, type a number from 1 through 9.

Putting more than one text element on a line

To list more than one text element on a line, specify the same number in the **Line** entry for both elements. For example:

**Input:**

<table>
<thead>
<tr>
<th>Line</th>
<th>Page heading text</th>
<th>Align</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Industrial Gizmos Inventory</td>
<td>CENTER</td>
</tr>
<tr>
<td>1</td>
<td>$date</td>
<td>right</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>CENTER</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>CENTER</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>CENTER</td>
</tr>
</tbody>
</table>

**Output:**

Industrial Gizmos Inventory 10/17/99

Session Options That Affect Page Headers and Footers:

<table>
<thead>
<tr>
<th>Session Option</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>CA OLQ displays page headers on your report.</td>
</tr>
<tr>
<td>Noheader</td>
<td>CA OLQ displays the report without page headers (even if you've created new headers on the Report Format - Page Header/Footer screen).</td>
</tr>
<tr>
<td>Olqheader</td>
<td>CA OLQ uses any headers you've typed on the Report Format - Header screen (or that have been created in the data dictionary).</td>
</tr>
<tr>
<td>Nooolqheader</td>
<td>CA OLQ ignores your headers and uses the column names from the database as the headers on the report.</td>
</tr>
</tbody>
</table>

To get to the Session Options screen, type **options** on the command line of any screen and press [Enter].
Specifying the date

The **Format for $DATE**: field displays the report date format. Valid date formats are listed in the following table.

### Specifying a Date Format on the Page Header/Footer Screen:

<table>
<thead>
<tr>
<th>Format Explanation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>MONT</td>
<td>Displays the full month name in uppercase.</td>
</tr>
<tr>
<td>MONTH</td>
<td>Displays the full month name in lowercase.</td>
</tr>
<tr>
<td>MONTH</td>
<td>Displays the full month name in lowercase.</td>
</tr>
<tr>
<td>MON</td>
<td>Displays the first three letters of the month name in uppercase.</td>
</tr>
<tr>
<td>Mon</td>
<td>Displays the first three letters of the month name. The first letter is uppercase; the last two letters are lowercase.</td>
</tr>
<tr>
<td>mon</td>
<td>Displays the first three letters of the month name in lowercase.</td>
</tr>
<tr>
<td>MM</td>
<td>Displays a zero-significant numeric representation of the month.</td>
</tr>
<tr>
<td>ZM</td>
<td>Displays a zero-suppressed numeric representation of the month.</td>
</tr>
<tr>
<td>DD</td>
<td>Displays a zero-significant numeric representation of the day.</td>
</tr>
<tr>
<td>ZD</td>
<td>Displays a zero-suppressed numeric representation of the month.</td>
</tr>
<tr>
<td>YY</td>
<td>Displays a two-digit representation of the year.</td>
</tr>
<tr>
<td>YYYY</td>
<td>Displays a four-digit representation of the year.</td>
</tr>
<tr>
<td>CC</td>
<td>Displays a two-digit representation of the century.</td>
</tr>
</tbody>
</table>

### Step 1 - Specify page headers and footers

Specify the text you want displayed at the top of each page of your report and the alignment that you prefer. Add **Created by** as part of your report footer. Include the **$user** variable to display the user ID of the person who created the report.

```
CA OLQ Release nn.n
->
152000 Specify page header(s), footer(s) and press the ENTER key
Format for $DATE: MM/DD/YY
Use variables: $DATE, $TIME, $PAGE, $LINE, $USER...
Skip lines before heading: 0
Skip lines after heading: 1
```

<table>
<thead>
<tr>
<th>Line</th>
<th>Page heading text</th>
<th>Align</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Salary Report For Central Division</strong></td>
<td>CENTER</td>
</tr>
<tr>
<td>2</td>
<td>$DATE</td>
<td>right</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>CENTER</td>
</tr>
</tbody>
</table>

16-Jan-2018 509/898
Step 2 - Display your report

CA OLQ Release nn.n

*** Display Report ***

Page 1 Line 1

125000 Press the ENTER key to go to the next page of the report.

SALARY REPORT FOR CENTRAL DIVISION
07/29/99

<table>
<thead>
<tr>
<th>DEPARTMENT ID</th>
<th>NAME</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>FITZHUGH</td>
<td>$13,000.00</td>
</tr>
<tr>
<td></td>
<td>JOHNSON</td>
<td>$13,500.00</td>
</tr>
<tr>
<td></td>
<td>ORGRATZI</td>
<td>$39,000.00</td>
</tr>
<tr>
<td></td>
<td>PEOPLES</td>
<td>$80,000.00</td>
</tr>
<tr>
<td>2000</td>
<td>BLOOMER</td>
<td>$15,000.00</td>
</tr>
<tr>
<td></td>
<td>HUTTON</td>
<td>$44,000.00</td>
</tr>
<tr>
<td></td>
<td>JENSON</td>
<td>$82,000.00</td>
</tr>
<tr>
<td></td>
<td>KIMBALL</td>
<td>$45,000.00</td>
</tr>
<tr>
<td></td>
<td>KING</td>
<td>$14,500.00</td>
</tr>
<tr>
<td></td>
<td>NICEMAN</td>
<td>$14,000.00</td>
</tr>
<tr>
<td>3100</td>
<td>DOUGH</td>
<td>$33,000.00</td>
</tr>
</tbody>
</table>

CREATED BY:RMG

- 1 -

1=HELP 3=QUIT 4=MESSAGE 6=MENU 8=FWD 10=LEFT 11=RIGHT

How to Create Report Totals and Subtotals

This section shows you how to create report subtotals. The discussion is broken up into the following parts:

- **A list of key terms** used to describe report groups and subtotals
- **A quick example** of how to use CA OLQ to create subtotals
- **A step-by-step example** that walks you through the process of creating a report containing subtotals

For more information, see the following topics:

- Report Totals and Subtotals Key terms (see page 511)
- A Quick Example Creating a Subtotal (see page 512)
- A Step-By-Step Example Creating a Report With Subtotals (see page 518)
- Create Your Report (see page 519)
- Sorting Your Report Rows 1 (see page 522)
- Creating Groups and Specifying Subtotals (see page 522)
Report Totals and Subtotals Key terms

Here are a few terms used to discuss report subtotals and totals:

**Aggregate function**
A function that performs a predefined operation on a group of report rows. You can apply aggregate functions to report groups. Examples of aggregate functions are: average, high value, low value, count, and total.

**Built-in function**
A function that performs a predefined string, arithmetic, date/time, or trigonometric calculation on your report rows. You can apply built-in functions to report groups by including them in a COMPUTE statement. Examples of built-in functions are: gregorian date, cosine, and square root.

**COMPUTE statement**
A CA OLQ syntax statement used to perform calculations in the menu facility. You can apply the COMPUTE statement to report groups. Any time you specify a built-in or aggregate function, CA OLQ creates a COMPUTE statement. You can also specify your own COMPUTE statements.

**Group by all**
A report total including all rows in your report. Group by all means the same thing as report total.

**Group field**
A report column whose value is used to divide your report rows into groups. For example, you could list all of the company's employees grouped according to which department they work in. In this case, DEPARTMENT-NAME is the group field.

**Report group**
A set of report rows where each row contains the same value of the group field. For example, the personnel department is a report group with DEPARTMENT NAME as the group field. Each row in this group contains PERSONNEL in the DEPARTMENT NAME field.

**Report subtotal**
A computation applied to a report group. For example, if you grouped your employees by department, you can create report subtotals that apply to that group, computing the average salary in each department.

**Report total**
A computation that includes all of the rows in your report. For example, you can compute the total sales revenue earned by all of your employees. Note that a report total does not have to be a sum. It can also be an average, a counter, a high value, or a low value.
A Quick Example Creating a Subtotal

Contents

- The Report Format - Sort screen (see page 513)
  - Step 1 - Sort your report rows (see page 513)
  - Step 2 - Create report groups (see page 515)
- The Report Format - Group By screen (see page 516)
  - Step 3 - Specify your group calculations (see page 517)

Creating a Report Subtotal

Step 1

Sort Report Rows

Step 2

Specify Report Groups

Step 3

<table>
<thead>
<tr>
<th>Report Format - Group By screen</th>
<th>Aggregate functions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>COUNT</td>
</tr>
<tr>
<td></td>
<td>Specify Group Calculations</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The COMPUTE statement

Creating a Report Containing Subtotals:

The Boston Marathon

Suppose you were put in charge of keeping statistics for the Boston Marathon. The marathon database lists information on each of the 7,000 runners, including their name, country, time, sex, etc..

This is what your report looks like when you start out:

BOSTON MARATHON

mm/dd/yy

LAST NAME SEX FINISH TIME COUNTRY

------------------------------- --- -------------- ---------------
What you're looking for

You want to create a subtotal that lists the time of the fastest female runner from each country. A portion of your report (one subtotal) would look something like this:

<table>
<thead>
<tr>
<th>LAST NAME</th>
<th>SEX</th>
<th>COUNTRY</th>
<th>FINISH TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>VANDER WYK</td>
<td>F</td>
<td>BELGIUM</td>
<td>2:54:12</td>
</tr>
<tr>
<td>FRECETTE</td>
<td>F</td>
<td>BELGIUM</td>
<td>3:10:01</td>
</tr>
<tr>
<td>HELFGOTT</td>
<td>F</td>
<td>BELGIUM</td>
<td>3:24:31</td>
</tr>
</tbody>
</table>

--------

FASTEST IN BELGIUM : 2:54:12

What To Do

You need to perform three steps to create this subtotal:

- On the Report Format - Sort screen:
  - **Step 1** - Sort your report rows.
  - **Step 2** - Create report groups.
- On the Report Format - Group By screen:
  - **Step 3** - Specify your group calculations.

The Report Format - Sort screen

**Step 1 - Sort your report rows**

You specify how you want to sort your report rows on the Report Format - Sort screen. Your Report Format - Sort screen looks like this:

<table>
<thead>
<tr>
<th>CA OLQ Release nn.n</th>
<th>*** Report Format - Sort ***</th>
</tr>
</thead>
<tbody>
<tr>
<td>-&gt;</td>
<td>Page 1 of 1</td>
</tr>
<tr>
<td>133000 Specify sort or group by request and press the ENTER key</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Disp Seq</td>
<td>Sort Priority (A/D)</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------</td>
</tr>
<tr>
<td>OLQ-EXAMPLE</td>
<td>X LAST-NAME</td>
</tr>
<tr>
<td>X COUNTRY</td>
<td>6</td>
</tr>
<tr>
<td>X FINISH-TIME</td>
<td>5</td>
</tr>
<tr>
<td>X SEX</td>
<td>4</td>
</tr>
</tbody>
</table>
Sort Priority is used to specify the sort level. Order (A/D) is used to specify the sort order (ascending or descending).

BOSTON MARATHON  
mm/dd/yy

<table>
<thead>
<tr>
<th>LAST NAME</th>
<th>SEX</th>
<th>COUNTRY</th>
<th>FINISH TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>VANDER WYK</td>
<td>F</td>
<td>BELGIUM</td>
<td>2:54:12</td>
</tr>
<tr>
<td>FRECETTE</td>
<td>F</td>
<td>BELGIUM</td>
<td>3:10:01</td>
</tr>
<tr>
<td>HELFGOTT</td>
<td>F</td>
<td>BELGIUM</td>
<td>3:24:31</td>
</tr>
</tbody>
</table>

Sort Level 3  
Sort Field = FINISH-TIME  
Sort Priority = 3  
Sort Order = Ascending

<table>
<thead>
<tr>
<th>LAST NAME</th>
<th>SEX</th>
<th>COUNTRY</th>
<th>FINISH TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECRE</td>
<td>F</td>
<td>FRANCE</td>
<td>3:11:09</td>
</tr>
<tr>
<td>BENOTTI</td>
<td>F</td>
<td>ITALY</td>
<td>3:12:42</td>
</tr>
<tr>
<td>FUCCI</td>
<td>F</td>
<td>ITALY</td>
<td>3:40:37</td>
</tr>
<tr>
<td>BJORNSEN</td>
<td>F</td>
<td>SWEDEN</td>
<td>2:59:12</td>
</tr>
<tr>
<td>KLEIN</td>
<td>F</td>
<td>USA</td>
<td>3:09:24</td>
</tr>
</tbody>
</table>

Sort Level 2  
Sort Field = COUNTRY  
Sort Priority = 2  
Sort Order = Ascending

<table>
<thead>
<tr>
<th>LAST NAME</th>
<th>SEX</th>
<th>COUNTRY</th>
<th>FINISH TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>VANPRAAG</td>
<td>M</td>
<td>BELGIUM</td>
<td>2:45:60</td>
</tr>
<tr>
<td>MARTIN</td>
<td>M</td>
<td>FRANCE</td>
<td>3:10:29</td>
</tr>
<tr>
<td>GUERRIERI</td>
<td>M</td>
<td>ITALY</td>
<td>2:19:30</td>
</tr>
<tr>
<td>BARBADO</td>
<td>M</td>
<td>ITALY</td>
<td>2:40:12</td>
</tr>
<tr>
<td>YAMADA</td>
<td>M</td>
<td>JAPAN</td>
<td>2:50:19</td>
</tr>
<tr>
<td>KEEFE</td>
<td>M</td>
<td>USA</td>
<td>2:52:48</td>
</tr>
<tr>
<td>SIMON</td>
<td>M</td>
<td>USA</td>
<td>3:00:04</td>
</tr>
<tr>
<td>PARSONS</td>
<td>M</td>
<td>USA</td>
<td>3:00:04</td>
</tr>
</tbody>
</table>

Sort Level 1  
Sort Field = SEX  
Sort Priority = 1  
Sort Order = Ascending

Sort Levels and Sort Order:  
In this report, you want to sort your report rows by sex (to group all of the women together), by country, and within each country by race time. You must sort before you can specify any report groups.

If You Do Sort First  
Your report accurately reflects the fastest time for each country.

BOSTON MARATHON  
mm/dd/yy

<table>
<thead>
<tr>
<th>LAST NAME</th>
<th>SEX</th>
<th>COUNTRY</th>
<th>FINISH TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>VANDER WYK</td>
<td>F</td>
<td>BELGIUM</td>
<td>2:54:12</td>
</tr>
<tr>
<td>FRECETTE</td>
<td>F</td>
<td>BELGIUM</td>
<td>3:10:01</td>
</tr>
<tr>
<td>HELFGOTT</td>
<td>F</td>
<td>BELGIUM</td>
<td>3:24:31</td>
</tr>
<tr>
<td>DECRE</td>
<td>F</td>
<td>FRANCE</td>
<td>3:11:09</td>
</tr>
</tbody>
</table>

FASTEST FROM BELGIUM : 2:54:12
If You Don't Sort First

If you sort by SEX, but forget to sort by COUNTRY, CA OLQ creates a group whenever it encounters a change in the COUNTRY column. This causes inaccurate results when you try to calculate the fastest time for each country.

Why You Have to Sort First: CA OLQ groups data by looking for a change in the value of the group field (COUNTRY). If the data is not sorted in order by the group field, the changes can occur at random and produce subtotals other than the ones you intended.

Step 2 - Create report groups

You also specify your group levels on the Report Format - Sort screen. Your Report Format - Sort screen looks like this:

---

CA OLQ - 19.0

<table>
<thead>
<tr>
<th>LAST NAME</th>
<th>SEX</th>
<th>COUNTRY</th>
<th>FINISH TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>BENOTTI</td>
<td>F</td>
<td>ITALY</td>
<td>3:12:42</td>
</tr>
<tr>
<td>FUCCI</td>
<td>F</td>
<td>ITALY</td>
<td>3:40:37</td>
</tr>
<tr>
<td>VANDER WYK</td>
<td>F</td>
<td>BELGIUM</td>
<td>2:54:12</td>
</tr>
<tr>
<td>BJORNSEN</td>
<td>F</td>
<td>SWEDEN</td>
<td>2:59:12</td>
</tr>
<tr>
<td>FRECETTE</td>
<td>F</td>
<td>BELGIUM</td>
<td>3:10:01</td>
</tr>
<tr>
<td>KLEIN</td>
<td>F</td>
<td>USA</td>
<td>3:09:24</td>
</tr>
<tr>
<td>DECRE</td>
<td>F</td>
<td>FRANCE</td>
<td>3:11:09</td>
</tr>
<tr>
<td>HELFGOTT</td>
<td>F</td>
<td>BELGIUM</td>
<td>3:24:31</td>
</tr>
</tbody>
</table>

FASTEST FROM FRANCE: 3:11:09
FASTEST FROM ITALY: 3:12:42
FASTEST FROM SWEDEN: 2:59:12
FASTEST FROM BELGIUM: 3:10:01
FASTEST FROM USA: 3:09:24

---
Display lines: Detail X and/or Summary X Group by all _

Compute:
1=HELP 3=QUIT 4=MESSAGE 5=DISPLAY 6=MENU 11=HEADER

Group By Level # is used to specify group levels.

BOSTON MARATHON

mm/dd/yy

LAST NAME SEX COUNTRY FINISH TIME

VANDER WYK F BELGIUM 2:54:12
FRECETTE F BELGIUM 3:10:01
HELFGOTT F BELGIUM 3:24:31
BENOTTI F ITALY 3:12:42
FUCCI F ITALY 3:40:37
BJORNSEN F SWEDEN 2:59:12

Group Level 1
Group field = SEX

KLEIN F USA 3:09:24
VANPRAAG M BELGIUM 2:45:60
MARTIN M FRANCE 3:10:29
GUERRIERI M ITALY 2:19:30
BARBADO M ITALY 2:40:12
YAMADA M JAPAN 2:30:41
KEEFE M USA 2:40:19
SIMON M USA 2:52:48
PARSONS M USA 3:00:04

Group Levels: This report is grouped at two levels: by SEX (level 1) and by COUNTRY (level 2).

The Report Format - Group By screen

You specify your group calculations on the Report Format - Group By screen. The Report Format - Group By screen looks like this:

CA OLQ Release nn.n

*** Report Format - Group By ***

Page 1 OF 1

136000 Specify summary computations and press the ENTER key

Group by: OLQ-EXAMPLE.COUNTRY

NEXT _

Seq Sum Avg Max Min Count _

OLQ-EXAMPLE 1
X LAST-NAME 6 _ _ _ _
X COUNTRY 5 _ _ _ _
X FINISH-TIME 4 _ _ _ _
X SEX

16-Jan-2018 516/898
Step 3 - Specify your group calculations

To find the fastest time for women from each country, you need to compute the lowest race score in each country’s group. You specify this calculation by selecting the \text{Min} (minimum) function in the row across from FINISH TIME. CA OLQ automatically calculates which woman in each country has the fastest time, and lists each group’s top time.

\begin{verbatim}
BOSTON MARATHON
4/21/99
\end{verbatim}

\begin{verbatim}
VANDER WYK  F  BELGIUM  2:54:12
FRECETTE   F  BELGIUM  3:10:01
HELFGOTT   F  BELGIUM  3:24:31
\end{verbatim}

\textbf{Group Calculation} (subgroup)

Group Level = 2 FASTEST FROM BELGIUM :2:54:12
Current Group Field = COUNTRY
Aggregate Function = MIN

\begin{verbatim}
DECRE       F  FRANCE   3:11:09
BENOTTI     F  ITALY    3:12:42
FUCCI       F  ITALY    3:40:37
\end{verbatim}

\textbf{Group Calculations}: This report uses the MIN (minimum) aggregate function to calculate the lowest time in each country’s group.

When you’re done, your report looks like this:

\begin{verbatim}
BOSTON MARATHON
05/19/99
\end{verbatim}

\begin{verbatim}
VANDER WYK  F  BELGIUM  2:54:12
FRECETTE   F  BELGIUM  3:10:01
HELFGOTT   F  BELGIUM  3:24:31
\end{verbatim}

\textbf{Group Calculations}: This report uses the MIN (minimum) aggregate function to calculate the lowest time in each country’s group.

When you’re done, your report looks like this:

\begin{verbatim}
BOSTON MARATHON
05/19/99
\end{verbatim}

\begin{verbatim}
VANDER WYK  F  BELGIUM  2:54:12
FRECETTE   F  BELGIUM  3:10:01
HELFGOTT   F  BELGIUM  3:24:31
\end{verbatim}
CA IDMS - 19.0

FASTEST IN FRANCE : 3:11:09
BENOTTI F ITALY 3:12:42
FUCCI F ITALY 3:40:37

---------
FASTEST IN ITALY : 3:12:42
BJORNSEN F SWEDEN 2:59:12

--------
FASTEST IN SWEDEN : 2:59:12
KLEIN F USA 3:09:23

Aggregate functions

CA OLQ provides five predefined calculations that you can apply to your report groups. These calculations are called aggregate functions. They are:

- Count counts the number of rows in the report group.
- Total adds up occurrences of numeric columns in the report group.
- Average finds the average of the specified column in the report group.
- Maximum finds the highest value of the specified column in the report group.
- Minimum finds the lowest value of the specified column in the report group.

How to use aggregate functions

Aggregate functions are listed on the Report Format - Group By screen. To use them:

1. Make sure that the current group field is the one you want. The current group field is listed in the Group by field.
2. Select an aggregate function corresponding to the appropriate report column.

A Step-By-Step Example Creating a Report With Subtotals

In the rest of this section, you will create a series of reports that contain subtotals. Each report uses an additional CA OLQ grouping feature to enhance the original report.

You will first:

1. Create the original report, using the EMPLOYEE data table. Your enhancements will then:
2. Sort the rows of the report.
3. Specify report groups, arranging the employees according to which department they work in.
4. Compute a report subtotal, finding the average salary of each department.
5. Compute a report total, finding the total amount of money paid in employee salaries.

6. Create nested subtotals, listing the number of employees working on each project within each department.

This is how your final report looks after you have added all of the grouping enhancements:

```
CA OLQ Release nn.n                          *** Display Report ***
->                                      Page 3 Line 25
125004 Press the ENTER key for DISPLAY/FORMAT ACTIVITY selections

EMPLOYEE REPORT
mm/dd/yy

-------------------------------  -------------------------  ---------------
PROJECT                        SALARY-AMOUNT    EMP-LAST-NAME
------------------------------  -------------------------  ---------------
RESEARCH                      45000.00        FINN
80000.00                      WILCO

COUNT FOR RESEARCH:   3

AVE FOR 5200 : 45400.00

=================
453000.00

END OF REPORT
```

Report rows are sorted by DEPT-ID, PROJECT, and SALARY-AMOUNT. Within each department, employees are grouped by which project they work on. The average salary for each department is listed. The total amount paid in employee salaries is listed at the bottom of the report.

Create Your Report

Contents

- Step 1 - Set your session options (see page 519)
- Step 2 - Select your table (see page 520)
- Step 3 - Select columns (see page 520)
- Step 4 - Retrieve your data (see page 521)
- Step 5 - Display your report (see page 521)

Step 1 - Set your session options

Specify N. In this example, you will use ASF tables. Change the Full/Sparse option to Sparse. This option makes reports containing groups easier to read.

```
CA OLQ Release nn.n                          *** Session Options ***
->                                      Page 1 of 2
141000 Select options to be changed and press the ENTER key
Current interrupt count:    100        Current underline character: -
User options:                Page columns spread: E (L-Left,E-Even,M-Max,nn)
Change
```
**Step 2 - Select your table**

Specify SELECT. Select the EMPLOYEE data table.

**Step 3 - Select columns**

Select EMP-LAST-NAME, DEPT-ID, SALARY-AMOUNT, and PROJECT.

Additional selection criteria:
Step 4 - Retrieve your data

Depending on how high the interrupt count at your site is set, CA OLQ may bypass the Retrieval Interrupted screen and proceed directly to the Retrieval Completed screen.

CA OLQ Release nn.n
*** Retrieval Completed ***
->
130000 Select activity and press the ENTER key

Number of whole rows . . . . . . . 11 Total displayable cols . 20
Total number of records read . . . 11 Formatted line length . . . 372
Total number of records selected . 11
Number of data errors . . . . . . . 0

Select
Option ---> Display/Format Activity <--- Command/
Screen Name
X Display report DISplay
_ Save report SAVE
_ Choose the sort sequence of report SORt
_ Change column headers HEAder
_ Change page header and footer PAGE HEAder
_ Change display format of data ($, commas) PICTure
_ Format columns (Alignment, sparse) EDIT
_ Specify summary computations (Totals) GROup BY
_ Send the report to a printer PRInt

1=HELP 3=QUIT 4=MESSAGE 6=MENU

Step 5 - Display your report

At this point, the data in your report is listed in the order in which it has been retrieved from the database.

CA OLQ Release nn.n
*** Display Report ***
1 + Page 1 Line 1
125004 Press the ENTER key for DISPLAY/FORMAT ACTIVITY selections

EMPLOYEE REPORT
mm/dd/yy
EMP-LAST-NAME DEPT-ID SALARY-AMOUNT PROJECT
--------------- ------- ------------- --------
BANK 4000 80000.00 TESTING
ANGELO 4000 18000.00 PLANNING
MCDUGALL 4000 18000.00 PLANNING
PENMAN 4000 39000.00 PLANNING
JACKSON 4000 34000.00 PLANNING
FINN 5200 45000.00 RESEARCH
WILCO 5200 60000.00 RESEARCH
TIME 5200 33000.00 RESEARCH
KASPAR 5200 31000.00 EVALUATE
ZEDI 4000 37000.00 EVALUATE
CLOTH 5200 38000.00 EVALUATE
END OF REPORT

- 1 -
1=HELP 3=QUIT 4=MESSAGE 6=MENU 10=LEFT 11=RIGHT
Sorting Your Report Rows 1

In this step, you sort the rows of the report on three different levels:

- By DEPT-ID
- Within each department group, by PROJECT
- Within each project group, by EMP-LAST-NAME

Creating Groups and Specifying Subtotals

Contents

- Step 1 - Specify your group level (see page 522)
- Step 2 - Specify your subtotal (see page 523)
- Step 3 - Display your report (see page 523)

In this step, you find the average salary of each department by specifying DEPT-ID as a group field and applying a calculation to that group. To do this:

1. **Sort** the rows of your report by department. You just did this in the last step of this example.

2. **Group** the employees according to which department they work in.

3. **Calculate** the average salary amount for that group.

Step 1 - Specify your group level

In this step, you calculate the average salary in each department.

Start on the Report Format - Sort screen. To get there, type SORT in the command line.

Specify that you want to group your employees by department by entering a 1 next to **Group By Level #**. Specifying 1 means that DEPT-ID is the highest level group field. You may later create subgroups within each department.
**Step 2 - Specify your subtotal**

*Group by* displays EMPLOYEE.DEPT-ID as the current group field. *Level#* displays 1 as the current group level. Use *Avg* to specify that you want CA OLQ to compute the average salary for each department.

```
CA OLQ Release nn.n
*** Report Format - Group By ***
->
136000 Specify summary computations and press the ENTER key
```

```
Group by: EMPLOYEE.DEPT-ID

<table>
<thead>
<tr>
<th>Seq</th>
<th>Sum</th>
<th>Avg</th>
<th>Max</th>
<th>Min</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE</td>
<td>EMP-LAST-NAME</td>
<td>4</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td></td>
<td>DEPT-ID</td>
<td>1</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td></td>
<td>SALARY AMOUNT</td>
<td>3</td>
<td>_</td>
<td>X</td>
<td>_</td>
</tr>
<tr>
<td></td>
<td>PROJECT</td>
<td>2</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
</tbody>
</table>
```

```
CA OLQ generates this COMPUTE statement to calculate the average salary for each department.
```

```
CA OLQ Release nn.n
*** Report Format - Sort ***
->
133000 Specify sort or group by request and press the ENTER key
```

```
Disp  | Sort  | Order  | Group By
Seq | Priority | (A/D) | Level #
EMPLOYEE | EMP-LAST-NAME | 4 |
| DEPT-ID | 1 | 1 |
| SALARY-AMOUNT | 3 | 3 |
| PROJECT | 2 | 2 |

```

```
COMPUTE FIELDS:
X 'EMPLOYEE.SALARY-AMOUNT-AVE-2'=AVE(E
MPLOYEE.SALARY-AMOUNT) GROUP BY
EMPLOYEE.DEPT-ID LEVEL 1
```

**Step 3 - Display your report**

CA OLQ displays the average salary for each department.
CA OLQ Release nn.n

*** Display Report ***

Press the ENTER key to go to the next page of the report.

125000

EMPLOYEE REPORT

DEPT-ID PROJECT SALARY-AMOUNT EMP-LAST-NAME
------- -------- ------------------------- ---------------
4000 EVALUATE 37000.00 ZEDI

PLANNING 18000.00 ANGELO

ANGEL 34000.00 JACKSON

TESTING 39000.00 PENMAN

80000.00 BANK

AVE FOR 4000 : 37666.66

5200 EVALUATE 31000.00 KASPAR

RESEARCH 38000.00 CLOTH

TIME 33000.00

CREATE REPORT TOTALS

Creating Report Totals

Contents

- Step 1 - Specify your group level (see page 524)
- Step 2 - Specify your report total (see page 525)
- Step 3 - Display your report (see page 525)

In this step, you find the total amount of money paid in employee salaries. Because this sum includes all of the rows in the report, you use the Group by all field.

Step 1 - Specify your group level

Start on the Report Format - Sort screen. To get there, type SORT in the command line.

Select Group by all to include all of the rows in your report in a group computation.

CA OLQ Release nn.n

*** Report Format - Sort ***

133000 Specify sort or group by request and press the ENTER key

Display lines: Detail X and/or Summary X Group by all x

1=HELP 3=QUIT 4=MESSAGE 6=MENU 8=FWD 10=LEFT 11=RIGHT
Step 2 - Specify your report total

**Group by** shows you that you are including all of the rows in the report in your computation. Specify that you want to know the sum of all the employee salaries in the report. Use **Seq** to specify the column under which you want the computed field displayed.

```
136000  Specify summary computations and press the ENTER key
```

<table>
<thead>
<tr>
<th>Seq</th>
<th>Sum</th>
<th>Avg</th>
<th>Max</th>
<th>Min</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE</td>
<td>X</td>
<td>EMP-LAST-NAME</td>
<td>4</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>X</td>
<td>DEPT-ID</td>
<td>1</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>X</td>
<td>SALARY-AMOUNT</td>
<td>3</td>
<td>X</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>X</td>
<td>PROJECT</td>
<td>2</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
</tbody>
</table>

**COMPUTE FIELDS:**

```
X 'EMPLOYEE.SALARY-AMOUNT-AVE-2'=AVE(EMPLOYEE.SALARY-AMOUNT) GROUP BY EMPLOYEE.DEPT-ID LEVEL 1
```

Skip lines after group 1  
Separator character =

Compute:

```
1=HELP  3=QUIT  4=MESSAGE  5=DISPLAY  6=MENU  10=PICTURE
```

CA OLQ generates this **COMPUTE statement** to calculate the report total.

```
133000  Specify sort or group by request and press the ENTER key
```

<table>
<thead>
<tr>
<th>Seq</th>
<th>Priority</th>
<th>Order (A/D)</th>
<th>Group By Level #</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE</td>
<td>X</td>
<td>EMP-LAST-NAME</td>
<td>4</td>
</tr>
<tr>
<td>X</td>
<td>DEPT-ID</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>X</td>
<td>SALARY-AMOUNT</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>X</td>
<td>PROJECT</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

**COMPUTE FIELDS:**

```
X 'EMPLOYEE.SALARY-AMOUNT-AVE-2'=AVE(EMPLOYEE.SALARY-AMOUNT) GROUP BY EMPLOYEE.DEPT-ID LEVEL 1
```

```
X TOTAL-3-ALL=TOTAL(EMPLOYEE.SALARY-AMOUNT) GROUP BY ALL LEVEL 1
```

Display lines: Detail X and/or Summary X  
Group by all X

Compute:

```
1=HELP  3=QUIT  4=MESSAGE  5=DISPLAY  6=MENU  11=HEADER
```

Step 3 - Display your report

CA OLQ displays the total amount spent in employee salaries.
125004 Press the ENTER key for DISPLAY/FORMAT ACTIVITY selections

EMPLOYEE REPORT

DEPT-ID PROJECT SALARY-AMOUNT EMP-LAST-NAME
------- -------- ------------------------- --------------
5200 EVALUATE 31000.00 KASPAR
38000.00 CLOTH
RESEARCH 33000.00 TIME
45000.00 FINN
80000.00 WILCO

AVE FOR 5200 : 45400.00

END OF REPORT

Creating Levels of Nested Subtotals

Contents
- Step 1 - Specify your group level (see page 526)
- Step 2 - Specify your subtotal (see page 527)
- Step 3 - Display your report (see page 528)

In this step, you list the number of employees in each project group. For example, you can find out how many people are currently on your department's planning committee.

Because each department has a planning committee, this kind of calculation requires two group levels:

- **Group level 1** groups the employees according to which department they work in. DEPT-ID is the group field.
- **Group level 2** groups the employees within each department according to which project they are currently assigned to. PROJECT is the group field.

**Step 1 - Specify your group level**

Start on the Report Format - Sort screen. To get there, type **Sort** in the command line.

Specify that you want to group by PROJECT within each department group.

---

Creating Levels of Nested Subtotals

Contents
- Step 1 - Specify your group level (see page 526)
- Step 2 - Specify your subtotal (see page 527)
- Step 3 - Display your report (see page 528)

In this step, you list the number of employees in each project group. For example, you can find out how many people are currently on your department's planning committee.

Because each department has a planning committee, this kind of calculation requires two group levels:

- **Group level 1** groups the employees according to which department they work in. DEPT-ID is the group field.
- **Group level 2** groups the employees within each department according to which project they are currently assigned to. PROJECT is the group field.

**Step 1 - Specify your group level**

Start on the Report Format - Sort screen. To get there, type **Sort** in the command line.

Specify that you want to group by PROJECT within each department group.

---

Creating Levels of Nested Subtotals

Contents
- Step 1 - Specify your group level (see page 526)
- Step 2 - Specify your subtotal (see page 527)
- Step 3 - Display your report (see page 528)

In this step, you list the number of employees in each project group. For example, you can find out how many people are currently on your department's planning committee.

Because each department has a planning committee, this kind of calculation requires two group levels:

- **Group level 1** groups the employees according to which department they work in. DEPT-ID is the group field.
- **Group level 2** groups the employees within each department according to which project they are currently assigned to. PROJECT is the group field.

**Step 1 - Specify your group level**

Start on the Report Format - Sort screen. To get there, type **Sort** in the command line.

Specify that you want to group by PROJECT within each department group.
Step 2 - Specify your subtotal

**Group by** displays EMPLOYEE.PROJECT as the current group level. **Level#** displays 2 as the current group level. Specify that you want to count the number of employees within each project.

CA OLQ generates this COMPUTE statement to count the total number of employees working on each project.

CA OLQ Release nn.n

136000 Specify summary computations and press the ENTER key

Group by: EMPLOYEE.PROJECT

<table>
<thead>
<tr>
<th>Seq</th>
<th>Sum</th>
<th>Avg</th>
<th>Max</th>
<th>Min</th>
<th>Count</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Compute:

Compute:

CA OLQ Release nn.n

136000 Specify summary computations and press the ENTER key

Group by: EMPLOYEE.PROJECT

<table>
<thead>
<tr>
<th>Seq</th>
<th>Sum</th>
<th>Avg</th>
<th>Max</th>
<th>Min</th>
<th>Count</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Compute:

CA OLQ generates this COMPUTE statement to count the total number of employees working on each project.

CA OLQ Release nn.n

136000 Specify summary computations and press the ENTER key

Group by: EMPLOYEE.PROJECT

<table>
<thead>
<tr>
<th>Seq</th>
<th>Sum</th>
<th>Avg</th>
<th>Max</th>
<th>Min</th>
<th>Count</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Compute:
Step 3 - Display your report

CA OLQ lists the number of employees assigned to each project.

EMPLOYEE REPORT
mm/dd/yy

PROJECT | SALARY-AMOUNT | EMP-LAST-NAME
---------|---------------|---------------
RESEARCH | 45000.00      | FINN
80000.00 | WILCO

COUNT FOR RESEARCH: 3

AVE FOR 5200 : 45400.00

453000.00

END OF REPORT

How To Format Reports Containing Calculations

In this section This section shows you how to enhance the appearance of groups and group calculations in your CA OLQ report. Using CA OLQ, you can:

- Specify how many lines you want to skip after a group
- Specify which character you want to use to set apart computations
- Tailor your computation headings
- Display only lines containing computations
- Display all lines except those containing computations
- Skip pages after a group

The report shown below illustrates some of the ways you can format your computations using CA OLQ.
CA IDMS - 19.0

CLOTH 38000.00
KASPAR 31000.00

FINN 45000.00
TIME 33000.00
WILCO 80000.00

AVERAGE SALARY: 45400.00

TOTAL SPENT IN SALARIES: 453000.00

Key Terms

Here are a few terms used to discuss formatting report groups and calculations:

Detail line
A report line that displays a row of data retrieved from the database. A detail line can contain a computed column, but does not contain a group computation.

Group computation
A calculation that CA OLQ performs on a report group.

Separator character
A character used to separate group computations from the rest of the report.

Summary computation
Another term for group computation.

Summary line
A report line that displays a group computation.

Create Your Reports

In this sample, you format the report you created in here (see page 510). To build your current report, follow the steps outlined in Section 6.

CA OLQ Release nn.n
*** Display Report ***
->
Page 1 Line 1
125000 Press the ENTER key to go to the next page of the report.

EMPLOYEE REPORT

DEPT-ID PROJECT EMP-LAST-NAME
4000 EVALUATE ZEDI
COUNT FOR EVALUATE: 1
PLANNING ANGELO
Skipping Lines After Groups

In this step, you specify how many blank lines you want CA OLQ to display between report groups.

You can indicate a different spacing for each group level. If your report contains more than one group level, you must establish the current group field and group level before you can indicate how many lines you want to skip.

<table>
<thead>
<tr>
<th>Step</th>
<th>What To Do</th>
<th>How To Do It</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Proceed to the Report Format - Group By screen.</td>
<td>Type GROUP BY in the command line of any screen.</td>
</tr>
<tr>
<td>2</td>
<td>Establish the current group field and level.</td>
<td>Type the group field next to Group by: Type the corresponding level number in the Level# field. Press [Enter].</td>
</tr>
<tr>
<td>3</td>
<td>Specify how many lines you want displayed after the current group.</td>
<td>Enter the number next to Skip lines after group.</td>
</tr>
</tbody>
</table>

If you're not sure how to specify your group field and level, they are listed in the COMPUTE statement on the Report Format - Group By screen. For example:

If your COMPUTE statement reads:

```plaintext
'EMPLOYEE.PROJECT-COUNT'=COUNT
GROUP BY EMPLOYEE.PROJECT LEVEL 2'
```

Your group field is:

- EMPLOYEE.PROJECT

Your group level is:

- 2

Specify `employee.project` as the current group field. Specify 2 as the current group level.

```plaintext
CA OLQ Release nn.n -> *** Report Format - Group By *** 1 OF 2
136000 Specify summary computations and press the ENTER key
Group by: employee.project

<table>
<thead>
<tr>
<th>Seq</th>
<th>Sum</th>
<th>Avg</th>
<th>Max</th>
<th>Min</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>1</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>4</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>2</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
</tbody>
</table>

COMPUTE FIELDS:
- X TOTAL-3-ALL=TOTAL(EMPLOYEE.SALARY-AM
```
Group by: **EMPLOYEE.PROJECT**

<table>
<thead>
<tr>
<th>Seq</th>
<th>Sum</th>
<th>Avg</th>
<th>Max</th>
<th>Min</th>
<th>Count</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMP-LAST-NAME</td>
<td>3</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td></td>
</tr>
<tr>
<td>DEPT-ID</td>
<td>1</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td></td>
</tr>
<tr>
<td>SALARY-AMOUNT</td>
<td>4</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td></td>
</tr>
<tr>
<td>PROJECT</td>
<td>2</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td></td>
</tr>
</tbody>
</table>

**Compute:**

- TOTAL-3-ALL=TOTAL(EMPLOYEE.SALARY-AMOUNT) GROUP BY ALL LEVEL 1
- 'EMPLOYEE.SALARY-AMOUNT-AVE-2'=AVE(EMPLOYEE.SALARY-AMOUNT) GROUP BY EMPLOYEE.DEPT-ID LEVEL 1

CA OLQ doesn't insert any blank lines between project groups.

**Display Report**

- DEPT-ID: 4000
- PROJECT: EVALUATE
- EMP-LAST-NAME: ZEDI

**Count for Evaluate:** 1

- PLANNING: ANGELO
- JACKSON
- MCDougALL
- PENMAN

**Count for Planning:** 4

- TESTING: BANK

**Count for Testing:** 1

```plaintext
1=HELP  3=QUIT  4=MESSAGE  5=DISPLAY  6=MENU  8=FWD  10=PICTURE
```
Specifying a SCparator character

In this step, you specify which character you want to use to set your group calculations apart from the rest of your report.

CA OLQ provides lines separating your group calculations from the detail lines in your report. The default separator characters that make up these lines are:

- A hyphen (-) for subtotals
- An equal sign (=) for report totals

You can change these default separator characters to modify your report’s appearance.

Each group level's separator character is modified individually. Before you can specify a separation character, you must establish a current group field and level.

<table>
<thead>
<tr>
<th>Step</th>
<th>What To Do</th>
<th>How To Do It</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Go to the Report Format - Group By screen.</td>
<td>Type <strong>group by</strong> in the command line of any screen.</td>
</tr>
<tr>
<td>2</td>
<td>Establish the current group field and level.</td>
<td>Type the group field in the Group by entry. Type the corresponding level number in the Level # field. Press [Enter].</td>
</tr>
<tr>
<td>3</td>
<td>Specify your separator character.</td>
<td>Type the character you want in the Separator character field.</td>
</tr>
</tbody>
</table>

Specify **all** as your current group field. Specify **1** as the current group level.

CA OLQ Release nn.n

Specify summary computations and press the ENTER key

**Group by: all**

<table>
<thead>
<tr>
<th>Seq</th>
<th>Sum</th>
<th>Avg</th>
<th>Max</th>
<th>Min</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE</td>
<td>X EMP-LAST-NAME</td>
<td>3</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>X DEPT-ID</td>
<td>1</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>X SALARY-AMOUNT</td>
<td>4</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>X PROJECT</td>
<td>2</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
</tbody>
</table>

**COMPUTE FIELDS:**

X TOTAL=TOTAL(EMPLOYEE.SALARY-AMOUNT) GROUP BY ALL LEVEL 1

X 'EMPLOYEE.SALARY-AMOUNT-AVE=AVE(E MPLOYEE.SALARY-AMOUNT) GROUP BY EMPLOYEE.DEPT-ID LEVEL 1

Skip lines after group 1

Separator character -

Compute:

1=HELP 3=QUIT 4=MESSAGE 5=DISPLAY 6=MENU 8=FwD 10=P IC TURE
CA OLQ echoes the screen, listing the current group field and level. Specify that you want the underscore (_) to set apart the report total.

<table>
<thead>
<tr>
<th>Group by: ALL</th>
<th>Seq</th>
<th>Sum</th>
<th>Avg</th>
<th>Max</th>
<th>Min</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE</td>
<td>3</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>X EMP-LAST-NAME</td>
<td>1</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>X DEPT-ID</td>
<td>4</td>
<td>X</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>X PROJECT</td>
<td>2</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
</tbody>
</table>

**COMPUTE FIELDS:**

- TOTAL-3-ALL=TOTAL(EMPLOYEE.SALARY-AMOUNT) GROUP BY ALL LEVEL 1
- 'EMPLOYEE.SALARY-AMOUNT-AVE-2'=AVE(EMPLOYEE.SALARY-AMOUNT) GROUP BY EMPLOYEE.DEPT-ID LEVEL 1

You may have to scroll forward (PF8) to view this computation. Underscores separate the report total computation from the preceding detail lines.

**Giving Your Computation a Heading**

In this step, you modify the default headers for your group computations.

Computations with ALL as the group field are not assigned a default header. All other group computations are assigned default headers. These headers include:
A literal that labels the report group, consisting of a dollar sign ($) plus the group field name

Any built-in or aggregate functions included in that group computation

For example, the default header for the average salary for each department is:

'AVE FOR $EMPLOYEE.DEPT-ID'

This header would look like this for department 5200:

AVE FOR 5200 :

The default headers are listed on the Report Format - Header screen. To get there, type header in the command line of any screen.

Underline character: -

<table>
<thead>
<tr>
<th>Disp</th>
<th>Seq</th>
<th>Header</th>
</tr>
</thead>
</table>
| EMPLOYEE
X EMP-LAST-NAME 3 EMP-LAST-NAME
X DEPT-ID 1 DEPT-ID
X SALARY-AMOUNT 4 SALARY-AMOUNT
X PROJECT 2 PROJECT

Compute fields:

X TOTAL-3-ALL=TOTAL(EMPLOYEE.SALARY-AMOUNT) GROUP BY ALL LEVEL 1
X 'EMPLOYEE.SALARY-AMOUNT-AVE-2'=AVE(EMPLOYEE.SALARY-AMOUNT) GROUP BY EMPLOYEE.DEPT-ID LEVEL 1

Specify more legible labels for your report total and department computations.

Underline character: -

<table>
<thead>
<tr>
<th>Disp</th>
<th>Seq</th>
<th>Header</th>
</tr>
</thead>
</table>
| EMPLOYEE
X EMPLOYEE.PROJECT-COUNT=COUNT 'COUNT FOR $EMPLOYEE.PROJECT:' GROUP BY EMPLOYEE.PROJECT LEVEL 2

Specify more legible labels for your report total and department computations.
134000 Specify column headers and press the ENTER key

Underline character: -

<table>
<thead>
<tr>
<th>Disp</th>
<th>Seq</th>
<th>Header</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE</td>
<td>EMP-LAST-NAME</td>
<td>3</td>
</tr>
<tr>
<td>DEPT-ID</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SALARY-AMOUNT</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>PROJECT</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

COMPUTE FIELDS:
X TOTAL-3-ALL=TOTAL(EMPLOYEE.SALARY-AMOUNT) GROUP BY ALL LEVEL 1

"TOTAL SPENT IN SALARIES:"

X 'EMPLOYEE.SALARY-AMOUNT-AVE-2'=AVE(EMPLOYEE.SALARY-AMOUNT) GROUP BY EMPLOYEE.DEPT-ID LEVEL 1

"AVERAGE SALARY:"

Compute:
1=HELP   3=QUIT   4=MESSAGE   5=DISPLAY   6=MENU   8=FWD   10=SORT   11=EDIT

Specify a more legible label for your project computation. In this example, the label contains a variable field ($EMPLOYEE.PROJECT) for the project name.

CA OLQ displays a new label for your project groups.

CA OLQ Release nn.n *** Display Report ***

125000 Press the ENTER key to go to the next page of the report.

CA OLQ Release nn.n *** Display Report ***

->

125000 Press the ENTER key to go to the next page of the report.

DEPT-ID      PROJECT      EMP-LAST-NAME
4000          EVALUATE    ZEDI

WORKING ON EVALUATE: 1
PLANNING       ANGELO
JACKSON
MCDougall
PENMAN
CA IDMS - 19.0

WORKING ON PLANNING: 4
TESTING BANK

WORKING ON TESTING: 1

CA OLQ displays a new label for your department computations and for your report total.

CA OLQ Release nn.n
-> *** Display Report ***
125004 Press the ENTER key for DISPLAY/FORMAT ACTIVITY selections

EMPLOYEE REPORT
mm/dd/yy

EMP-LAST-NAME SALARY-AMOUNT
CLOTH 38000.00
KASPAR 31000.00
FINN 45000.00
TIME 33000.00
WILCO 80000.00

AVERAGE SALARY: 45400.00

TOTAL SPENT IN SALARIES: 453000.00

Note that if you lengthen your header, the column width lengthens accordingly. If you make your headers too long, this can make your report difficult to read.

Displaying Only Summary Computations

In this step, you modify your report definition so that it displays only those report rows that contain summary computations.

Start on the Report Format - Sort screen. To get there, type sort on the command line of any screen.

Type a blank next to Detail to suppress the display of detail lines.

Displaying Only Summary Computations

In this step, you modify your report definition so that it displays only those report rows that contain summary computations.

Start on the Report Format - Sort screen. To get there, type sort on the command line of any screen.

Type a blank next to Detail to suppress the display of detail lines.
Displaying Only Detail Lines

In this step, you modify your report definition so that it displays only detail lines (those lines that do not contain any summary computations).

Start on the Report Format - Sort screen. To get there, type `sort` on the command line of any screen. Type a character next to `Detail` to restore the display of detail lines. Type a blank next to `Summary` to suppress the display of rows containing summary computations.
Compute:
1=HELP 3=QUIT 4=MESSAGE 5=DISPLAY 6=MENU 8=FWD 11=HEADER

CA OLQ displays only detail lines.

CA OLQ Release nn.n

*** Display Report ***

Page 1 Line 1

DISPLAY RIGHT to see more report columns

125004 Press the ENTER key for DISPLAY/FORMAT ACTIVITY selections

EMPLOYEE REPORT

mm/dd/yy

DEPT-ID SALARY-AMOUNT EMP-LAST-NAME

4000 37000.00 ZEDI
    18000.00 ANGELO
    34000.00 JACKSON
    18000.00 McDougall
    39000.00 Penman
    80000.00 Bank

5200 38000.00 Cloth
    31000.00 Kaspar
    45000.00 Finn
    33000.00 Time
    80000.00 Wilco

END OF REPORT

1=HELP 3=QUIT 4=MESSAGE 6=MENU 10=LEFT 11=RIGHT

Skipping To a New Page After Computations

In this step, you request that CA OLQ skip to a new page after each change in the group field.

Before you specify this option, you must establish a current group field and group level.

<table>
<thead>
<tr>
<th>Step</th>
<th>What To Do</th>
<th>How To Do It</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Go to the Report Format - Group By screen.</td>
<td>Type group by in the command line of any screen.</td>
</tr>
</tbody>
</table>
| 2    | Establish the current group field and level. | Type the group field in the Group by field.
Type the corresponding level number in the Level# field.
Press [Enter]. |
| 3    | Ask CA OLQ to skip to a new page after a computation. | Enter pg in the Skip lines after group field. |

Specify employee.dept-id as your current group field. Specify 1 as the current group level.

CA OLQ Release nn.n

*** Report Format - Group By ***

Page 1 OF 2

136000 Specify summary computations and press the ENTER key

Group by: employee.dept-id

<table>
<thead>
<tr>
<th>Seq</th>
<th>Sum</th>
<th>Avg</th>
<th>Max</th>
<th>Min</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE X EMP-LAST-NAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X DEPT-ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X SALARY-AMOUNT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X PROJECT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
COMPUTE FIELDS:
X TOTAL-3-ALL=TOTAL(EMPLOYEE.SALARY-AMOUNT) GROUP BY ALL LEVEL 1
X 'EMPLOYEE.SALARY-AMOUNT-AVE-2'=AVE(EMPLOYEE.SALARY-AMOUNT) GROUP BY EMPLOYEE.DEPT-ID LEVEL 1

Skip lines after group 1
Separator character -

Compute:

Type sort in the command line. CA OLQ echoes the screen, listing the current group field and level. Type pg next to Skip lines after group to specify that you want CA OLQ to skip a page after each change in the current group field (DEPT-ID).

Display lines: Detail X and/or Summary X Group by all X
CA OLQ displays each department's statistics on a separate page. Press [PF8] to page through the whole report.

CA OLQ displays each department's statistics on a separate page. Press [PF8] to page through the whole report.

 EMPLOYEE REPORT
---
DEPT-ID | SALARY-AMOUNT | EMP-LAST-NAME
---|---|---
4000 | 37000.00 | ZEDI
18000.00 | ANGELO
34000.00 | JACKSON
18000.00 | MCDougALL
39000.00 | PENMAN
80000.00 | BANK
AVERAGE SALARY: 37666.66

CA OLQ displays each department's statistics on a separate page. Press [PF8] to page through the whole report.

 EMPLOYEE REPORT
---
DEPT-ID | SALARY-AMOUNT | EMP-LAST-NAME
---|---|---
5200 | 38000.00 | CLOTH
31000.00 | KASPAR
45000.00 | FINN
33000.00 | TIME
80000.00 | WILCO
AVERAGE SALARY: 45400.00

---

How to Save a Set of Commands as a Qfile

What is a qfile?

A qfile is a way to save a set of CA OLQ commands in a file.

What do you do with qfiles?

You can execute a qfile to retrieve data from the database and create a report. The qfile formats the report the same way each time it is executed, but the data in the report changes to reflect the current data in the database.
Example

You could create a qfile that displays quarterly financial information. When you execute the routine at the end of each fiscal quarter, the report looks the same each time, but the financial information changes.

Creating and Executing the QUARTER Qfile:

Create:
1. Use OLQ to create a quarterly qfile.
2. Use the qfile processing screen to save these commands in IDD as the QUARTER qfile.

Execute:
Each fiscal quarter execute the QUARTER qfile to create a new report.

Qfiles differ from other CA OLQ functions in that they:

- Reflect the changing nature of data in the database
- Retain any formatting enhancements you have made to your report
- Re-execute any report calculations each time the qfile is executed

<table>
<thead>
<tr>
<th>What You Want to Do</th>
<th>Which CA OLQ Function to Use</th>
<th>Formatting Retained?</th>
<th>Calculations Reexecuted?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save the set of commands used to build a report</td>
<td>Qfile</td>
<td>Retained</td>
<td>Reexecuted each time the report is built</td>
</tr>
<tr>
<td>Save a copy of a report for later use</td>
<td>Saved report</td>
<td>Retained</td>
<td>Executed at report definition</td>
</tr>
<tr>
<td>Create a table from your report</td>
<td>Saved table</td>
<td>Not retained</td>
<td>Retained as they are executed when the table is saved</td>
</tr>
</tbody>
</table>

IDMSDB--How to Save a Set of Commands as a Qfile
For more information, see the following topics:
- Qfiles and Reports Key Terms (see page 542)
- Creating a Qfile (see page 542)
- Executing a Qfile (see page 548)
- Using One Qfile to Create Different Reports (see page 551)
- Modifying Your Qfile Definition (see page 555)
- Looking at Your Qfile Definition Syntax (see page 559)
- Modifying Your Qfile Definition Syntax (see page 563)
- Executing a Qfile in Batch Mode (see page 567)
- Deleting a Qfile (see page 569)

Qfiles and Reports Key Terms

Here are some terms used to discuss qfiles and reports:

**Current report**

The report you're working on in an active CA OLQ session. If you retrieve a saved report, CA OLQ clears out the current report.

**Data dictionary**

The storage facility used by CA products as a central source for data definitions, modules, and runtime information. Qfile definitions are stored in the data dictionary.

**Integrated Data Dictionary (IDD)**

The CA product used to access definitions stored in the dictionary.

**Qfile definition**

The CA OLQ syntax statements stored in the data dictionary when you create your qfile.

Creating a Qfile

Contents
- Step 1 - Build the salary report (see page 543)
- Step 2 - Add formatting enhancements (see page 545)
- Step 3 - Save the report as a Qfile (see page 547)

In this example, you create a qfile using the following steps:

1. Build a SALARY report containing employee salary information.
2. Add formatting enhancements you want kept in the qfile.
3. Save the report definition as a qfile.
Creating a Qfile:

1. Use OLQ to create the SALARY report.
2. Save these comments as the SALARY qfile.

Step 1 - Build the salary report

When you create a qfile, CA OLQ takes the set of commands you used to build your current report and saves them as the qfile definition. In this step, you build your current Salary report.

This report uses the EMPLOYEE and EMPOSITION records from the sample database. These records reside in the EMPSS01 subschema.

Start on the Signon Database View screen. To get there, type sub in the command line of any screen.

Select the EMPSS01 subschema.

```
CA OLQ Release nn.n
*** Signon Database View ***
Page 3 OF 3
121000 Select a subschema and press the ENTER key

Dictionary name . : TSTDICT
Database name . : 
Specify Subschema : of Schema . . : Version :

Select subschema: Description:
  EMPSS01 OF EMPSCHM VER 100 DEPARTMENT AND EMPLOYEE INFORMATION
  FINAN01 OF EMPSCHM VER 100 3Q91
  SALES01 OF EMPSCHM VER 100 SALES QUOTAS
```
Select the EMPLOYEE and EMPOSITION records.

CA OLQ Release nn.n
*** Record Select ***
->
123000 Select records and press the ENTER key

Enter records:
-and/or-
Select records:
- COVERAGE
- DENTAL-CLAIM
- DEPARTMENT
x EMPLOYEE
x EMPOSITION
- EXPERTISE
- HOSPITAL-CLAIM
- INSURANCE-PLAN
- JOB
- NON-HOSP-CLAIM
- OFFICE
- SKILL
- STRUCTURE

1=HELP 3=QUIT 4=MESSAGE 6=MENU PA2=REFRESH

Select EMP-NAME-0415, START-YEAR-0415, START-MONTH-0415, START-DAY-0415, SALARY GRADE-0420, SALARY-AMOUNT-0420, and BONUS-PERCENT-0420. Page forward (PF8) to view all of the columns you need.

CA OLQ Release nn.n
*** Column Select ***
->
124000 Select columns, specify selection criteria and press the ENTER key

Columns Currently Selected: 0
Selection Criteria Distinct N Y/N
EMPLOYEE
- 02 EMP-ID-0415 *
 0 EMP-NAME-0415 *
 03 EMP-FIRST-NAME-0415 *
 03 EMP-LAST-NAME-0415 *
 02 EMP-ADDRESS-0415
 03 EMP-STREET-0415
 03 EMP-CITY-0415
 03 EMP-STATE-0415
 03 EMP-ZIP-0415
 04 EMP-ZIP-FIRST-FIVE-0415
 04 EMP-ZIP-LAST-FOUR-0415
 02 EMP-PHONE-0415
 02 STATUS-0415

Enter additional criteria:

Proceed to Selection Criteria Screen? N Y/N

Select Yes to continue data retrieval.
Step 2 - Add formatting enhancements

Your qfile definition saves all of the formatting enhancements you have included in your report. Enhancements include sorts, break processing, external pictures, headings, and report titles.

See Section 5, "How to Format Your Report" for more information on how to format your report.

In this step, you enhance your qfile definition by:

- Sorting the report rows by salary grade
Grouping the report rows according to salary grade

Changing the report headers

Start on the Report Format - Sort screen. To get there, type sort on the command line of any screen.

To sort the rows by salary grade, specify 1 in the Sort Priority column next to SALARY-GRADE-0420. Specify a (for ascending) in the Order (A/D) column next to SALARY-GRADE-0420.

To group you report rows by salary grade, specify 1 in the Group By Level # column next to SALARY-GRADE-0420.

Display lines: Detail X and/or Summary X Group by all_

Compute:
1=HELP 3=QUIT 4=MESSAGE 5=DISPLAY 6=MENU 11=HEADER

Type header on the command line.
Specify column headers and press the ENTER key.

Underline character: -

Compute:

1=HELP 3=QUIT 4=MESSAGE 5=DISPLAY 6=MENU 8=FWD 10=SORT 11=EDIT

105022 Sort successfully completed. 68 records in. 68 records out.

125000 Press the ENTER key to go to the next page of the report.

Step 3 - Save the report as a Qfile

You now save your report as a qfile. CA OLQ takes all of the commands used to build the current report and saves them in a qfile definition in the data dictionary.

Start on the Qfile Processing screen. To get there, type qfile in the command line.

Select Create. Specify salary next to Routine name. Add any descriptive comments you would like to be shown on the screen.
CA IDMS - 19.0

Execute batch

Routine name: salary  Version: 
Comments: example

<table>
<thead>
<tr>
<th>SELECT ROUTINE</th>
<th>VERSION</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEPARTMENT</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>MANAGERS</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>NEW EMPLOYEES</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SALARY GRADE</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

1=HELP  3=QUIT  4=MESSAGE  6=MENU

CA OLQ returns a message indicating that your current report definition has been saved as a qfile. And CA OLQ lists the SALARY qfile.

CA OLQ Release nn.n

Requested operation for SALARY(1) has been successfully completed

Select function and press the ENTER key

User: SYB

Dictionary name: TSTDICT

Function: Execute  Create  Replace  Delete
          Execute with new criteria  List  Edit
          Execute batch

Routine name: 
Comments:

<table>
<thead>
<tr>
<th>SELECT ROUTINE</th>
<th>VERSION</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEPARTMENT</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>MANAGERS</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>NEW EMPLOYEES</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SALARY GRADE</td>
<td>1</td>
<td>SALARY SUMMARY COMPUTATIONS</td>
</tr>
<tr>
<td>SALARY</td>
<td>1</td>
<td>EXAMPLE</td>
</tr>
</tbody>
</table>

1=HELP  3=QUIT  4=MESSAGE  6=MENU

Executing a Qfile

In this example, you execute a qfile to create a report.

Executing a Qfile:
1. On the qfile processing screen, select the EXECUTE activity and the SALARY qfile.

2. OLQ retrieves the SALARY qfile definition from the data dictionary and executes it, creating the salary report.

Start on the Qfile Processing screen. To get there, type qfile in the top line of any screen.

Select Execute. Name the SALARY qfile.

<table>
<thead>
<tr>
<th>Function</th>
<th>Version</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEPARTMENT</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>MANAGERS</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>NEW EMPLOYEES</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SALARY GRADE</td>
<td>1</td>
<td>SALARY SUMMARY COMPUTATIONS</td>
</tr>
<tr>
<td>x SALARY</td>
<td>1</td>
<td>EXAMPLE</td>
</tr>
</tbody>
</table>

IDMSDB--Executing a Qfile
Specify **Yes** to continue data retrieval.

```
CA OLQ Release nn.n                     *** Retrieval Interrupted ***

092018 The default DICTNAME value has been modified.
092027 Underline character has been modified to -. 
Retrieval interrupted due to excessive data base accesses.
```

Number of whole rows: 54
Total number of records read: 100
Total number of records selected: 99
Number of data errors: 0

```
Continue execution: x Yes
 X No
```

Current interrupt interval is 100 data base accesses.

```
CA OLQ Release nn.n                     *** Retrieval Completed ***

105022 Sort successfully completed. 68 records in. 68 records out.
092021 Execution has completed for q-file: SALARY
```

```
Number of whole rows: 68
Total number of records read: 125
Total number of records selected: 124
Number of data errors: 0
```

Select
--->
```
Command/ Option       Display/Format Activity <--- Screen Name
X   Display report    DISplay
_   Save report       SAVE
_   Choose the sort sequence of report    SORt
_   Change column headers       HEAder
_   Change page header and footer      PAge HEAder
_   Change display format of data (,commas)     PICTure
_   Format columns (Alignment, sparse)            EDIT
_   Specify summary computations (Totals)             GROup BY
_   Send the report to a printer                 PRInt
```

```
CA OLQ Release nn.n                     *** Display Report ***

125000 Press the ENTER key to go to the next page of the report.
```

```
EMPLOYEE/EMPOSITION REPORT

   mm/dd/yy

   FIRST      LAST         START   GRADE  SAL.     BONUS
   NAME       NAME         AMOUNT  %

TOM        FITZHUGH      09 19 81  11 13000.00  .004
NANCY       TERNER       05 26 82  11 13000.00  .004

```
Using One Qfile to Create Different Reports

Contents

- Step 1 - Use execute with new criteria (see page 552)
- Step 2 - Change the Selection criteria (see page 553)
- Step 3 - Resume Qfile execution (see page 553)

In this example, you use the same qfile to create two or more reports that are nearly the same. Using **Execute with new criteria**, you can suspend qfile execution and change the retrieval selection criteria used to build the report.

Example

By executing the SALARY qfile and modifying the retrieval criteria, you can create two reports:

- One displays all salary grades above $30,000.
- One displays all salary grades below $30,000.

Using One Qfile to Make More Than One Report:
In this example, you execute the SALARY qfile using the following steps:

1. Execute the SALARY qfile using **Execute with new criteria**.

2. CA OLQ starts to execute the qfile. Execution continues up to the first statement that would retrieve data from the database.

3. CA OLQ suspends execution of the qfile and displays the Column Select screen.

4. You can change the retrieval criteria to display only those salary grades larger than 16.

5. When you press [Enter], CA OLQ resumes execution of the qfile.

**Step 1 - Use execute with new criteria**

In this step, you use the Execute with new criteria option to suspend execution of the SALARY qfile before it retrieves any data from the database.

Start on the Qfile Processing screen. To get there, type `qfile` in the command line of any screen.

Select **Execute with new criteria** and select the **SALARY** qfile.

```
CA OLQ Release nn.n                                      *** QFILE Processing ***
           ->                                   Page 1 of 1
140000 Select function and press the ENTER key

User: SYB
Dictionary name: TSTDICT
```

Dictionary node:
Step 2 - Change the Selection criteria

In this step, you change the data retrieval selection criteria to retrieve only those employees in salary grades greater than 16.

Specify greater than 16 in the Selection criteria column for the SALARY-GRADE-0420 field.

Step 3 - Resume Qfile execution

In this step, you resume execution of the qfile to apply new data retrieval selection criteria.

The final report displays only those employees in salary grades of 16 or higher.

Select Yes to continue data retrieval.
Number of whole rows . . . . . . . . . . 48
Total number of records read . . . . . . 100
Total number of records selected . . . . 87
Number of data errors . . . . . . . . . . 0

Continue execution  x Yes
X  No

Current interrupt interval is 100 data base accesses.

Select activity and press the ENTER key

1=HELP 3=QUIT 4=MESSAGE

Depending on how high the interrupt count at your site is set, CA OLQ may bypass the Retrieval Interrupted screen and proceed directly to the Retrieval Completed screen.

CA OLQ Release nn.n  *** Retrieval Completed ***
->
130000

EMPLOYEE/POSITION REPORT

FIRST NAME  LAST NAME  START  GRADE SAL. AMOUNT BONUS %

HERBERT  LIPSICH  04 29 81  21  18500.00 .004
MICHAEL  ANGELO  09 08 79  21  18000.00 .004
RALPH  TYRO  12 21 80  21  20000.00 .004
MICHAEL  ANGELO  09 08 79  21  17000.00 .004
DOUGLAS  KAHALLY  09 29 79  21  20000.00 .004
CAROL  MCDOUGALL  06 07 80  21  18000.00 .004

AVE FOR 21: 18583.33

JANE  FERNDALE  09 09 79  22  22500.00 .004

AVE FOR 22: 22500.00 .004

- 1 -
Modifying Your Qfile Definition

Contents

- Step 1 - Establish a current report (see page 556)
- Step 2 - Modify the report definition (see page 558)
- Step 3 - Replace the Qfile definition (see page 559)

When you modify a qfile, CA OLQ overlays your IDD qfile definition with the commands used to build your current report. Using this feature, you can:

- Execute an existing qfile, change the report, and replace the qfile
- Execute a qfile, change the report, and save it as a different qfile
- Create a new report from scratch and save it under the name of an existing qfile

In this example, you modify the SALARY qfile using the following steps:

1. Establish a current report by executing the SALARY qfile
2. Modify the report definition to suppress the display of detail lines (those not containing any summary calculations)
3. Replace the qfile, overriding the current SALARY qfile definition with the commands used to build the summary report

Modifying Your Qfile Definition:
Step 1 - Establish a current report

In this step, you execute the SALARY qfile to create a current report. Start on the Qfile Processing screen. To get there type `qfile` in the command line of any screen.

Select **Execute** and select the **SALARY** qfile.

```
CA OLQ Release nn.n
-> *** QFILE Processing ***
```

Replace the SALARY qfile with the commands used to build the modified report.
140000 Select function and press the ENTER key

User: SYB
Dictionary name: TSTDICT
Dictionary node:

Function:  x Execute  Create  Replace  Delete
> Execute with new criteria  List  Edit
> Execute batch

Routine name:  Version:
Comments:

SELECT ROUTINE  VERSION  COMMENTS
x DEPARTMENT  1
x MANAGERS  1
x NEW EMPLOYEES  1
x SALARY GRADE  1 SALARY SUMMARY COMPUTATIONS
x SALARY  1 EXAMPLE

1=HELP  3=QUIT  4=MESSAGE  6=MENU

Specify Yes to continue data retrieval.

> CA OLQ Release nn.n
> *** Retrieval Interrupted ***

192018 The default DICTNAME value has been modified.
092027 Underline character has been modified to -
  Retrieval interrupted due to excessive data base accesses.

Number of whole rows. . . . . . . . 54
Total number of records read. . . . . 100
Total number of records selected. . 99
Number of data errors . . . . . . . 0

Continue execution  x Yes
x No

Current interrupt interval is 100 data base accesses.

1=HELP  3=QUIT  4=MESSAGE

Specify that you want to proceed to the Report Format - Sort screen.

> CA OLQ Release nn.n
> *** Retrieval Completed ***

105022 Sort successfully completed. 68 records in. 68 records out.
092021 Execution has completed for q-file: SALARY

Number of whole rows. . . . . . . . 68
Total number of records read. . . . . 125
Total number of records selected. . 124
Number of data errors . . . . . . . 0

Select Option  ---> Display/Format Activity <--- Command/
Screen Name
x Display report DISplay
Save report SAVE
x Choose the sort sequence of report SORT
x Change column headers HEAder
x Change page header and footer PAGe HEAder
Step 2 - Modify the report definition

In this step, you modify the SALARY qfile to suppress the display of all detail lines (those not containing any summary calculations) in the report. Thus the report displays only the average salary for each salary grade.

Type a space next to **Detail** to suppress the display of detail lines.

Display lines:  **Detail** and/or  **Summary**

Compute:

Type **qfile** in the command line to proceed to the Qfile Processing screen.

Type  **qfile** in the command line to proceed to the Qfile Processing screen.
Step 3 - Replace the Qfile definition

In this step you override the SALARY command file with the commands used to make the current report.

Specify Replace. Select the SALARY qfile.

```
>> CA OLQ Release nn.n
   *** QFILE Processing ***
   Page 1 of 1

140000 Select function and press the ENTER key
User: SYB
Dictionary name: TSTDICT
Function:  _ Execute _ Create  x Replace  _ Delete
          _ Execute with new criteria  _ List  _ Edit
          _ Execute batch
Routine name:  
Comments: 

SELECT ROUTINE VERSION COMMENTS
  _ DEPARTMENT 1
  _ MANAGERS 1
  _ NEW EMPLOYEES 1
  _ SALARY GRADE 1 SALARY SUMMARY COMPUTATIONS
  x SALARY 1 EXAMPLE

1 HELP  3 QUIT  4 MESSAGE  6 MENU
```

CA OLQ responds with a message that the qfile has been replaced.

```
>> CA OLQ Release nn.n
   *** QFILE Processing ***
   Page 1 of 1

109017 Requested operation for SALARY(1) has been successfully completed
140000 Select function and press the ENTER key
User: SYB
Dictionary name: TSTDICT
Function:  _ Execute _ Create  _ Replace  _ Delete
          _ Execute with new criteria  _ List  _ Edit
          _ Execute batch
Routine name:  
Comments: 

SELECT ROUTINE VERSION COMMENTS
  _ DEPARTMENT 1
  _ MANAGERS 1
  _ NEW EMPLOYEES 1
  _ SALARY GRADE 1 SALARY SUMMARY COMPUTATIONS
  _ SALARY 1 EXAMPLE

1=HELP  3=QUIT  4=MESSAGE  6=MENU
```

Looking at Your Qfile Definition Syntax

Contents
- Step 1 - Switch to IDD (see page 562)
- Step 2 - View your Qfile in IDD (see page 562)
Where are qfile definitions stored?

When you create a qfile, CA OLQ takes all the commands used in your current session and saves them as CA OLQ syntax statements in the data dictionary. Definitions stored in the data dictionary are accessed using a CA product called the Integrated Data Dictionary (IDD).

How to look at your qfile definition

Using the List option on the Qfile Processing screen, you can switch from CA OLQ to IDD to see how your qfile commands are stored in the data dictionary.

For more information on IDD refer to the CA IDMS IDD DDDL Reference Section.

⚠️ Note: Depending on the security level that has been assigned to you, you may or may not be able to access your qfile definition in IDD. If your specifying List or Edit on the Qfile Processing screen does not switch you out of CA OLQ, you probably do not have authority to access IDD.

Looking at Qfile Syntax:
IDMSDB—Looking at Your Qfile Definition Syntax

In this example, you see how the SALARY qfile is stored in the data dictionary by using the following steps:

1. Switch to IDD using the List option on the Qfile Processing screen.
2. Look at the SALARY qfile definition in IDD.
3. Transfer from IDD back to CA OLQ.
Step 1 - Switch to IDD

Select List and the SALARY qfile.

```
CA OLQ Release nn.n                                      *** QFILE Processing ***
-> Requested operation for SALARY (1) has been successfully completed
140000 Select function and press the ENTER key
User: SYB
Dictionary name: TSTDICT
Function: _ Execute _ Create _ Replace _ Delete
          _ Execute with new criteria x List _ Edit
          _ Execute batch
Routine name: SELECT ROUTINE
Comments:

SELECT ROUTINE                VERSION   COMMENTS
  _ DEPARTMENT                1
  _ MANAGERS                  1
  _ NEW EMPLOYEES             1
  _ SALARY GRADE             1 SALARY SUMMARY COMPUTATIONS
  _ SALARY                   1 EXAMPLE

1=HELP  3=QUIT  4=MESSAGE  6=MENU
```

Step 2 - View your Qfile in IDD

In this step, you look at your qfile definition:

- To page forward through the qfile definition, press [PF8].
- To page backward, press [PF7].

Note that whenever CA OLQ retrieves data from the database, it creates a Structured Query Language (SQL) SELECT statement. The data retrieval selection criteria that you specified on the Column Select screen are specified in the WHERE clause of the SELECT statement.

For a more detailed explanation of this and other CA OLQ syntax statements, refer to the CA OLQ Reference Section.

Your qfile syntax includes a SELECT (OLQ access mode) statement. The WHERE clause specifies data retrieval selection criteria.

```
*+ SET DICTNAME TSTDICT
*+ SET UNDERLINE '-'
*+ SET ACCESS OLQ++ SIGNON SS EMPSS01 SCHEMA EMPSCHM ( 100)
*+ OPTIONS ALL HEADER ECHO NOPILLER FULL WHOLE INTERRUPT OLOHEADER
*+ NOPATHSTAT NOSTAT COMMENT VERBOSO NODBKEY PICTURE CODETAB NOSYN
*+ SELECT EMPLOYEE.EMP-FIRST-NAME-0415 EMPLOYEE.EMP-LAST-NAME-0415 EMPLOYEE.ST
*+ -YEAR-0415 EMPLOYEE.START-MONTH-0415 EMPLOYEE.START-DAY-0415 EMPOSITION.SAL
*+ -GRADE-0420 EMPOSITION.SALARY-AMOUNT-0420 EMPOSITION.BONUS-PERCENT-0420 FRD
*+ EMPLOYEE, EMPOSITION WHERE (EMP-EMPOSITION.EMPLOYEE.EMPOSITION)
```

K LINES AFTER 1:

```
*+ LINE 1 'EMPLOYEE/EMPOSITION REPORT' CENTER
*+ LINE 2 'PLACE' CENTER
*+ PAGE FOOTER BLANK LINES BEFORE 1
*+ LINE 1'- $PAGE '-' CENTER
```
Step 3 - Leave IDD

In this step, you switch from IDD back to CA OLQ.

Type `end` in the command line.

```
*+ SET DICTNAME TSTDICT
*+ SET UNDERLINE '-'
*+ SET ACCESS OLQ
*+ SIGNON SS EMPSS01 SCHEMA EMPSCHM (100)
*+ OPTIONS ALL HEADER ECHO NOFILLER FULL WHOLE INTERRUPT CA OLQHEADER
*+ SELECT EMPLOYEE.EMP-FIRST-NAME-0415 EMPLOYEE.EMP-LAST-NAME-0415 EMPLOYEE.ST
*+ -YEAR-0415 EMPLOYEE.START-MONTH-0415 EMPLOYEE.START-DAY-0415 EMPOSITION.SAL
*+ -GRADE-0420 EMPOSITION.SALARY-AMOUNT-0420 EMPOSITION.BONUS-PERCENT-0420 FRO
*+ EMPLOYEE.EMPOSITION WHERE (EMP-EMPOSITION.EMPLOYEE.EMPOSITION)
*+ PAGE HEADER BLANK LINES AFTER 1
*+ LINE 1 'EMPLOYEE/EMPOSITION REPORT' CENTER
*+ PAGE FOOTER BLANK LINES BEFORE 1
*+ LINE 1 '-' $PAGE '-' CENTER
*+ EDIT EMP-FIRST-NAME-0415
*+ ALIGN LEFT
*+ CA OLQHEADER 'FIRST'
*+ 'NAME'
*+ PICTURE 'X(10)'
*+ EDIT EMP-LAST-NAME-0415
*+ ALIGN LEFT
*+ CA OLQHEADER 'LAST'
```

Modifying Your Qfile Definition Syntax

**Contents**

- **Step 1 - Switch to IDD** (see page 564)
- **Step 2 - Modify the SALARY Qfile Definition** (see page 565)
- **Step 3 - Replace the SALARY Qfile definition** (see page 565)
- **Step 4 - Switch from IDD back to CA OLQ** (see page 566)

In this example, you edit the SALARY qfile definition in the data dictionary using the following steps:

1. Switch to IDD using the `Edit` field on the Qfile Processing screen.
2. Modify the SALARY qfile definition.
3. Replace the SALARY qfile definition.
4. Switch from IDD back to CA OLQ.
Step 1 - Switch to IDD

In this step, you suspend your current CA OLQ session and transfer to the Integrated Data Dictionary (IDD).

Select Edit and the SALARY qfile.

Select function and press the ENTER key

User: SYB
Dictionary name: TSTDICT

Function: _ Execute _ Create _ Replace _ Delete
  _ Execute with new criteria _ List _ x Edit

Routine name: .
Comments:

<table>
<thead>
<tr>
<th>SELECT ROUTINE</th>
<th>VERSION</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEPARTMENT</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>MANAGERS</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>NEW EMPLOYEES</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SALARY GRADE</td>
<td>1</td>
<td>SALARY SUMMARY COMPUTATIONS</td>
</tr>
<tr>
<td>x SALARY</td>
<td>1</td>
<td>EXAMPLE</td>
</tr>
</tbody>
</table>

1=HELP 3=QUIT 4=MESSAGE 6=MENU

Note: When you first see the qfile definition, erase the first three lines (those that sign you on to IDD, specify the dictionary name, and display the qfile). After you have erased these lines, your first line should begin with the verb MODIFY. If you don't do this, IDD will see DISPLAY as the first verb when you press [Enter], and will redisplay the qfile (without the changes you have made).

In this example, VERB MODIFY means that the qfile definition can be updated. Press [PF8] to scroll to the next page.
Step 2 - Modify the SALARY Qfile Definition

In this step, you change the SALARY qfile definition using the following steps:

1. Press [PF4] to supply space to type in your changes.

2. Add a restriction to the SELECT statement WHERE clause that limits retrieval to salary grades higher than 18.

3. Press [PF5] to apply the change.

For more information on how to change dictionary definitions in IDD, refer to the CA IDMS Dictionary Module Editor User Section.

In this example, your qfile syntax includes a SELECT (OLQ access mode) statement. Position the cursor on the line that contains the WHERE clause. Press [PF4] to insert additional criteria.

IDD nn.n ONLINE PAGE 2 LINE 1 DICT=TSTDICT 24/86
QFILE SOURCE FOLLOWS
SET DICTNAME TSTDICT
SET UNDERLINE '-'
SET ACCESS OLQ SIGNON SS EMPSS01 SCHEMA EMPSCHM ( 100)
OPTIONS ALL HEADER ECHO NOFILLER FULL WHOLE INTERRUPT OLOHEADER -
NOPATHSTAT NOSTAT COMMENT VERBOSE NODBKEY PICTURE CODETAB NOSYN
SELECT EMPLOYEE.EMP-FIRST-NAME-0415 EMPLOYEE.EMP-LAST-NAME-0415 EMPLOYEE.START-YEAR-0415 EMPLOYEE.START-MONTH-0415 EMPLOYEE.START-DAY-0415 EMPOSITION.SALARY-GRADE-0420 EMPOSITION.SALARY-AMOUNT-0420 EMPOSITION.BONUS-PERCENT-0420 FROM EMPLOYEE, EMPOSITION
WHERE (EMP-EMPOSITION.EMPLOYEE.EMPOSITION AND SALARY-GRADE-0420 GT 18)

Type a blank over the closing parenthesis on the WHERE clause. Add AND SALARY-GRADE-0420 GT 18) to the WHERE clause. Don't forget the closing parenthesis. Then press [PF5] to apply the change.

IDD nn.n ONLINE PAGE 2 LINE 11 DICT=TSTDICT 34/86
EMPLOYEE, EMPOSITION WHERE (EMP-EMPOSITION.EMPLOYEE.EMPOSITION AND SALARY-GRADE-0420 gt 18)

Step 3 - Replace the SALARY Qfile definition

In this step, you replace the qfile definition in the data dictionary.

Press [PF6] to replace the qfile definition in the data dictionary.

IDD nn.n ONLINE PAGE 2 LINE 11 DICT=TSTDICT 34/91
EMPLOYEE, EMPOSITION WHERE (EMP-EMPOSITION.EMPLOYEE.EMPOSITION AND SALARY-GRADE-0420 gt 18)
PAGE HEADER BLANK LINES AFTER 1 -
Step 4 - Switch from IDD back to CA OLQ

In this step, you terminate your IDD session and switch back to the Qfile Processing screen.

Type `end` in the command line to get back to CA OLQ.
Executing a Qfile in Batch Mode

Contents
- Step 1 - Select a Qfile (see page 568)
- Step 2 - Using the Batch Processing screen (see page 568)

When do you use batch?

If you have a qfile that you know is going to access a large amount of data, or if you want to execute your qfile overnight, the batch option is the most efficient operating method. Executing a job in batch mode frees computer resources for other users. Your DBA can help you decide when to execute your qfile in batch.

Because using batch requires advance preparation by your DBA, you will not actually execute a qfile in this example. This example shows you the CA OLQ tools you need to know about to execute a qfile in batch mode.

You will:

1. Select a qfile.

2. Transfer to the Batch Processing screen.
Step 1 - Select a Qfile

Select **Execute batch** and select the **SALARY** qfile.

```
140000 Select function and press the ENTER key

User: SYB
Dictionary name: TSTDICT

Function: Execute Create Replace Delete
   - Execute with new criteria
   - List
   - Edit
   X Execute batch
```

Routine name: Version:
Comments:

```
SELECT ROUTINE VERSION COMMENTS
   - DEPARTMENT 1
   - MANAGERS 1
   - NEW EMPLOYEES 1
   - SALARY GRADE 1 SALARY SUMMARY COMPUTATIONS
   X SALARY 1 EXAMPLE
```

1=HELP 3=QUIT 4=MESSAGE 6=MENU

The default values for executing a qfile in batch are displayed when you access this screen from the Qfile Processing screen.

```
132000 Select activity, output selection and press the ENTER key

Password:
Job control module: **OLQBATCH-JCL-SYB**
Select activity:
   - Submit current report request and comments
   X Submit OLQ syntax

To automatically generate output syntax:
   X Display (Output to SYSLST)
   - Print (DC printer) Class: Dest:
   - Save report Name:

Enter OLQ Syntax/Comments:
> **QFIL 'SALARY '(1) DICT TSTDICT DICTNO ' **
> 1=HELP 3=QUIT 4=MESSAGE 6=MENU
```

Step 2 - Using the Batch Processing screen

To execute a qfile in batch mode:

1. Enter your user password.

2. Make sure that the proper Job Control Module is displayed. Ask your DBA to make sure you have the right name.
3. Select an output destination, or accept the default destination that has been listed for you. In addition, the output of the batch job is always displayed at your terminal.

4. Check to see if the correct qfile name is listed in the Enter OLQ Syntax/Comments: field.

5. Press [Enter] to initiate your batch job.

For more information on how to use batch processing in CA OLQ, refer to Section12, "How to Use CA OLQ in Batch Mode".

Deleting a Qfile

When you delete a qfile, CA OLQ uses the following:

- Deletes the qfile definition in the data dictionary
- Removes the name of the definition from the list of qfiles on the Qfile screen

In this example, you delete a qfile using the Qfile Processing screen.

Start on the Qfile Processing screen. To get there, type qfile in the command line of any screen.

Select Delete and select the SALARY qfile.

CA OLQ responds with a message indicating that the qfile definition has been deleted. The name of the qfile is deleted from the list of routine names.
How to Make a Report from Database Records

In this section Using CA OLQ, you can report on records stored in a CA IDMS/DB.

Access mode: This section only applies when the access switch is set to OLQ.

For information on setting the access switch, see Step 1? Select the type of table.

For example, you can make an employee report based on data retrieved from the EMPLOYEE record.

Creating a Report:
Key Terms 5

Here are some of the terms used to discuss making a report from database records:

Database view

A more descriptive term for subschema. The two terms are used interchangeably.

Element
The smallest significant unit of data in a CA IDMS/DB database. Record elements correspond to columns in a table. For example, the record element DEPT-ID-0410 corresponds to the DEPARTMENT ID column in a table.

**Record**

A group of related elements. For example, the DEPT-NAME-0410, DEPT-ID-0410, and DEPT-HEAD-ID-0410 elements are all grouped into the DEPARTMENT record. Records correspond to rows in a table. For example, the record element DEPT-NAME-0410 corresponds to the DEPARTMENT NAME column in a table.

**Selection criteria**

A logical expression that you use to tell CA OLQ which rows of data to retrieve for your report.

**Additional selection criteria**

A logical expression that you use to tell CA OLQ which rows of data to retrieve for your report. Unlike selection criteria, you can also use:

- Logical record keywords
- Criteria expressions for subscripted fields

**Subschema**

A view of the database that contains a subset of the records, elements, sets, and areas that make up the entire database. A subschema usually views functionally related data.

For example, the personnel department uses a subschema that views employee information such as salary, date of hire, and personal information. All of the information used by the corporation (sales, accounting, and personnel) are held in the same database, but the personnel department views only the information that it needs.

**Using Subschemas:**
Create a Report

In this sample, you create a report using the following steps:

- **Step 1 -- Choose a subschema.** (see page 573)
- **Step 2 -- Select your records** (see page 574)
- **Step 3 -- Choose columns and specify selection criteria** (see page 574)
- **Step 4 -- Enter additional selection criteria** (see page 579)
- **Step 5 -- Retrieve the data and display your report** (see page 580)

**Step 1 -- Choose a subschema.**

In this step, you select a subschema that views employee information.

Select the EMPSS01 subschema.

```plaintext
CA OLQ Release nn.n
* * * Signon Database View * * *
121000 Select a subschema and press the ENTER key
```
Step 2 -- Select your records

In this step, you specify that you want to report on the EMPLOYEE and OFFICE records.

1=HELP 3=QUIT 4=MESSAGE 6=MENU 7=BWD 8=FWD PA2=REFRESH

Step 3 -- Choose columns and specify selection criteria

In this step, you:

1. Choose which columns you want to display in the report.

2. Specify retrieval selection criteria that restrict what data is retrieved from the database. In this sample, your selection criteria limit the report to those employees from Boston.
CA IDMS - 19.0

| 03 EMP-FIRST-NAME-0415 | *
| x 03 EMP-LAST-NAME-0415 | *
| 02 EMP-ADDRESS-0415 |
| 03 EMP-STREET-0415 |
| x 03 EMP-CITY-0415 | eq boston
| 03 EMP-STATE-0415 |
| 04 EMP-ZIP-FIRST-FIVE-0415 |
| 04 EMP-ZIP-LAST-FOUR-0415 |
| x 02 EMP-PHONE-0415 |
| 02 STATUS-0415 |

Additional Selection Criteria:

More about selection criteria

Selection criteria are logical expressions you use to tell CA OLQ which rows of data you want to include in your report. You don't have to specify any selection criteria. If you don't, CA OLQ retrieves all rows from the record.

If you specify... Your report looks like...

<p>| EMPLOYEE Report |
| mm/dd/yy |</p>
<table>
<thead>
<tr>
<th>Emp ID</th>
<th>Last Name</th>
<th>City</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMP-ID-0415 EQ '0075' -&gt; 0075</td>
<td>Lanzarotta</td>
<td>Lowell</td>
<td>MA</td>
</tr>
</tbody>
</table>

<p>| EMPLOYEE Report |
| mm/dd/yy |</p>
<table>
<thead>
<tr>
<th>Emp ID</th>
<th>Last Name</th>
<th>City</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMP-CITY-0415 EQ 'SHELBURNE FALLS' -&gt; 3302</td>
<td>Elopoulos</td>
<td>Shelburne Falls</td>
<td>MA</td>
</tr>
<tr>
<td>3871</td>
<td>Mahoney</td>
<td>Shelburne Falls</td>
<td>MA</td>
</tr>
<tr>
<td>4230</td>
<td>Ho</td>
<td>Shelburne Falls</td>
<td>MA</td>
</tr>
</tbody>
</table>

<p>| EMPLOYEE Report |
| mm/dd/yy |</p>
<table>
<thead>
<tr>
<th>Emp ID</th>
<th>Last Name</th>
<th>City</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMP-STATE-0415 NE 'MA' -&gt; 2789</td>
<td>Vangelis</td>
<td>Buckatunna</td>
<td>MS</td>
</tr>
<tr>
<td>5558</td>
<td>Runningbrook</td>
<td>Casper</td>
<td>WY</td>
</tr>
</tbody>
</table>

<p>| EMPLOYEE Report |
| mm/dd/yy |</p>
<table>
<thead>
<tr>
<th>Emp ID</th>
<th>Last Name</th>
<th>City</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMP-CITY-0415 MATCHES &quot;WAL@@@@&quot; -&gt; 1164</td>
<td>Rokoski</td>
<td>Waltham</td>
<td>MA</td>
</tr>
<tr>
<td>2112</td>
<td>Walesa</td>
<td>Walden</td>
<td>MA</td>
</tr>
<tr>
<td>3881</td>
<td>Li</td>
<td>Walpole</td>
<td>MA</td>
</tr>
<tr>
<td>4003</td>
<td>Barrett</td>
<td>Waltham</td>
<td>MA</td>
</tr>
</tbody>
</table>
How to specify selection criteria

Each column has its own Selection criteria field. If you want to retrieve rows based on the value in a certain column, fill in the Selection criteria field next to that column. For example:

<table>
<thead>
<tr>
<th>Selection criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>X 02 COLUMN1</td>
</tr>
<tr>
<td>02 COLUMN2</td>
</tr>
<tr>
<td>X 02 COLUMN3</td>
</tr>
<tr>
<td>= 2000</td>
</tr>
<tr>
<td>X 02 COLUMN4</td>
</tr>
<tr>
<td>&gt; 5 * COLUMN1</td>
</tr>
</tbody>
</table>

The following table gives examples of the types of operators you can include in your selection criteria.

<table>
<thead>
<tr>
<th>Type of Operator</th>
<th>Operator Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical comparison operators</td>
<td>=  Equal to</td>
<td>= 100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 100 to 500 (exclusive)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>= 100 thru 500 (inclusive)</td>
</tr>
<tr>
<td></td>
<td>-= Not equal to</td>
<td>-= COLUMN2</td>
</tr>
<tr>
<td></td>
<td>&gt;  Greater than</td>
<td>&gt; COLUMN1 * COLUMN2</td>
</tr>
<tr>
<td></td>
<td>&gt;= Greater than or equal to</td>
<td>&gt;= COLUMN4 + 100</td>
</tr>
<tr>
<td></td>
<td>&lt;  Less than</td>
<td>&lt; COLUMN4 - COLUMN1</td>
</tr>
<tr>
<td></td>
<td>&lt;= Less than or equal to</td>
<td>&lt;= 750 / COLUMN4</td>
</tr>
<tr>
<td>Arithmetic operators</td>
<td>+  Addition</td>
<td>COLUMN1 + COLUMN2</td>
</tr>
<tr>
<td></td>
<td>-  Subtraction</td>
<td>COLUMN1 - 5</td>
</tr>
<tr>
<td></td>
<td>*  Multiplication</td>
<td>COLUMN1 * 1/2</td>
</tr>
<tr>
<td></td>
<td>/  Division</td>
<td>COLUMN1/COLUMN2</td>
</tr>
<tr>
<td>MATCHES and CONTAINS characters</td>
<td>* Stands for any character</td>
<td>contains &quot;Z**05&quot;</td>
</tr>
<tr>
<td></td>
<td>@  Stands for any alphabetic character</td>
<td>matches &quot;NEW@@@@@@@@@@&quot;</td>
</tr>
<tr>
<td></td>
<td>#  Stands for any numeric character</td>
<td>matches &quot;669####&quot;</td>
</tr>
</tbody>
</table>

The IN clause

You can include the IN clause in your selection criteria. The IN clause is a subset of the SQL SELECT statement that compares your column value to a list of data values.

For complete syntax and syntax rules, refer to the CA OLQ Reference Section.

IN IN yields a true comparison if the column value matches one or more of the data values. For example, you can compare the value of the EMP-LAST-NAME field to the following IN clause:

```
IN ('JONES', 'TANAKA', 'ANDERSON')
```

NOT IN NOT IN yields a true comparison if the column value does not match or more of the data values. For example, you can compare the value of the EMP-LAST-NAME field to the following IN clause:

```
NOT IN NOT IN("JONES", "TANAKA", "ANDERSON")
```
The LIKE clause

You can also include the LIKE clause in your selection criteria. The LIKE clause is a subset of the SQL SELECT statement that searches for a pattern string.

For complete syntax and syntax rules, refer to the CA OLQ Reference Section.

How to use the LIKE clause

Your LIKE clause contains two parts:

- The LIKE/NOT LIKE keywords
- A pattern string

LIKE

LIKE determines whether a column expression contains a pattern string. LIKE yields a true comparison if the column contains the pattern. For example, you can compare the EMP-LAST-NAME field to the following LIKE clause:

LIKE 'MAC%'

This comparison is true if the last name is any number of characters long and begins with the string MAC.

NOT LIKE

NOT LIKE yields a true comparison if the column does not contain the pattern. For example, you can compare the OFFICE-CODE field to the following NOT LIKE clause:

NOT LIKE '002'

This comparison is true if the office code is anything except for 002.

Pattern string

The pattern string is what is compared to the column value. The pattern can contain:

- Alphanumeric characters for an exact match
- Special characters used as wild cards
- An escape character to exactly match the special characters

The LIKE clause is summarized in the table below.

<table>
<thead>
<tr>
<th>Object String</th>
<th>Pattern String</th>
<th>Example of Syntax</th>
<th>Example of True comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>Underscore(_)</td>
<td>Any single character</td>
<td>NAME LIKE 'S___'</td>
<td>True if NAME is exactly 3 characters long and the first character is S</td>
</tr>
</tbody>
</table>
Object String | Pattern String | Example of Syntax | Example of True comparison
---|---|---|---
Percent sign (%) | Any sequence of zero or more characters | NAME LIKE '%C__' | True if NAME is 3 or more characters long and the third from last character is C

Single alphanumeric character | Exact match to that alphanumeric character | NAME LIKE 'MAC' | True if NAME is MAC

Escape character + underscore (_) | Exact match to the underscore (_) | PARTNUM LIKE '*_115' ESCAPE '*' | True if PARTNUM is __115

Escape character + percent sign (%) | Exact match to the percent sign (%) | PARTNUM LIKE '**%15' ESCAPE '**' | True if PARTNUM is *%15

Escape character alone | Exact match to the escape character | PARTNUM LIKE '***' | True if PARTNUM is ***

1 The escape character can be any single alphanumeric character and is set by specifying ESCAPE 'escape-character' in your SELECT statement.

The AND operator

If you enter selection criteria for more than one column, CA OLQ combines the expressions by placing the AND operator between them. AND means that a row must meet both conditions in order to be retrieved.

For example, when you specify:

```
X 02 COLUMN1
 02 COLUMN2
X 02 COLUMN3  eq 2000
X 02 COLUMN4  gt 5 * COLUMN1
```

CA OLQ produces:

COLUMN3 eq 2000 and COLUMN4 gt 5 * COLUMN1

The OR operator

You may not want to combine your expressions with AND operators. As an alternative, you can type the OR operator at the beginning of any expression. OR means that a row need only meet one of the conditions in order to be retrieved.

For example, when you specify:

```
X 02 COLUMN1
 02 COLUMN2
X 02 COLUMN3  eq 2000
X 02 COLUMN4  or gt 5 * COLUMN1
```

CA OLQ produces:
COLUMN3 eq 2000 or COLUMN4 gt 5 * COLUMN1

Selection criteria for unchosen columns

You can specify criteria for columns that do not appear on your report. CA OLQ tests the values in such columns when it chooses which rows to retrieve, but does not display those values on the report.

Built-in functions

CA OLQ provides built-in functions that you can use in your selection criteria. Built-in functions are a shorthand way of performing common calculations (such as square root) and string manipulations (such as concatenation).

For more information on CA OLQ built-in functions, refer to the CA OLQ Reference Section.

Step 4 -- Enter additional selection criteria

In this step, you:

Specify Y to proceed to the Additional selection criteria screen.

CA OLQ Release nn.n *** Column Select ***

->

124000 Select columns, specify selection criteria and press the ENTER key

Columns Currently Selected: 0

<table>
<thead>
<tr>
<th>Columns Selected</th>
<th>Selection Criteria</th>
<th>Distinct N Y/N</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMP-ID-0415</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>EMP-NAME-0415</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMP-FIRST-NAME-0415</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMP-LAST-NAME-0415</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMP-ADDRESS-0415</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMP-STREET-0415</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMP-CITY-0415</td>
<td>eq boston</td>
<td></td>
</tr>
<tr>
<td>EMP-STATE-0415</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMP-ZIP-0415</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMP-ZIP-FIRST-FIVE-0415</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMP-ZIP-LAST-FOUR-0415</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMP-PHONE-0415</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STATUS-0415</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Additional Selection Criteria:

Proceed to Selection Criteria Screen? Y Y/N

1=HELP  3=QUIT  4=MESSAGE  6=MENU  8=FWD  PA2=REFRESH

At the Selection criteria screen, specify additional retrieval selection criteria that restrict what data is retrieved from the database. In this sample, your additional selection criteria limit the employees whose:

- Area code (first 3 digits) is '508'
- City of residence is not one beginning with 'New'
- Zip code begins with a '01'

146000 Type in selection criteria, and press the ENTER key.
Please Enter Additional Selection Criteria:
(EMP-PHONE-0415 MATCHES '508#######') AND (EMP-CITY-0415 NOT LIKE 'NEW%' OR EMP-CITY-0415 NOT LIKE 'BOSTON') AND (EMP-ZIP-FIRST-FIVE LIKE '01###')

1=HELP 3=QUIT 4=MESSAGE 6=MENU PA2=REFRESH

More about additional selection criteria

The Additional selection criteria field works like the Selection criteria field, enabling you to enter simple and compound expressions. In addition, you can enter:

- Logical record keywords
- Criteria expressions for subscripted fields

<table>
<thead>
<tr>
<th>Type of Expression</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Expressions</td>
<td>COLUMN1 eq 2000 thru 5000</td>
</tr>
<tr>
<td></td>
<td>not COLUMN1 eq 2000 thru 5000</td>
</tr>
<tr>
<td>Compound Expressions</td>
<td>(COLUMN3 eq 2000) and (COLUMN4 gt 5 * COLUMN1)</td>
</tr>
<tr>
<td></td>
<td>(COLUMN3 eq 2000) or (COLUMN4 gt 5 * COLUMN1)</td>
</tr>
<tr>
<td>Logical Record Keywords</td>
<td>ACCOUNTANT and EMPLOYEE-ON-LEAVE</td>
</tr>
<tr>
<td>Subscripted Fields</td>
<td>COLUMN5(3) matches &quot;329****&quot;</td>
</tr>
<tr>
<td></td>
<td>(COLUMN1 eq (2 * COLUMN6(1))) or (COLUMN7 ne 1)</td>
</tr>
</tbody>
</table>

Step 5 -- Retrieve the data and display your report

In this step, you retrieve your report data from the database and display your report.

CA OLQ Release nn.n
*** Retrieval Completed ***
-> 130000 Select activity and press the ENTER key

Number of whole rows . . . . . . . 2 Total displayable cols . 20
Total number of records read . . . 57 Formatted line length . 372
Total number of records selected . 2
Number of data errors . . . . . . . 0
How to Save a Table From a Report

**In this section** This section shows you how to save your current report as a data table. In this example, you will:

1. Create an EMPLOYEE/OFFICE report, retrieving data from the EMPLOYEE and OFFICE records
2. Save your current report as an ASF table
3. Save your current report as an SQL table
4. View the tables you just created

In this sample, you retrieve data from database records and store it in a table format. Each column of your table corresponds to an element in one of the records you retrieved. For example, the EMP-ID-0415 column corresponds to the EMP-ID-0415 element in the EMPLOYEE record.
Saving a Table from Your Current Report:

IDMSDB--How to Save a Table From a Report

Each row in the table you create corresponds to a single occurrence of a record. For example, the table row that contains data for employee ANGELO corresponds to one occurrence each of the EMPLOYEE and OFFICE records.

**Record Type:** EMPLOYEE

<table>
<thead>
<tr>
<th>EMP_ID</th>
<th>EMP-FIRST-NAME</th>
<th>EMP-LAST-NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>0145</td>
<td>NAME-0145</td>
<td>NAME-0145</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EMP_ID</th>
<th>EMP-LAST-NAME</th>
<th>EMP-FIRST-NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>0075</td>
<td>JENNIFER LANZAROTTA</td>
<td>0075 Lanzarotta Jennifer</td>
</tr>
</tbody>
</table>
Save a Table Key Terms

Some terms used to discuss saving a table from a report are:

**Automatic System Facility (ASF)**

A tool in CA IDMS/DB used to create and manage tables. You can use ASF to modify the table definition, once you have created a table using CA OLQ.

**ASF Dictionary**

An application dictionary used by ASF. You must be using the ASF dictionary when you are creating data tables with CA OLQ. The application dictionary contains schema definitions, subschema definitions, user information, and application definitions (such as records, maps, dialogs, and programs).

**Current report**

The report you’re working on in an active CA OLQ session. If you retrieve a saved report, CA OLQ clears out the current report.

**Table**

A presentation of data as a series of rows and columns.

- **ASF tables** refers to tables associated with the IDMSR schema.
- **SQL tables** refers to tables associated with an SQL schema.

Creating a Report 3

Contents

- Step 1 -- Select a subschema (see page 584)
- Step 2 -- Select what records you want (see page 584)
- Step 3 -- Select your columns (see page 585)
- Step 4 -- Retrieve your data (see page 585)
- Step 5 -- Display your report (see page 585)
In this step, you create an OFFICE/EMPLOYEE report. This report retrieves information from the EMPLOYEE and OFFICE records.

Step 1 -- Select a subschema

The EMPLOYEE and OFFICE records are defined in the EMPSS01 subschema for the sample employee database.

Start on the Signon Database View screen. To get there, type sub (for subschema) in the command line of any screen.

Select the EMPSS01 subschema.

Step 2 -- Select what records you want

Select the EMPLOYEE and OFFICE records.
Step 3 -- Select your columns


Step 4 -- Retrieve your data

Press [Enter] to view your report.

Step 5 -- Display your report

Your report lists data from the EMPLOYEE and OFFICE records.
#### Saving Your Report As an SQL Table

**Note:** Before beginning this step, make sure that the access switch has been set to `idms`.

For information on setting the access mode switch, see Step 1? Select the type of table.

In this step, you save your report as a SQL table. Type `table` in the command line to proceed to the Table Processing screen.

Select **Create**. Name your table `emp_office`.

```
Dictname: EMPLOYEE  Schema: EMPLOYEE
Function:   Select  x Create  Delete
      Add
Enter table:  emp_office
-or-
Select table
  - EMPLOYEE.ACCOUNTING
  - EMPLOYEE.BUDGET
  - EMPLOYEE.DEPARTMENT
```

```
1=HELP  3=QUIT  4=MESSAGE  6=MENU  8=FWD
```
CA OLQ responds with a message that your table has been defined. The EMP-OFFICE table is listed.

```
CA OLQ Release nn.n         *** Data Table Processing ***
  Page 2 of 2
102020 The appending, replacing, or creating of an SQL table has been successful.
102021 A total of 100 rows have been inserted into the named SQL table.
```

```
Dictname: EMPLOYEE
Function: X Select _ Create _ Delete
            _ Add _ Replace

Enter table:
-or-
Select table
  _ EMPLOYEE.ACCOUNTING
  _ EMPLOYEE.BUDGET
  _ EMPLOYEE.DEPARTMENT
  _ EMPLOYEE.EMP_OFFICE
```

```
1=HELP 3=QUIT 4=MESSAGE 6=MENU 8=FWD
```

### Saving Your Report As an ASF Table

**Note:** Before beginning this step, make sure that the access switch has been set to `olq`.

For information on setting the access mode switch, see Step 1? Select the type of table.

Before you begin, make sure that you are signed on to the correct **ASF dictionary**. Ask your DBA or support staff to make sure you are signed on to the right dictionary.

In this step, you save your report as an ASF table. Type `table` in the command line to proceed to the Table Processing screen.

Select **Create**. Name your table `emp-office`.

```
CA OLQ Release nn.n         *** Table Processing ***
  Page 1 of 2
138000 Select function, table(s) and press the ENTER key
```

```
Owner: DOC1
Catalog: ASFDICT Location:
Function: X Select _ Create Delete
            _ Add _ Replace

Enter table: emp-office
-or-
Select table
  _ ACCOUNTING
  _ BUDGET
  _ DEPARTMENT
```

16-Jan-2018 587/898
CA OLQ responds with a message that your table has been defined. The EMP-OFFICE table is listed.

CA OLQ release nn.n

->

Table processing has been successfully completed

138000 Select function, table(s) and press the ENTER key

1102017

Owner: DOC1

Catalog: ASFDICT Location:

Function: X Select _ Create _ Delete

_Add _ Replace

Enter table:
-or-

Select table

_ACCOUNTING

_BUDGET

_DEPARTMENT

_EMP-OFFICE

Viewing Your SQL Table

Note: Before beginning this step, make sure that the access switch has been set to idms.

For information on setting the access mode switch, see Step 1? Select the type of table.

In this step, you look at the table you just created.

Choose Select and select the EMP_OFFICE table.

CA OLQ release nn.n

->

138000 Select function, table(s) and press the ENTER key

Dictname: Schema: EMPLOYEE

Function: X Select _ Create _ Delete

_Add _ Replace

Enter table:
-or-

Select table

_EMPLOYEE.ACCOUNTING

_EMPLOYEE.BUDGET

_EMPLOYEE.DEPARTMENT

_EMPLOYEE.EMP_OFFICE
Viewing Your ASF Table

Note: Before beginning this step, make sure that the access switch has been set to `olq`.

For information on setting the access mode switch, see Step 1? Select the type of table.

In this step, you look at the table you just created.

Choose Select and select the EMP-OFFICE table.
How to Use CA OLQ in Batch Mode

What is batch processing? Batch processing is a way to create a report without requiring any interaction from the user. Once you have set up the requirements of the batch job, you can start it running and it will run until it has completed, without needing any input from you.

When do you use batch? Batch processing is used to process large amounts of data, or to perform an operation in off hours.

If you are creating a report that contains a very large number of records, the amount of time that the system spends retrieving your report’s data restricts with the system’s availability to other users. Thus, if you are creating a very large report, you probably want to perform the data retrieval at off hours, when the system’s resources aren’t at high demand.

Example

Suppose you were put in charge of creating a personnel report for the phone company. Your report retrieves 30,000 records, one record for each employee. Since you are creating this report for the first time, you will probably want to change it once you see how it looks.
Because your report contains so many records, you don't want to retrieve all of them until you are sure that you have the report exactly the way you want it. You can create a report prototype to make sure that you are satisfied with the report before you retrieve the data.

Using batch processing

The step-by-step solution for the personnel report would be to:

1. Set your interrupt count to a very low level (for example, 20 records)
2. Create the report, but limit it to 20 records
3. Save the report definition as a qfile
4. Execute the qfile in off hours using batch processing

Processing a Qfile in Batch:
Set the interrupt count to 20 records

Create a current report

Save it as a qfile

Execute the qfile using batch processing
OLQ Batch Mode Key Terms

Some terms used to discuss batch processing are:

**Batch**

Batch processing means that the user doesn’t have to interact with the computer system in order to perform a function. Usually, a batch job is set up in advance (such as when you fill out your Batch Processing screen). Once the job has started running, you cannot intervene except to cancel execution.

**Input file**

A file that contains input for a batch job.

**Interactive**

A way of performing a function in which the computer system requires the user to provide input, and then responds to that input. An example would be CA OLQ menu mode. Another term to describe interactive processing is **online**.

**Interrupt count**

The maximum number of records CA OLQ will retrieve when building a report. If the number of records that meet the selection criteria for that report exceeds the interrupt count, CA OLQ suspends data retrieval and issues a message asking you if you want to continue to retrieve records.

**Job Control Language**

A language used to define the special requirements of your batch program to the system. Job Control Language (JCL) statements name input and output files, the name of your program, and your output destination.

**Output destination**

Any type of device that receives the report that you have created through your batch job. Output destinations can be a printer, a terminal, or a log.

**Output file**

A file that contains the results of your batch job.

---

**Batch Process to Create a Report**

To create a report using batch, you do the following:

**Steps to Create a Report using Batch:**

<table>
<thead>
<tr>
<th>Step</th>
<th>What To Do</th>
<th>How To Do It</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Step What To Do | How To Do It
---|---
Specify which report you want to execute batch. | Build it from a table or records.  
Execute a qfile.  
Retrieve a saved report.  
Specify your processing commands on the Batch Processing screen.  

Move to the Batch Processing screen. | Type BATCH in the command line.  

Specify your batch processing options. | See Table 12-2  

Initiate batch processing. | Press [Enter].  

### The Batch Processing screen

Password: Job control module: OLQBATCH-JCL-RMG  
Select activity:  
Submit current report request and comments  
X Submit CA OLQ syntax  
To automatically generate output syntax:  
X Display (Output to SYSLST)  
Print (DC printer)  
_ Save report  
Name :  

Enter CA OLQ Syntax/Comments:  

> QFIL ‘SALARY ’(1) DICT TSTDICT DICTNO ’  

1=HELP 3=QUIT 4=MESSAGE 6=MENU  

You can accept the default values on the Batch Processing screen or type over them. You’ll probably need to ask your DBA how to fill out these fields.  

When you press [Enter], your batch job begins processing.  

Batch Processing screen fields are explained in the following.  

### Batch Processing Screen Fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password:</td>
<td>Provides security for saved reports and qfile definitions. Usually, you want to enter your user password (it will not show on the screen).</td>
</tr>
<tr>
<td>Job control module:</td>
<td>Names a module that contains Job control language (JCL) statements that tell the system how to run your batch job. Your DBA defines Job Control Language modules for your site.</td>
</tr>
</tbody>
</table>
| Select activity: | Tells CA OLQ what is to be processed in batch mode. You can:  
Execute the current report in batch mode (Submit current report request). If you access the Batch Processing screen from the Saved Reports screen, this field is the default. |
How to Print Your Report

Some terms used to discuss printing reports are:

Current report

The report you are working on in an active CA OLQ session. If you retrieve a saved report or execute a qfile, CA OLQ clears out the current report.

Destination

When you print a report, you specify an output destination where the report is to be printed. Usually, the destination is a file associated with a printer.

For more information, see the following topics:
- Printing a Report (see page 595)

Printing a Report

To print a report, do the following:

Steps for Printing a Report:

<table>
<thead>
<tr>
<th>Step</th>
<th>What To Do</th>
<th>How To Do It</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Create a current report.</td>
<td>Build it from a table or records.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Execute a qfile.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Retrieve a saved report.</td>
</tr>
<tr>
<td>2</td>
<td>Move to the Print Processing screen.</td>
<td>Type PRINT in the command line.</td>
</tr>
<tr>
<td>3</td>
<td>Specify your print options.</td>
<td>See Table 13-2</td>
</tr>
<tr>
<td>4</td>
<td>Initiate print processing.</td>
<td>Press [Enter].</td>
</tr>
</tbody>
</table>
When you get to the Print Processing screen, the fields are filled out with default values. You can accept these default values or type over them. When you press [Enter], your job is routed to a printer.

Print Processing screen fields are explained in table in the section "Batch Processing".

Print Processing Screen Fields:

<table>
<thead>
<tr>
<th>What to Specify</th>
<th>Which Field to Use</th>
<th>Restrictions</th>
<th>Defaults</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whether you want to print the report</td>
<td>Print current report:</td>
<td>Specify Yes or No</td>
<td>No</td>
</tr>
<tr>
<td>How many characters per print line</td>
<td>Line size:</td>
<td>1 through 9999</td>
<td>80</td>
</tr>
<tr>
<td>The number of lines per page</td>
<td>Line count:</td>
<td>1 through 9999</td>
<td>60</td>
</tr>
<tr>
<td>How many copies of the report you want to print</td>
<td>Number of copies:</td>
<td>1 to 256</td>
<td>1</td>
</tr>
<tr>
<td>The number of the first page you want to print</td>
<td>From page:</td>
<td>Any page number in the report</td>
<td>Page 1</td>
</tr>
<tr>
<td>How many report pages you want to print</td>
<td>for:</td>
<td>The number of pages All of the report</td>
<td></td>
</tr>
<tr>
<td>The printer used to print the report</td>
<td>Enter destination:</td>
<td>A printer destination defined by your DBA Site specific. Ask your support staff which printer is set up as the default printer.</td>
<td></td>
</tr>
</tbody>
</table>
Introducing the OLQ SELECT statement

What this section presents This section introduces the OLQ SELECT statement, which you can use to retrieve information in a CA IDMS/DB database. It also contains some tips for using the command mode and reporting features of CA OLQ.

When to use OLQ SELECT

Use the OLQ SELECT statement, to access:

- ASF tables
- Database records
- Logical records
- Sequential files (batch only)

Important! The examples of the OLQ SELECT statement in this section are valid when the access switch is set to `olq`.

For more information, see the following topics:

- The OLQ SELECT Statement (see page 597)
- Defining Your Data (see page 599)
- Some Tips On Using CA OLQ (see page 599)

The CA IDMS/DB SQL SELECT statement is used when the access switch is set to `sql`. For more information, see the CA IDMS SQL Reference Section.

The OLQ SELECT Statement

When to use the SELECT statement

CA OLQ uses the OLQ SELECT statement to access information from ASF tables, logical records, and database records.

Generate queries

By using the SELECT statement, you can formulate both simple and complex queries for information. CA OLQ interprets the SELECT statement and produces a report.

SELECT statement syntax
Depending on your request, use some or all of these clauses in the sequence listed:

<table>
<thead>
<tr>
<th>Clause</th>
<th>Information you supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT</td>
<td>Columns or fields to display.</td>
</tr>
<tr>
<td>FROM</td>
<td>The source of information (that is, table or record names).</td>
</tr>
<tr>
<td>WHERE</td>
<td>Selection criteria applied to rows or record occurrences.</td>
</tr>
<tr>
<td>GROUP</td>
<td>Column or field names for grouping information with the same value (for example, all rows with the same department ID).</td>
</tr>
<tr>
<td>HAVING</td>
<td>Selection criteria applied to grouped information (for example, a summary row where the average salary is less than $35,000).</td>
</tr>
<tr>
<td>UNION</td>
<td>Multiple SELECT statements. UNION combines the rows retrieved from each SELECT statement.</td>
</tr>
<tr>
<td>ORDER</td>
<td>Column or field names on which to sort retrieved information.</td>
</tr>
<tr>
<td>BY</td>
<td></td>
</tr>
</tbody>
</table>

**Reading a SELECT statement**

Read a SELECT statement like an English sentence. The order of the clauses falls into place. For example, to get all employee IDs less than 100 displayed in ascending order, you would:

- **SELECT** the employee ID
- **FROM** the EMP table
- **WHERE** the employee ID is less than 100
- **ORDER** the display **BY** employee ID

Here is the complete SELECT statement:

```
select empid from emp where empid < 100 order by empid
```

**Some examples**

These examples show how the SELECT statement is used to retrieve information from a database:

- List all information on corporate departments:
  ```
  select * from dept
  ```

- List the names and phone numbers of employees in department 4000 sorted by the employee's last name:
  ```
  select lastname, firstname, phone from emp where deptid = 4000 order by lastname
  ```

- Display the date Themis Papazeus was hired:
  ```
  select startdate from emp where firstname=&xq.themis' and lastname=&xq.papazeus'
  ```

- List the number of employees in department 4000:
  ```
  select count(*) from emp where deptid = 4000
  ```
How many employees in each department were hired before January 1, 1975?

```
select deptid, count(*) from emp where startdate < 750101 group by deptid
```

### Defining Your Data

#### Creating sample tables

If you want to use the SELECT statement as you proceed through this section, you need two or more ASF-generated tables.

To create an ASF-generated table, see the *CA IDMS ASF User Section*.

#### Sample tables used in this section

The sample tables used in this section are ASF-generated tables created in part from data stored in the Employee database your site receives at installation. The sample tables, which appear in Appendix A, "Sample Tables and Database", are:

<table>
<thead>
<tr>
<th>The table</th>
<th>Contains information on</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMP</td>
<td>Employees in departments 3100, 4000, and 6666</td>
</tr>
<tr>
<td>DEPT</td>
<td>All corporate departments</td>
</tr>
<tr>
<td>JOBLIST</td>
<td>Position held by each corporate employee</td>
</tr>
<tr>
<td>JOBCLASS</td>
<td>Class and associated salary range assigned to a position</td>
</tr>
<tr>
<td>BOSTON</td>
<td>Employees in the Boston office</td>
</tr>
<tr>
<td>WESTON</td>
<td>Employees in the Weston office</td>
</tr>
<tr>
<td>SPRINGFIELD</td>
<td>Employees in the Springfield office</td>
</tr>
</tbody>
</table>

#### Sample database

For the most part, examples in this section are based on the sample tables listed above. Examples that retrieve information from network records use the Employee database. The data structure diagram for the Employee database appears in Appendix A.

### Some Tips On Using CA OLQ

#### Contents

- Signing on (see page 600)
- Signing off (see page 601)
- Entering commands (see page 601)
- Setting the access to IDMS (see page 603)
- Tailoring reports (see page 603)

This section gives you some tips on using CA OLQ:

- Signing on
Signing off

- Entering statements
- Setting the access to IDMS
- Tailoring your reports

Signing on

To signon to CA OLQ follow these steps:

**Step 1 -- Sign on to a CA IDMS/DC or CA IDMS UCF system**

Access your CA IDMS/DC or CA IDMS UCF system and enter your signon id and password at the ENTER NEXT TASK CODE prompt:

```
V85  REL nn.n ENTER NEXT TASK CODE:  
    signon cub scout
```

**Step 2 -- Specify a dictionary**

Specify the data dictionary that contains the table or subschema definitions describing the data you want. You may be assigned to a dictionary when you sign on depending on the contents of your signon profile. You can change your signon dictionary or specify a dictionary by issuing a DCUF command to set the dictionary name:

```
IDMS DC258003 V85 USER CUB SIGNED ON LTERM VL85008 AT 11:57:35.75 88.040
V85  REL nn.n ENTER NEXT TASK CODE:  
    dcuf set dictname asfdict
```

**Note:** You can also specify a dictionary in the SIGNON statement.

**Step 3 -- Enter CA OLQ's task code**

Issue your site's task code for CA OLQ at the ENTER NEXT TASK CODE command. The task code provided at installation is OLQ:

```
SET DICTNAME ASFDICT
IDMS DC402009 V85 DICTNAME ASFDICT HAS BEEN SET
V85  REL nn.n ENTER NEXT TASK CODE:  
    olq
```

You are now signed on to CA OLQ.

**Step 4 -- Set your default access to OLQ**

The examples of the SELECT statement in this section are only valid when your default access is set to OLQ. You can do this for your CA OLQ session by specifying:

```
set access olq092033 00 Processing mode changed to OLQ.
```
Signing off

To end your DC/UCF session, follow these steps:

Step 1 -- Sign off CA OLQ

Sign off CA OLQ by entering BYE on the command line:

```
bye
```

```
OLQ 091057 00 Please enter next command.
```

CA OLQ terminates the CA OLQ session and returns you to DC/UCF.

Step 2 -- End your session

End your session by entering BYE at the ENTER NEXT TASK CODE prompt:

```
OLQ 100029 00 Signoff accepted - OLQ session terminated
```

```
V401 ENTER NEXT TASK CODE:
bye
```

Entering commands

Switching between menu/command mode

If you have signed on to CA OLQ and you see a screen of menu options, you are in CA OLQ's menu facility. Press [PF9] to swap to CA OLQ's command mode screen, which looks like this:

```
OLQ 107017 00 CA OLQ Release nn.n
OLQ 017019 Copyright(C) 2003 CA,Inc.
OLQ 091057 00 Please enter next command.
```

Command area of the screen

Type CA OLQ commands in the space above the CA OLQ messages you see on the screen above. You may notice that your terminal shows more or less space in the command area than you see in the above example. The size of the command portion is established at system generation by the INPUT LINE SIZE clause of the OLQ system generation statement. If the command portion of your screen is too small, ask your system administrator to adjust the value assigned to the INPUT LINE SIZE clause.

How to enter commands

Enter CA OLQ commands starting at the upper left corner of the screen. Depending on your preference, enter commands in lower case letters, upper case letters, or both. When CA OLQ interprets a command, it changes the display to upper case letters.

⚠️ Note: By issuing a DCUF SET UPLow command before signing on to CA OLQ, CA OLQ recognizes case in character strings and displays processed commands and headings as you entered them.
Separate each word in a command by at least one blank. At a 3270-type terminal, press &retsym. to continue a command on the next line. Press [Enter] to process the command. The processed command appears on the screen. You can edit the command and resubmit it to CA OLQ.

Some useful commands

The SELECT statement is the focus of this section. However, you'll need some of the commands listed below to facilitate your CA OLQ session:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIGNON</td>
<td>Initiates a CA OLQ session by defining a view of the database</td>
<td>signon ss</td>
</tr>
<tr>
<td>HELP</td>
<td>Explains how to use CA OLQ commands and displays information on your tables and subschema records</td>
<td>help table</td>
</tr>
<tr>
<td>SET</td>
<td>Sets session parameters in a signon profile, such as ACCESS</td>
<td>set access olq</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>Sets processing and display options for the session</td>
<td>options = olqheader</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>Displays a report</td>
<td>display</td>
</tr>
<tr>
<td>BYE</td>
<td>Ends a CA OLQ session</td>
<td>bye</td>
</tr>
</tbody>
</table>

Some of these commands are described below. For a complete description, see the *CA OLQ Reference Section*.

Command delimiters

CA OLQ provides a separator character, which lets you string many commands together. The system default is an exclamation mark (!). All the examples in this section show an exclamation mark to separate the SELECT statement from a DISPLAY command, which displays the report: `select * from dept ! display`.

CA OLQ also provides a comment character, which tells CA OLQ treat all text following the delimiter as a comment. The system default is a semi-colon (;). By using a comment character, you can save portions of a command that you want to edit and resubmit. This example shows a DISPLAY command typed over a SELECT statement CA OLQ has already processed:

```
SELECT * FROM DEPT
display; FROM DEPT
```

To change the separator and comment characters for your session, use the SET command:

```bash
SET SEPARATOR CHARACTER ' $'
```

`OLQ 092014 00` The SEPARATOR CHARACTER has been modified.

Signing on to a table or subschema

Issue a SIGNON command to tell CA OLQ what subschema to access for data. A SIGNON command is required for CA IDMS/DB subschemas. It is optional for tables.

To sign on to a specific table or subschema enter a SIGNON command: `signon table emp`.
CA OLQ responds that it is ready to retrieve data:

SIGNON TABLE EMP

CA OLQ 100021 00 Ready to retrieve data from subschema RU00371 CA OLQ 100022 00 Sc
hema: CA-IDMSDB Version: l
CA OLQ 100023 00 Database name: ASFDICT
CA OLQ 100025 00 Dictionary name: ASFDICT

Using the SELECT statement

You can use CA OLQ command mode to create your own SELECT statements, or you can let CA OLQ's menu facility build the statements for you when you select data. All the examples in this section use command mode to enter the SELECT statement.

Setting the access to IDMS

When you want to access SQL-defined tables, your default access setting must be set to IDMS. You can do this for your CA OLQ session by specifying:

```
set access idms
```

You are now able to use the SELECT statement to access SQL-defined tables.

**Note:** The examples of the SELECT statement in this section are not necessarily valid for SQL-defined tables.

For more information on using the SELECT statement with SQL-defined tables, refer to the CA IDMS SQL Reference Section.

Tailoring reports

**Tailoring your report**

The screen displays in this section show the direct results of the SELECT statement. The displays reflect default formatting options. For example, reports have the name of the source table as a title, fields aren’t edited, and groups of data are not separated by blank lines.

You can easily tailor your reports by using options available in CA OLQ's menu facility.

The following series of screens show a few of these options:

**Display the report**

In command mode, issue a SELECT statement to CA OLQ:

```
select deptid, lastname, salary, phone from emp order by deptid, lastname ! display
```

```
SELECT DEPTID, LASTNAME, SALARY, PHONE FROM EMP
ORDER BY DEPTID, LASTNAME ! DISPLAY
```

```
EMP REPORT
mm/dd/yy
```
Go to the menu facility

Swap to CA OLQ's menu facility by pressing [PF9].

**Note:** You can also move directly to a screen by entering the name of the screen on the command line.

CA OLQ displays the Menu screen:

```
CA OLQ Release nn.n
*** Menu ***
->
122000 Select an option and press the ENTER key.
```

<table>
<thead>
<tr>
<th>Pfkey</th>
<th>Select Option</th>
<th>Description</th>
<th>Command/Screen Name</th>
<th>Show Help</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Data Source for Report</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Choose data tables</td>
<td>TABle</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Choose subschema</td>
<td>SUBschema</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Retrieval Activity</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Choose records from selected subschema</td>
<td>REcord</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Choose columns for report</td>
<td>COLumn</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Retrieve data to build report</td>
<td>RETrieve</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Alter database access strategy</td>
<td>LINkage</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Processing Mode</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Execute or create a predefined routine</td>
<td>QFile</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>View existing or save current report</td>
<td>SAVe</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Submit batch report request</td>
<td>BATch</td>
<td></td>
</tr>
</tbody>
</table>

1=HELP 2=GLOBAL HELP 3=QUIT 4=MESSAGE 8=FWD

Change the column headings

On page 2 of the Menu screen, select the option to change column headers. CA OLQ brings you to the Report Format - Header screen. You can assign new column headings for your report and specify a character to underline the column headings. The default character is a hyphen (-).

```
CA OLQ Release nn.n
*** Report Format - Header ***
->
134000 Specify column headers and press the ENTER key.
```
When you press [Enter], CA OLQ displays a revised report:

CA OLQ Release nn.n

->

Press the ENTER key to go to the next page of the report.

Once you have entered the information, press [Enter]. CA OLQ displays this report:

CA OLQ Release nn.n

->

Press the ENTER key to go to the next page of the report.

Format the phone and salary data

Return to the Menu screen by pressing [PF6] and choose the option to change the display format of data. CA OLQ brings you to the Report Format - Picture screen where you can:
Edit the phone number display

Select options to display the SALARY field with a dollar sign and commas

CA OLQ displays this report once it processes the information entered on the Report Format - Picture screen:

Group the rows by department

Press [PF6] to return to the Menu screen and choose the option to sort the sequence of the report.

CA OLQ brings you to the Report Format - Sort screen. On this screen, you can indicate:

- The sort priority for the fields (which isn't really necessary because the ORDER BY clause of the SELECT statement sorted the report's contents)

- The field on which to group data for summary calculations. In this case, the DEPTID field is assigned Group Level # 1.

- Whether to display detail lines, summary lines (subtotals), or both. In this screen, both types of lines are chosen.
Display lines: Detail X and/or Summary X   Group by all _

Compute:
1=HELP   3=QUIT   4=MESSAGE   5=DISPLAY   6=MENU   11=HEADER

Total the salaries

Once CA OLQ processes the information provided above, it brings you to the Report Format - Group By screen where you can enter summary information for the report. In this example, the report will display the total salaries for each department. The SKIP LINES and SEPARATOR CHARACTER fields indicate that CA OLQ will:

- Insert one line between the detail and subtotal lines
- Write a separator line (composed of hyphens) between the detail and subtotal lines

<table>
<thead>
<tr>
<th>Group by: EMP.DEPTID</th>
<th>Seq</th>
<th>Sum</th>
<th>Avg</th>
<th>Max</th>
<th>Min</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMP</td>
<td>X</td>
<td>DEPTID</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X LASTNAME</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X PHONE</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X SALARY</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Skip lines after group 1   Separator character -

Compute:
1=HELP   3=QUIT   4=MESSAGE   5=DISPLAY   6=MENU   10=PICTURE

When you press [Enter], CA OLQ displays this report:

<table>
<thead>
<tr>
<th>Department ID</th>
<th>EMPLOYEE NAME</th>
<th>Salary</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>3100</td>
<td>DOUGH</td>
<td>$33,000.00</td>
<td>(617)445-8155</td>
</tr>
<tr>
<td>3100</td>
<td>GALLWAY</td>
<td>$33,000.00</td>
<td>(617)334-9155</td>
</tr>
<tr>
<td>3100</td>
<td>GARFIELD</td>
<td>$65,000.00</td>
<td>(617)332-1967</td>
</tr>
<tr>
<td>3100</td>
<td>GRANGER</td>
<td>$34,500.00</td>
<td>(617)334-8912</td>
</tr>
<tr>
<td>3100</td>
<td>HEAROWITZ</td>
<td>$33,000.00</td>
<td>(617)334-9634</td>
</tr>
<tr>
<td>3100</td>
<td>JACOBI</td>
<td>$55,000.00</td>
<td>(617)334-8912</td>
</tr>
<tr>
<td>3100</td>
<td>JENSEN</td>
<td>$37,000.00</td>
<td>(617)224-1955</td>
</tr>
<tr>
<td>3100</td>
<td>LITERATA</td>
<td>$37,500.00</td>
<td>(617)591-2323</td>
</tr>
</tbody>
</table>
The end result

The report shown above contains the same data as the original report. However, this report is easier to read and contains summary information for each department.

Retrieving Information from a Table

Retrieving All Columns

Use an asterisk (*) To retrieve all of the columns defined for a table, you can use an asterisk (*) in the SELECT statement. An asterisk is shorthand for all of the table's columns. By using an asterisk, you don't need to know the table's column names in order to retrieve data.

Example

To display the DEPT table, enter:

```
select * from dept ! display
```

<table>
<thead>
<tr>
<th>DEPTID</th>
<th>DEPTNAME</th>
<th>MGRID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>PERSONNEL</td>
<td>0013</td>
</tr>
<tr>
<td>2000</td>
<td>ACCOUNTING AND PAYROLL</td>
<td>0011</td>
</tr>
<tr>
<td>3100</td>
<td>INTERNAL SOFTWARE</td>
<td>0003</td>
</tr>
<tr>
<td>3200</td>
<td>COMPUTER OPERATIONS</td>
<td>0004</td>
</tr>
<tr>
<td>4000</td>
<td>PUBLIC RELATIONS</td>
<td>0007</td>
</tr>
<tr>
<td>5100</td>
<td>BRAINSTORMING</td>
<td>0015</td>
</tr>
<tr>
<td>5200</td>
<td>THERMOREGULATION</td>
<td>0349</td>
</tr>
<tr>
<td>5300</td>
<td>BLUE SKIES</td>
<td>0321</td>
</tr>
<tr>
<td>6666</td>
<td>EXECUTIVE ADMINISTRATION</td>
<td>0030</td>
</tr>
</tbody>
</table>

END OF REPORT

If all of the columns or all of the rows of the table don't fit on one screen, use these PF keys to scroll:

<table>
<thead>
<tr>
<th>Key</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>[PF7]</td>
<td>Scroll backward</td>
</tr>
<tr>
<td>[PF8]</td>
<td>Scroll forward</td>
</tr>
<tr>
<td>[PF10]</td>
<td>Scroll left</td>
</tr>
<tr>
<td>[PF11]</td>
<td>Scroll right</td>
</tr>
</tbody>
</table>

**Note:** If your PF key assignments are different than those described above, talk to your CA IDMS/DC system administrator.
Retrieving Selected Columns

Name the columns

To retrieve some of the columns from your table, list the columns you want in the SELECT statement, using a comma to separate each name. CA OLQ displays the columns in the order you enter them. Each column name becomes the column header in the report.

Note: To determine the column names used in the sample tables, look in Appendix A or retrieve all of the table's columns by using an asterisk in the SELECT statement.

Example

List each department's name and manager ID:

```
select deptname, mgrid from dept ! display
```

```
DEPT REPORT
mm/dd/yy

DEPTNAME | MGRID
---------|------
PERSONNEL | 0013
ACCOUNTING AND PAYROLL | 0011
INTERNAL SOFTWARE | 0003
COMPUTER OPERATIONS | 0004
PUBLIC RELATIONS | 0007
BRAINSTORMING | 0015
THERMOREGULATION | 0349
BLUE SKIES | 0321
EXECUTIVE ADMINISTRATION | 0030

END OF REPORT
```

Eliminating Duplicate Data

Use DISTINCT

A table can have one or more columns with duplicate data entries. For example, the JOBLIST table lists COMPUTER OPERATOR three times because three employees have that title. To retrieve unique, rather than duplicate, values in a column, use the keyword DISTINCT in the SELECT statement.

Example

List unique job titles in the company:

```
select distinct title from joblist ! display
```

```
JOBLIST REPORT
mm/dd/yy

TITLE
ACCOUNTANT
AP CLERK
AR CLERK
```

16-Jan-2018 609/898
Displaying Calculations in Columns

Compute new values

The SELECT statement displays calculated values, in addition to values stored in the table. For example, you might want to calculate a 6% bonus for each employee.

To display a calculated column, include an arithmetic expression in the column list following the SELECT statement. An arithmetic expression uses these operators:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Addition</td>
</tr>
<tr>
<td>-</td>
<td>Subtraction</td>
</tr>
<tr>
<td>*</td>
<td>Multiplication</td>
</tr>
<tr>
<td>/</td>
<td>Division</td>
</tr>
</tbody>
</table>

You can also use a built-in function to display a calculated column. Built-in functions are described later in this section.

Some examples

A few examples of arithmetic expressions appear below. These examples show you can:

- Leave spaces, or omit spaces, before and after these arithmetic operators: *, +, / . You must include a blank space before and after a minus (-) sign.
- Use parentheses to show how the arithmetic expression should be evaluated and to improve readability.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>salary/52</td>
<td>Evaluates a weekly salary</td>
</tr>
<tr>
<td>maxsalary - minsalary</td>
<td>Evaluates a salary range</td>
</tr>
<tr>
<td>(salary * 0.06) / 4</td>
<td>Evaluates a 6% bonus, to be distributed in 4 payments</td>
</tr>
</tbody>
</table>

Provide a heading for a calculation
To give each calculated column a heading, use the keyword AS and a heading name following the arithmetic expression. Enclose headings with two or more words in single quotation marks.

Example

List each employee's salary and end-of-year bonus, based on 6% of salary:

```
select empid, salary as &xq.annual salary', salary * 0.06 as bonus from emp ! display
```

```
EMPID ANNUAL SALARY BONUS
0007 80000.00 4800.0000
0019 37000.00 2220.0000
0020 55000.00 3300.0000
0021 20000.00 1200.0000
0024 33000.00 1980.0000
0027 33000.00 1980.0000
0028 34500.00 2070.0000
0029 33000.00 1980.0000
0030 240000.00 14400.0000
0035 37500.00 2250.0000
0120 18000.00 1080.0000
0127 18000.00 1080.0000
```

Putting Rows in Order

**Use ORDER BY**

To sort selected rows by the values in a column, use the ORDER BY clause. CA OLQ assumes you want rows in ascending order. If you want rows in descending order, specify DESCENDING as Example 2 shows below.

CA OLQ sorts selected rows by the first column named in the ORDER BY clause. It then sorts each group of rows sharing a common value in order of the second column named in the ORDER BY clause, and so on. For example, you might want to display a table of bank transactions in order of branch number, and within each branch number, in order of transaction date.

Example 1 -- Sorting on 1 column

For each employee in department 4000, list the ID, name, and hire date starting with the first person hired to the last person hired:

```
select empid, lastname, firstname, startdate as &xq.hire date' from emp where deptid = 4000 order by startdate ! display
```

```
EMPID LASTNAME  FIRSTNAME HIRE DATE
0476 ZEDI Betsy 760223
0158 JACKSON Jock 770707
0149 PENMAN Laura 770908
```
Example 2 -- Sorting on 2 columns

List the department ID, employee name, and hire date of all employees sorted by department. Within each department, list the employees in alphabetic descending order:

```
select deptid, lastname, startdate as 'hire date'
from emp
order by deptid, lastname descending
```

```
DEPTID     LASTNAME     HIRE DATE
3100       TYRO          801221
3100       LITERATA      800909
3100       JENSEN         820929
3100       JACOBI         811111
3100       HEAROWITZ      810909
3100       GRANGER        800527
3100       GARFIELD       770121
3100       GALLWAY        811010
3100       DOUGH          760808
4000       ZEDI           760223
4000       PENMAN         770908
4000       MCDOUGALL      800607
4000       JACKSON        770787
4000       BANK           780430
```

Sorting on calculated columns

If you want to sort the contents of your report based on values in a calculated column, specify the column number in the ORDER BY clause. Count the columns from left to right, beginning with 1.

Example

List salaries and anticipated 5% year-end bonus for employees in department 4000. List rows in order of smallest to largest bonus:

```
select lastname, salary, YEAR END BONUS= salary*0.005 from emp where deptid=4000
order by 4
```

```
LASTNAME  SALARY  YEAR END BONUS=
ANGELO     18000.00  90.00000
MCDOUGALL  18000.00  90.00000
JACKSON    34000.00  170.00000
ZEDI       37000.00  185.00000
PENMAN     39000.00  195.00000
BANK       80000.00  400.00000
```
Retrieving Selected Rows

Contents

- A simple comparison (see page 613)
- Complex comparisons (see page 615)
- Comparisons to a list of values (see page 617)
- Exclusive comparisons (see page 617)
- Range comparisons (see page 618)
- Character string comparisons (see page 619)

Use WHERE

You can retrieve selected rows from a table by specifying selection criteria in the WHERE clause. A WHERE clause contains one or more comparison expressions. A comparison expression compares one value to another value. The simplest comparison expression compares column values to a constant. If the comparison expression is true, CA OLQ selects the row for the report.

Within the WHERE clause, you can select rows by using these symbols and keywords:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comparison operators</td>
<td>Compares each value in a column to another value.</td>
</tr>
<tr>
<td>(for example, = and &lt;=)</td>
<td></td>
</tr>
<tr>
<td>AND and OR</td>
<td>Connects comparison expressions. AND is true if all the comparisons are true. OR is true if any of the comparisons is true.</td>
</tr>
<tr>
<td>IN</td>
<td>Compares a column's values to a list of values. The expression is true if the value in the column equals one of the listed values.</td>
</tr>
<tr>
<td>NOT</td>
<td>Negates the comparison expression. That is, if the comparison expression is true, NOT returns a value of false so the row is not selected.</td>
</tr>
<tr>
<td>BETWEEN</td>
<td>Compares each value in a column to a specified range, including the starting and ending range values.</td>
</tr>
<tr>
<td>LIKE</td>
<td>Compares a character string to a mask (that is, pattern).</td>
</tr>
<tr>
<td>Arithmetic expression</td>
<td>Compares a value in a column to the result of an arithmetic expression.</td>
</tr>
<tr>
<td>Built-in function</td>
<td>Compares each value in a column to a value calculated by a predefined function.</td>
</tr>
</tbody>
</table>

CA OLQ does not support NULL, an ANSI-standard keyword, for this release.

A simple comparison

Defining a comparison expression

As stated above, the simplest comparison expression compares each value in a column to a constant. The constant can be either:

- Numeric (that is, decimal, integer, real, hexadecimal, binary, or multibit binary).
• Character. A character constant is enclosed in single quotation marks.

The symbols that compare one value to the other are:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Equal to</td>
</tr>
<tr>
<td>&lt;&gt; or ¬=</td>
<td>Not equal to</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to</td>
</tr>
</tbody>
</table>

If you prefer, you can also use these symbols: EQ, NE, GT, LT, GE, and LE.

**Example 1 -- Comparing a number**

List all the employees that work in department 4000:

```sql
select deptid, firstname, lastname
from emp
where deptid = 4000 ! display
```

<table>
<thead>
<tr>
<th>DEPTID</th>
<th>FIRSTNAME</th>
<th>LASTNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>4000</td>
<td>MONTE</td>
<td>BANK</td>
</tr>
<tr>
<td>4000</td>
<td>MICHAEL</td>
<td>ANGELO</td>
</tr>
<tr>
<td>4000</td>
<td>CAROL</td>
<td>MCDougall</td>
</tr>
<tr>
<td>4000</td>
<td>LAURA</td>
<td>PENMAN</td>
</tr>
<tr>
<td>4000</td>
<td>JOCK</td>
<td>JACKSON</td>
</tr>
<tr>
<td>4000</td>
<td>BETSY</td>
<td>ZEDI</td>
</tr>
</tbody>
</table>

**Example 2 -- Comparing a character**

List the department and names of all female employees:

```sql
select deptid, firstname, lastname
from emp
where sex = 'f' ! display
```

<table>
<thead>
<tr>
<th>DEPTID</th>
<th>FIRSTNAME</th>
<th>LASTNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>3100</td>
<td>JENNIFER</td>
<td>GARFIELD</td>
</tr>
<tr>
<td>3100</td>
<td>JULIE</td>
<td>JENSEN</td>
</tr>
<tr>
<td>3100</td>
<td>JANE</td>
<td>DOUGH</td>
</tr>
<tr>
<td>6666</td>
<td>HENRIETTA</td>
<td>HENDON</td>
</tr>
<tr>
<td>4000</td>
<td>CAROL</td>
<td>MCDougall</td>
</tr>
<tr>
<td>4000</td>
<td>LAURA</td>
<td>PENMAN</td>
</tr>
<tr>
<td>4000</td>
<td>BETSY</td>
<td>ZEDI</td>
</tr>
</tbody>
</table>

**Example 3 -- Selecting lower values**

List the hire date, in ascending order, and names of all employees employed before January 1, 1978:

```sql
select startdate as 'hire date', firstname, lastname
from emp
where startdate < 780101 order by startdate ! display
```
Example 4 -- Using a calculation

List all job classes where the salary range is less than $3,000:

```
select class, minsalary as 'minimum salary', maxsalary as 'maximum salary'
from jobclass
where (maxsalary - minsalary) < 3000 ! display
```

JOBCLASS REPORT

<table>
<thead>
<tr>
<th>CLASS</th>
<th>MINIMUM SALARY</th>
<th>MAXIMUM SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>18000.00</td>
<td>20000.00</td>
</tr>
<tr>
<td>33</td>
<td>37000.00</td>
<td>39000.00</td>
</tr>
<tr>
<td>42</td>
<td>33000.00</td>
<td>35000.00</td>
</tr>
</tbody>
</table>

END OF REPORT

Complex comparisons

Using AND and OR

You can specify more than one comparison expression to select rows from a table by connecting each expression with an AND or an OR:

- **AND** Retrieves the row if *each* comparison expression is true
- **OR** Retrieves the row if *any* comparison expression is true

If a WHERE clause contains both AND and OR, CA OLQ evaluates the OR expressions first. For example, to process the example below, CA OLQ determines whether the employee's department is 4000 or whether the employee was hired before January 1, 1980. If either of these conditions is true, CA OLQ determines if the employee's job class is 21:

```
where class=21 and deptid=4000 or startdate<800101
```

**Note**: You can use parentheses to indicate the order in which CA OLQ evaluates the expressions. You can also use parentheses to improve the readability and accuracy of complex expressions.

Example 1 -- Using AND
List information on all employees who were hired before January 1, 1980 and whose salary exceeds $50,000:

```
select empid, lastname, startdate as 'hire date', salary from emp where (startdate < 800101) and (salary > 50000) ! display
```

```
<table>
<thead>
<tr>
<th>EMPID</th>
<th>LASTNAME</th>
<th>HIRE DATE</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>RUPEE</td>
<td>750223</td>
<td>76000.00</td>
</tr>
<tr>
<td>0003</td>
<td>GARFIELD</td>
<td>770121</td>
<td>65000.00</td>
</tr>
<tr>
<td>0007</td>
<td>BANK</td>
<td>780430</td>
<td>80000.00</td>
</tr>
<tr>
<td>0030</td>
<td>HENDON</td>
<td>731121</td>
<td>240000.00</td>
</tr>
<tr>
<td>0471</td>
<td>PAPAZEUS</td>
<td>780907</td>
<td>90000.00</td>
</tr>
<tr>
<td>0472</td>
<td>WILDER</td>
<td>790716</td>
<td>90000.00</td>
</tr>
</tbody>
</table>
END OF REPORT
```

**Example 2 -- Using parentheses with AND and OR**

Parentheses determine how CA OLQ evaluates complex comparisons. Each of the SELECT statements shown below have the same three comparison expressions. However, the first SELECT statement uses parentheses to group the expressions connected by AND. The second groups the expressions connected by OR.

**SELECT Statement 1**

List the names, birthdays, and salaries of all employees who are either:

- Female, born before January 1, 1947 or
- Making less than $25,000

```
select lastname, firstname, birthdate, salary from emp where (sex = 'F' and birthdate < 470101) or (salary < 25000) order by birthdate ! display
```

```
<table>
<thead>
<tr>
<th>LASTNAME</th>
<th>FIRSTNAME</th>
<th>BIRTHDATE</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>HENDON</td>
<td>HENRIETTA</td>
<td>331006</td>
<td>240000.0</td>
</tr>
<tr>
<td>ZEDI</td>
<td>BETSY</td>
<td>401229</td>
<td>37000.00</td>
</tr>
<tr>
<td>PENMAN</td>
<td>LAURA</td>
<td>440504</td>
<td>39000.00</td>
</tr>
<tr>
<td>GARFIELD</td>
<td>JENNIFER</td>
<td>450818</td>
<td>65000.00</td>
</tr>
<tr>
<td>TYRO</td>
<td>RALPH</td>
<td>551225</td>
<td>20000.00</td>
</tr>
<tr>
<td>ANGELO</td>
<td>MICHAEL</td>
<td>570405</td>
<td>18000.00</td>
</tr>
<tr>
<td>MCDougall</td>
<td>CAROL</td>
<td>590304</td>
<td>18000.00</td>
</tr>
</tbody>
</table>
END OF REPORT
```

**SELECT Statement 2**

In comparison, list the same information on employees who:

- Are female and
- Earn less than $25,000 or were born before January 1, 1947

```
select lastname, firstname, birthdate, salary from emp where (sex = 'F' and birthdate < 470101 or salary < 25000) order by birthdate ! display
```

```
<table>
<thead>
<tr>
<th>LASTNAME</th>
<th>FIRSTNAME</th>
<th>BIRTHDATE</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>HENDON</td>
<td>HENRIETTA</td>
<td>331006</td>
<td>240000.0</td>
</tr>
<tr>
<td>ZEDI</td>
<td>BETSY</td>
<td>401229</td>
<td>37000.00</td>
</tr>
<tr>
<td>PENMAN</td>
<td>LAURA</td>
<td>440504</td>
<td>39000.00</td>
</tr>
<tr>
<td>GARFIELD</td>
<td>JENNIFER</td>
<td>450818</td>
<td>65000.00</td>
</tr>
<tr>
<td>TYRO</td>
<td>RALPH</td>
<td>551225</td>
<td>20000.00</td>
</tr>
<tr>
<td>ANGELO</td>
<td>MICHAEL</td>
<td>570405</td>
<td>18000.00</td>
</tr>
<tr>
<td>MCDougall</td>
<td>CAROL</td>
<td>590304</td>
<td>18000.00</td>
</tr>
</tbody>
</table>
END OF REPORT
```
Comparisons to a list of values

Using IN

To compare a value to one of several values in a list, use the IN keyword. The IN keyword is a short way of coding two or more comparison expressions connected by an OR. For example, both of these SELECT statements yield the same result:

```
select empid from emp where deptid in (3100, 4000)
select empid from emp where (deptid = 3100) or (deptid = 4000)
```

Separate each value in the list by a comma. A blank following the comma is optional.

Example

List all employees whose job falls into one of these classes: 11, 21, 43, or 71 (the report shown below indicates that there are no employees that have a job in class 11):

```
select firstname, lastname, class from emp where class in (11, 21, 43, 71) order by class ! display
```

```
FIRSTNAME  LASTNAME  CLASS
RALPH       TYRO       21
MICHAEL     ANGELO     21
CAROL       MCDougall  21
JULIE       JENSEN     43
LARRY       LITERATA   43
JENNIFER    GARFIELD   71
```

Exclusive comparisons

Using NOT

You can retrieve all rows that are exceptions to the comparison expression by using the keyword NOT. For example, you can retrieve information on all employees except those that work in departments 6666 and 3100.

A few examples appear below:

<table>
<thead>
<tr>
<th>Example</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>where not (deptid &lt; 4000)</td>
<td>All rows where the department id is greater than or equal to 4000</td>
</tr>
</tbody>
</table>
Example

List all employees except those with manager id 0007:

```
select firstname, lastname, mgrid from emp where not (mgrid = 0007) order by mgrid !
```

<p>| EMP REPORT |
| mm/dd/yy   |</p>
<table>
<thead>
<tr>
<th>FIRSTNAME</th>
<th>LASTNAME</th>
<th>MGRID</th>
</tr>
</thead>
<tbody>
<tr>
<td>LARRY</td>
<td>LITERATA</td>
<td>0003</td>
</tr>
<tr>
<td>RALPH</td>
<td>TYRO</td>
<td>0003</td>
</tr>
<tr>
<td>JULIE</td>
<td>JENSEN</td>
<td>0003</td>
</tr>
<tr>
<td>JAMES</td>
<td>JACOBI</td>
<td>0003</td>
</tr>
<tr>
<td>VLADIMIR</td>
<td>HEAROWITZ</td>
<td>0003</td>
</tr>
<tr>
<td>JANE</td>
<td>DOUGHER</td>
<td>0003</td>
</tr>
<tr>
<td>PERCY</td>
<td>GRANGER</td>
<td>0003</td>
</tr>
<tr>
<td>JAMES</td>
<td>GALLWAY</td>
<td>0003</td>
</tr>
<tr>
<td>MONTE</td>
<td>BANK</td>
<td>0030</td>
</tr>
<tr>
<td>JENNIFER</td>
<td>GARFIELD</td>
<td>0030</td>
</tr>
<tr>
<td>HENRIETTA</td>
<td>HENDON</td>
<td>0030</td>
</tr>
<tr>
<td>JOHN</td>
<td>RUPEE</td>
<td>0030</td>
</tr>
<tr>
<td>THEMIS</td>
<td>PAPAZIEUS</td>
<td>0030</td>
</tr>
<tr>
<td>ROBBY</td>
<td>WILDER</td>
<td>0030</td>
</tr>
</tbody>
</table>

- 1 -

Range comparisons

Using BETWEEN

To retrieve rows from a table where the values of one column fall in a range of values, use BETWEEN. BETWEEN selects all rows that have values in between and equal to the starting and ending values of the specified range.

Example

List the ID, job class, and salary of all employees who earn from $33,000 to $39,000:

```
select empid, class, salary from emp where salary between 33000 and 39000 order by salary !
```

<p>| EMP REPORT |
| mm/dd/yy   |</p>
<table>
<thead>
<tr>
<th>EMPID</th>
<th>CLASS</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0024</td>
<td>42</td>
<td>33000.00</td>
</tr>
<tr>
<td>0027</td>
<td>42</td>
<td>33000.00</td>
</tr>
<tr>
<td>0029</td>
<td>42</td>
<td>33000.00</td>
</tr>
<tr>
<td>0158</td>
<td>42</td>
<td>34000.00</td>
</tr>
<tr>
<td>0028</td>
<td>42</td>
<td>34500.00</td>
</tr>
<tr>
<td>0019</td>
<td>43</td>
<td>37000.00</td>
</tr>
<tr>
<td>0476</td>
<td>33</td>
<td>37000.00</td>
</tr>
<tr>
<td>0035</td>
<td>43</td>
<td>37500.00</td>
</tr>
<tr>
<td>0149</td>
<td>33</td>
<td>39000.00</td>
</tr>
</tbody>
</table>

END OF REPORT
Character string comparisons

Using LIKE

You can compare an alphanumeric field to a mask (pattern) that contains alphanumeric characters and wild card symbols. For example, you might want to retrieve information on jobs that have CLERK in the job title.

To code the pattern, use the wild card symbols described below. If the pattern contains embedded blanks, enclose it in single quotation marks.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent (%)</td>
<td>Specifies from 0 to any number of unknown characters</td>
</tr>
<tr>
<td>Underscore (_)</td>
<td>Specifies a single unknown character</td>
</tr>
</tbody>
</table>

Some examples using these symbols appear below:

**Example**

<table>
<thead>
<tr>
<th>Selects the row if the value contains</th>
</tr>
</thead>
<tbody>
<tr>
<td>%m%</td>
</tr>
<tr>
<td><em>m</em></td>
</tr>
</tbody>
</table>

**Note:** To improve CA OLQ's performance, use conditional operators, rather than LIKE, to perform character string comparisons. For example, use `where firstname = 'b'` to retrieve all employees whose first name begins with B.

**Example 1 -- Using a % sign**

List the names of all employees with initials JG:

```sql
select firstname, lastname from emp where firstname like &xq.j% and lastname like &xq.g% ! display
```

```
EMP REPORT
mm/dd/yy

FIRSTNAME            LASTNAME
JENNIFER             GARFIELD
JAMES                GALLWAY

END OF REPORT
```

**Example 2 -- Using two % signs**

List all employees whose name contains the letter Z:

```sql
select lastname from emp where lastname like &xq.%z% ! display
```

```
EMP REPORT
mm/dd/yy

LASTNAME
HEAROWITZ
```

Example 3 -- Using an underscore (_)

List all employees whose name contains 5 letters, beginning with J:

```sql
select firstname, lastname from emp where firstname like &xq.j____' ! display
```

EMP REPORT

<table>
<thead>
<tr>
<th>FIRSTNAME</th>
<th>LASTNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>JULIE</td>
<td>JENSEN</td>
</tr>
<tr>
<td>JAMES</td>
<td>JACOBI</td>
</tr>
<tr>
<td>JAMES</td>
<td>GALLWAY</td>
</tr>
</tbody>
</table>

END OF REPORT

Using Built-In Functions

Definition

CA OLQ provides many predefined functions that evaluate expressions and return results. These functions, called **built-in functions**, can be used anywhere you would normally specify arithmetic or comparison expressions. The built-in functions fall into these categories:

<table>
<thead>
<tr>
<th>Category</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>Perform operations on character strings, such as concatenating BUSY and BEE</td>
</tr>
<tr>
<td>Arithmetic</td>
<td>Perform arithmetic operations, such as rounding 5.77 to 6</td>
</tr>
<tr>
<td>Trigonometric</td>
<td>Perform mathematical calculations, such as evaluating the cosine of 30 degrees</td>
</tr>
<tr>
<td>Date</td>
<td>Perform calculations on dates, such as evaluating the number of days between January 14, 1956 and June 26, 1987</td>
</tr>
</tbody>
</table>

The examples shown below are only a few of the built-in functions you can use. For information on all the built-in functions CA OLQ provides, see the **CA OLQ Reference Section**.

Example 1 -- Retrieving a substring

Create a 3-letter department code for each department by using the first three letters of the department name:

```sql
select substring(deptname,1,3) as &xq.dept code' from dept ! display
```

DEPT REPORT

<table>
<thead>
<tr>
<th>DEPT CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC</td>
</tr>
<tr>
<td>BLU</td>
</tr>
<tr>
<td>BRA</td>
</tr>
<tr>
<td>COM</td>
</tr>
<tr>
<td>EXE</td>
</tr>
</tbody>
</table>

END OF REPORT
Example 2 -- Concatenating names

List the last name of each employee, followed by a comma and a blank, and the employee’s first name. The EXTRACT function deletes all trailing blanks from the employee’s last name. The CONCATENATE function strings together:

- The character string returned by the EXTRACT function
- The literal containing the comma and the blank
- The employee’s first name

select concatenate(extract(lastname), ', ', firstname) from emp ! display

<table>
<thead>
<tr>
<th>LASTNAME</th>
<th>BIRTHDATE</th>
<th>WEEKDAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUPEE, JOHN</td>
<td>330219</td>
<td>SUNDAY</td>
</tr>
<tr>
<td>GARFIELD, JENNIFER</td>
<td>450818</td>
<td>SATURDAY</td>
</tr>
<tr>
<td>BANK, MONTE</td>
<td>500101</td>
<td>SUNDAY</td>
</tr>
<tr>
<td>JENSEN, JULIE</td>
<td>480730</td>
<td>FRIDAY</td>
</tr>
<tr>
<td>JACOBI, JAMES</td>
<td>401101</td>
<td>FRIDAY</td>
</tr>
<tr>
<td>TYRO, RALPH</td>
<td>551225</td>
<td>SUNDAY</td>
</tr>
<tr>
<td>DOUGH, JANE</td>
<td>510329</td>
<td>THURSDAY</td>
</tr>
<tr>
<td>HEAROWITZ, VLADIMIR</td>
<td>560425</td>
<td>WEDNESDAY</td>
</tr>
<tr>
<td>GRANGER, PERCY</td>
<td>580222</td>
<td>SATURDAY</td>
</tr>
<tr>
<td>GALLWAY, JAMES</td>
<td>471006</td>
<td>MONDAY</td>
</tr>
<tr>
<td>HENDON, HENRIETTA</td>
<td>331006</td>
<td>FRIDAY</td>
</tr>
<tr>
<td>LITERATA, LARRY</td>
<td>558430</td>
<td>SATURDAY</td>
</tr>
</tbody>
</table>

Example 3 -- Determining a date’s weekday

List the day of the week each employee was born. The GWEEKDAY function accepts a Gregorian date and returns the day of the week on which that date falls:

select lastname, birthdate, gweekday(birthdate) as weekday from emp ! display

<table>
<thead>
<tr>
<th>LASTNAME</th>
<th>BIRTHDATE</th>
<th>WEEKDAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUPEE</td>
<td>330219</td>
<td>SUNDAY</td>
</tr>
<tr>
<td>GARFIELD</td>
<td>450818</td>
<td>SATURDAY</td>
</tr>
<tr>
<td>BANK</td>
<td>500101</td>
<td>SUNDAY</td>
</tr>
<tr>
<td>JENSEN</td>
<td>480730</td>
<td>FRIDAY</td>
</tr>
<tr>
<td>JACOBI</td>
<td>401101</td>
<td>FRIDAY</td>
</tr>
<tr>
<td>TYRO</td>
<td>551225</td>
<td>SUNDAY</td>
</tr>
<tr>
<td>DOUGH</td>
<td>510329</td>
<td>THURSDAY</td>
</tr>
<tr>
<td>HEAROWITZ</td>
<td>560425</td>
<td>WEDNESDAY</td>
</tr>
<tr>
<td>GRANGER</td>
<td>580222</td>
<td>SATURDAY</td>
</tr>
<tr>
<td>GALLWAY</td>
<td>471006</td>
<td>MONDAY</td>
</tr>
<tr>
<td>HENDON</td>
<td>331006</td>
<td>FRIDAY</td>
</tr>
<tr>
<td>LITERATA</td>
<td>558430</td>
<td>SATURDAY</td>
</tr>
</tbody>
</table>
Example 4 -- Calculating employee age

List each employee's name and age. The DATEDIF function determines the number of days between today's date (January 13, 1999) and the employee's birthday. The NEXT-INT-EQLO function rounds the number of years to the next lowest integer:

```
select firstname, lastname, next-int-eqlo(datedif(990113, birthdate)/365) as employee age'
from emp
```

```
EMP REPORT
mm/dd/yy
FIRSTNAME     LASTNAME  EMPLOYEE AGE
JOHN          RUPEE     54
JENNIFER      GARFIELD  42
MONTE         BANK      38
JULIE         JENSEN    39
JAMES         JACOOBI   47
RALPH         TYRO      32
JANE          DOUGH     36
VLADIMIR      HEAROWITZ 31
PERCY         GRANGER   29
JAMES         GALLWAY   40
HENRIETTA     HENDON    54
LARRY         LITERATA  32
MICHAEL       ANGELO    30
CAROL         MCDougall 28
```

Testing Your Knowledge

Using the sample tables in Appendix A, "Sample Tables and Database", code a SELECT statement for each of the queries listed below. , contains one possible answer. Remember, there can be more than one way to achieve the same result when you use the SELECT statement.

1. List the salary range for each job class

2. List all employees who have both a:
   - Manager with ID 0007 or 0003
   - Job that begins with the number 3

3. Identify all employees whose sex code was entered incorrectly (that is, is not M or F)

4. List the number of years employees have from January 1, 1988 until they reach retirement age (65)
Summarizing Information

**Summary lines**  This section tells you how to create reports containing *summary lines*, rather than *detail lines*. A detail line contains information on individual rows in a table. Summary lines contain summary information on detail lines. For example, a line containing salary information on Henrietta Hendon is a detail line. A line containing total salaries for all employees is a summary line.

**Aggregate functions**

To summarize information, CA OLQ provides the *aggregate functions* shown below. An aggregate function is a type of built-in function that evaluates all the values in a column and returns a single value.

<table>
<thead>
<tr>
<th>Function</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>COUNT</td>
<td>Counts the number of rows</td>
</tr>
<tr>
<td>SUM</td>
<td>Supplies a total value for the named column</td>
</tr>
<tr>
<td>AVG</td>
<td>Supplies an average value for the named column</td>
</tr>
<tr>
<td>MIN</td>
<td>Supplies the lowest value in the named column</td>
</tr>
<tr>
<td>MAX</td>
<td>Supplies the highest value in the named column</td>
</tr>
<tr>
<td>STD</td>
<td>Standard deviation</td>
</tr>
<tr>
<td>VAR</td>
<td>Variance</td>
</tr>
</tbody>
</table>

**How to use aggregate functions**

You use aggregate functions in the list of column names or expressions following the SELECT statement. The function is calculated by giving the function (for example, SUM) followed by a column name or value in parentheses. You can use all aggregate functions, except COUNT, with:

- Values in a column: `sum(salary)`
- An arithmetic expression: `sum(salary * 0.06)`
- Any combination of the items listed above

To count the number of selected rows, use COUNT followed by an asterisk in parentheses: `count(*)`. CA OLQ displays the value in the report.

**Example -- Aggregate functions in column lists**

List the number of employees within the company, and the company's total salary payment and average salary payment:

```
select count(*) as &xq.number of employees', sum(salary) as &xq.total salary', avg (salary) as &xq.average salary' from emp ! display
```
For more information, see the following topics:

- Summarizing Data In Groups (see page 624)
- Specifying a Condition For a Group (see page 625)
- Summarizing Information From Another Table (see page 625)
- Testing Your Knowledge 1 (see page 626)

### Summarizing Data In Groups

#### Groups

You can display summary information on groups. A group is a collection of detail lines that share a common value in one or more columns. For example, you can display summary salary information for each department. Summarizing information on groups in similar to break processing for those familiar with that reporting terminology.

#### Use GROUP BY

To summarize information for groups of values, use the GROUP BY clause. The GROUP BY clause indicates which columns to group. For example, this clause groups all rows that share the same department ID:

```
group by deptid
```

You can specify up to 31 columns in the GROUP BY clause. For example, you can group rows by department, and within the department, by job ID:

```
group by deptid, jobid
```

When you group rows, each column listed in the SELECT statement, except those named in the GROUP BY clause, must be acted upon by an aggregate function, such as AVG or MIN. For example, you might group rows by department ID to return average and minimum salaries for each department. CA OLQ displays one row for each group it evaluates.

#### Example 1 -- Grouping based on 1 column

List the number of employees in each department and the department's total and average salaries:

```sql
select deptid, count(*) as &xq.number of employees', sum(salary) as &xq.total salary',
   avg(salary) as &xq.average salary' from emp group by deptid ! display
```

```plaintext
DEPTID  NUMBER OF EMPLOYEES TOTAL SALARY AVERAGE SALARY
    3100        9  348000.00        38666.66
    4000        6  226000.00        37666.66
    6666        4  496000.00       124000.00
```
For each department, list the number of employees, total salary, and average salary by gender. Notice that a sex code was entered incorrectly for an employee in department 6666:

```
select deptid, sex, count(*) as number of employees', sum(salary) as total salary', avg(salary) as average salary'
from emp
group by deptid, sex
```

### EMP REPORT

<table>
<thead>
<tr>
<th>DEPTID</th>
<th>SEX</th>
<th>NUMBER OF EMPLOYEES</th>
<th>TOTAL SALARY</th>
<th>AVERAGE SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>3100</td>
<td>F</td>
<td>3</td>
<td>135000.00</td>
<td>45000.00</td>
</tr>
<tr>
<td>3100</td>
<td>M</td>
<td>6</td>
<td>213000.00</td>
<td>35500.00</td>
</tr>
<tr>
<td>4000</td>
<td>F</td>
<td>3</td>
<td>94000.00</td>
<td>31333.33</td>
</tr>
<tr>
<td>4000</td>
<td>M</td>
<td>3</td>
<td>132000.00</td>
<td>44000.00</td>
</tr>
<tr>
<td>6666</td>
<td>F</td>
<td>1</td>
<td>240000.00</td>
<td>240000.00</td>
</tr>
<tr>
<td>6666</td>
<td>M</td>
<td>2</td>
<td>166000.00</td>
<td>83000.00</td>
</tr>
<tr>
<td>6666</td>
<td>N</td>
<td>1</td>
<td>90000.00</td>
<td>90000.00</td>
</tr>
</tbody>
</table>

### Specifying a Condition For a Group

**Use HAVING**

To apply selection criteria to the result of a GROUP BY clause, use a HAVING clause after the GROUP BY clause. The HAVING clause is similar to a WHERE clause, except that it applies to summary rows only. As in the WHERE clause, you can connect multiple conditional expressions by using AND and OR.

**Example**

List the departments where the total salary exceeds $300,000:

```
select deptid, sum(salary) as total salary', avg(salary) as average salary'
from emp
group by deptid
having sum(salary) > 300000
```

### EMP REPORT

<table>
<thead>
<tr>
<th>DEPTID</th>
<th>TOTAL SALARY</th>
<th>AVERAGE SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>3100</td>
<td>348000.00</td>
<td>38666.66</td>
</tr>
<tr>
<td>6666</td>
<td>496000.00</td>
<td>124000.00</td>
</tr>
</tbody>
</table>

### Summarizing Information From Another Table

**Using a nested SELECT statement**

A SELECT statement within a WHERE clause is called a nested SELECT. You can use a nested SELECT statement to retrieve information from one table based on summary information in another table.

**Example -- COUNT in a nested SELECT**

List the jobs held by more than one employee. The SELECT statement contains a nested SELECT that returns the number of employees assigned to each job ID. The primary SELECT statement determines if the number is greater than 1:
Testing Your Knowledge 1

Using the sample tables in num=A.Sample Tables and Database, code a SELECT statement for each of the queries listed below. num=B.Answers, contains one possible answer. Remember, there can be more than one way to achieve the same result when you use the SELECT statement.

1. For each manager in table EMP, determine the number of staff reporting to the manager and the average salary of the staff members.

2. List the number of jobs greater than 3 that are assigned to classes 10 and 50. Display the report in order by the number of jobs.

3. For departments 6666, 4000, and 3100, list the minimum and maximum salaries within the department, provided the average departmental salary in greater than $37,800.

4. List information on employees earning less than the average salary.

Joining Tables or Database Records

Tables

In many cases, you want to retrieve information from two or more tables. For example, to retrieve the name of each department manager, you must get the manager's ID from the DEPT table and the name that corresponds to the ID from the EMP table. With CA OLQ, you can obtain information from two tables by joining the tables based on a column both tables have in common.

Database records

Additionally, if you are using CA IDMS/DB database records, you can join the records based on a common value or you can join the records by specifying a record-to-record set relationship. For example, to retrieve information on employees and where they work, you would use the OFFICE-EMPLOYEE set of the Employee Database.

Tables and records

In some cases, you will want to retrieve information that exists in a table and in a record; for example, a CA IDMS/DB record (DEPARTMENT) and a table (EMP).

This section explains how to retrieve data from each of these data structures.
Joining Tables

Contents
- Joining different tables (see page 627)
- Joining a table to itself (see page 630)
- Comparing a column to more than one value (see page 632)

You can join tables that share a column of common values. For example, you can join:

- Two or more different tables.
- A table to itself when one column contains similar values to another column. For example, table EMP contains a column of manager IDs. You would join table EMP to itself to determine the name of an employee’s manager.

Both of these topics are described below. This section also describes a method to translate a nested SELECT that retrieves more than one value into a SELECT statement that is a join operation.

Joining different tables

Associate one column with another

To join tables together, each table must have at least one column that corresponds to a column in another table. You join tables together by equating these columns in the WHERE clause of the SELECT statement. The WHERE clause defines the join condition. This figure joins the EMP and DEPT tables by equating the department ID values in both tables:

<table>
<thead>
<tr>
<th>EMP table</th>
<th>DEPT table</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPNAME</td>
<td>DEPTID</td>
</tr>
<tr>
<td>Jennifer Lanzarotta</td>
<td>3100</td>
</tr>
<tr>
<td>Bart Elopoulos</td>
<td>3200</td>
</tr>
<tr>
<td>Reginald Mahoney</td>
<td>4000</td>
</tr>
<tr>
<td>Duc Ho</td>
<td>2000</td>
</tr>
<tr>
<td>Daniel Ortega</td>
<td>1000</td>
</tr>
<tr>
<td>Edith Jones</td>
<td>5100</td>
</tr>
<tr>
<td>Anthea Poznanski</td>
<td>0100</td>
</tr>
<tr>
<td>Ankur Sahu</td>
<td>5300</td>
</tr>
<tr>
<td>William Sternbach</td>
<td>5200</td>
</tr>
</tbody>
</table>

Join rows in EMP table and DEPT table that have DEPTID equal to 5300 to list information about department 5300 and its employees.
Which columns can you use?

The columns should contain comparable data. For example, you could compare the EMPID column in table EMP with the MGRID column in table DEPT. Both columns contain employee ID values.

Choosing a column

In some cases, the tables you want to join will have more than one corresponding column. For example, you can join the EMP and DEPT tables by comparing:

- The DEPTID column defined in both tables, or
- The EMPID column defined in the EMP table and the MGRID column defined in the DEPT table

If you compare the department ID values, CA OLQ retrieves information on all employees and their departments. If you compare employee and manager ID values, CA OLQ retrieves information on all employees who are department managers and the departments they manage.

Qualifying column names

When the tables you want to join have the same names for some or all of the columns (like the DEPTID column in the EMP and DEPT tables), qualify the column names by specifying the table name, followed by a period and the column name: `dept.deptid`. In fact, its a good idea to qualify all column names in join operations to make the SELECT statement easier to read.

Coding the SELECT statement

To join tables in a SELECT statement:

1. Name selected columns from any or all of the tables in the column list following the SELECT keyword: `dept.deptid, emp.lastname`.
2. Name the tables, separated by a comma, in the FROM clause of the SELECT statement: `dept, emp`. The order of the tables is not important.
3. Compare the values of the associated columns in the WHERE clause: `dept.deptid = emp.deptid`. The WHERE clause can contain more than one comparison expression, as shown in Example 2 below.

Example 1 -- Name the department managers

List information on each department manager. The SELECT statement joins the DEPT and EMP tables by getting rows from both tables where the manager ID in the department table is the same as the employee ID in the EMP table:

```
select dept.deptid, dept.deptname, emp.firstname, emp.lastname from dept, emp where dept.mgrid = emp.empid ! display
```
Example 2 -- Name employees hired before their manager

Assuming that employee IDs are assigned sequentially, list all employees who have worked at the company longer than their manager; that is, those employees who have a lower ID than that of the department's manager. The SELECT statement joins the EMP and DEPT tables by retrieving all rows where:

- The employee's department is the same as the manager's department, and
- The employee's ID is less than the manager's ID

```
select emp.empid, emp.lastname, emp.firstname, dept.mgrid from emp, dept
where (dept.deptid = emp.deptid) and (emp.empid < dept.mgrid)
```

Joining more than two tables

If you need to join more than two tables, specify a join condition for each pair of tables. That is, to join three tables, you'll need at least two join conditions. For example, to join the EMP, DEPT, and JOBCLASS tables, you could join the tables this way:

<table>
<thead>
<tr>
<th>Tables</th>
<th>Join condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMP and DEPT</td>
<td>dept.deptid = emp.deptid</td>
</tr>
<tr>
<td>EMP and JOBCLASS</td>
<td>emp.class = jobclass.class</td>
</tr>
</tbody>
</table>

The resulting report would contain information on each employee's department and job class.

Example -- Join three tables

List job information on all employees who earn the minimum salary for their job class. The SELECT statement joins three tables: EMP, JOBLIST, and JOBCLASS. CA OLQ retrieves all rows where:

- The employee ID in the EMP table matches the employee ID assigned to a job in the JOBLIST table, and
- The employee's job class matches the class assigned in the JOBCLASS table, and
- The employee's salary equals the minimum salary in the JOBCLASS table
Joining a table to itself

Why join a table to itself?

You join a table to itself when one column in a table requires the table itself to supply additional information. For example, table EMP has a column of manager IDs. To find the name of Michael Angelo's manager, you find the manager's ID in the MGRID column and then find the same ID in the EMPID column. The manager's name is associated with the employee ID:

EMP table

<table>
<thead>
<tr>
<th>EMPID</th>
<th>EMPNAME</th>
<th>MGRID</th>
</tr>
</thead>
<tbody>
<tr>
<td>0075</td>
<td>Jennifer Lanzarotta</td>
<td>0003</td>
</tr>
<tr>
<td>3302</td>
<td>Bart Elopoulos</td>
<td>0004</td>
</tr>
<tr>
<td>3871</td>
<td>Reginald Mahoney</td>
<td>0007</td>
</tr>
<tr>
<td>4230</td>
<td>Duc Ho</td>
<td>0011</td>
</tr>
<tr>
<td>6264</td>
<td>Daniel Ortega</td>
<td>0013</td>
</tr>
<tr>
<td>6348</td>
<td>Edith Jones</td>
<td>0015</td>
</tr>
<tr>
<td>7170</td>
<td>Anthea Poznanski</td>
<td>0075</td>
</tr>
<tr>
<td>8939</td>
<td>Ankur Sahu</td>
<td>0321</td>
</tr>
<tr>
<td>8957</td>
<td>William Sternbach</td>
<td>0349</td>
</tr>
</tbody>
</table>

Join table EMP to itself by equating Anthea Poznanski's manager ID to an employee ID

<table>
<thead>
<tr>
<th>EMPID</th>
<th>EMPNAME</th>
<th>MGRID</th>
<th>EMPID</th>
<th>EMPNAME</th>
<th>MGRID</th>
</tr>
</thead>
<tbody>
<tr>
<td>7170</td>
<td>Anthea Poznanski</td>
<td>0075</td>
<td>0075</td>
<td>Jennifer Lanzarotta</td>
<td>0003</td>
</tr>
</tbody>
</table>

This type of join is called a reflexive join and is used to implement a nested structure, which is also called a bill-of-materials structure. A nested structure is one where there is a relationship between columns in the same table. For example, nested relationships exist:

- In an industrial environment where a part in a component of another part and can contain component parts itself. For example, a door is a component of a car and contains these component parts: handle, lock, and window.
In the corporate environment used for examples in this section where an employee manages other employees. Likewise, an employee can report to more than one supervisor.

How to join a table to itself

To join a table to itself, you simply treat the table as two tables by assigning aliases, or alternative names, to the table in the SELECT statement. Thereafter, the SELECT statement coding requirements listed below are the same as if you were joining two different tables. This figure illustrates how to join table EMP to itself by assigning two alias table names -- MANAGER and WORKER:

### EMP table

<table>
<thead>
<tr>
<th>EMPID</th>
<th>EMPNAME</th>
<th>MGRID</th>
</tr>
</thead>
<tbody>
<tr>
<td>0075</td>
<td>Jennifer Lanzarotta</td>
<td>0003</td>
</tr>
<tr>
<td>3302</td>
<td>Bart Elopoulos</td>
<td>0004</td>
</tr>
<tr>
<td>3871</td>
<td>Reginald Mahoney</td>
<td>0007</td>
</tr>
<tr>
<td>4242</td>
<td>Chinua Achebe</td>
<td>0075</td>
</tr>
<tr>
<td>7170</td>
<td>Anthea Poznanski</td>
<td>0075</td>
</tr>
</tbody>
</table>

Assign alias table names

### MANAGER table

<table>
<thead>
<tr>
<th>EMPID</th>
<th>EMPNAME</th>
<th>MGRID</th>
</tr>
</thead>
<tbody>
<tr>
<td>0075</td>
<td>Jennifer Lanzarotta</td>
<td>0003</td>
</tr>
<tr>
<td>3302</td>
<td>Bart Elopoulos</td>
<td>0004</td>
</tr>
<tr>
<td>3871</td>
<td>Reginald Mahoney</td>
<td>0007</td>
</tr>
<tr>
<td>4242</td>
<td>Chinua Achebe</td>
<td>0075</td>
</tr>
<tr>
<td>7170</td>
<td>Anthea Poznanski</td>
<td>0075</td>
</tr>
</tbody>
</table>

### WORKER table

Join the MANAGER and WORKER tables by equating the ID of the worker's manager to the employee ID of the manager

<table>
<thead>
<tr>
<th>EMPID</th>
<th>EMPNAME</th>
<th>MGRID</th>
</tr>
</thead>
<tbody>
<tr>
<td>0075</td>
<td>Jennifer Lanzarotta</td>
<td>0003</td>
</tr>
<tr>
<td>3302</td>
<td>Bart Elopoulos</td>
<td>0004</td>
</tr>
<tr>
<td>3871</td>
<td>Reginald Mahoney</td>
<td>0007</td>
</tr>
<tr>
<td>4242</td>
<td>Chinua Achebe</td>
<td>0075</td>
</tr>
<tr>
<td>7170</td>
<td>Anthea Poznanski</td>
<td>0075</td>
</tr>
</tbody>
</table>

### Coding the SELECT statement

To join a table to itself, follow these steps:

1. Qualify each column listed after the SELECT keyword with an alias table name: `manager.lastname`
2. For each reflexive join (that is, for each time you join a table to itself), assign an alias by coding:

   a. The table name
   b. A blank
   c. The alias

   Separate each table and its alias from another with a comma: emp manager, emp worker

3. In the WHERE clause, compare two columns that share the same type of information:
   manager.empid = worker.mgrid

Example

List each manager and associated staff. To retrieve this information, join the EMP table to itself, equating a manager’s employee ID to the ID of a staff member’s manager.

The SELECT statement assigns these aliases to EMP table: MANAGER and WORKER. The WHERE clause selects rows where the employee ID in the MANAGER table equals the manager’s ID in the WORKER table. The columns display the manager’s name retrieved from the MANAGER table and the worker’s name retrieved from the WORKER table:

```sql
select manager.lastname as supervisor, worker.lastname as staff
from emp manager, emp worker
where manager.empid = worker.mgrid
```

<table>
<thead>
<tr>
<th>SUPERVISOR</th>
<th>STAFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>GARFIELD</td>
<td>JENSEN</td>
</tr>
<tr>
<td>GARFIELD</td>
<td>JACOBI</td>
</tr>
<tr>
<td>GARFIELD</td>
<td>TYRO</td>
</tr>
<tr>
<td>GARFIELD</td>
<td>DOUGHERTZ</td>
</tr>
<tr>
<td>GARFIELD</td>
<td>GRANGER</td>
</tr>
<tr>
<td>GARFIELD</td>
<td>GALWAY</td>
</tr>
<tr>
<td>GARFIELD</td>
<td>LITERATA</td>
</tr>
<tr>
<td>BANK</td>
<td>ANGELO</td>
</tr>
<tr>
<td>BANK</td>
<td>MCDUGALL</td>
</tr>
<tr>
<td>BANK</td>
<td>PENMAN</td>
</tr>
<tr>
<td>BANK</td>
<td>JACKSON</td>
</tr>
<tr>
<td>HENDON</td>
<td>RUPEE</td>
</tr>
<tr>
<td>HENDON</td>
<td>GARFIELD</td>
</tr>
<tr>
<td>HENDON</td>
<td>BANK</td>
</tr>
</tbody>
</table>

Comparing a column to more than one value

Why you compare more than one value

You may want to compare values in one table to a list of values in another table. For example, you might want to obtain information on all employees whose ID matches the manager IDs in table DEPT. If you know the manager IDs in table DEPT, you could code a SELECT statement that compares employee IDs to a list of manager IDs:
select empid, lastname from emp where empid in (0013, 0011, 0003, 0004, 0007, 0015, 0349, 0321, 0030)

However, this type of query isn't practical in some cases, especially for large tables. CA OLQ provides a method to retrieve this information.

**Coding it as a join operation**

You can obtain the same information by joining the EMP and DEPT tables where the manager ID in table DEPT is the same as the employee ID in table EMP:

```
select emp.lastname from dept, emp where emp.empid = dept.mgrid
```

## Retrieving Information From CA IDMS/DB Records

### Contents
- Retrieving data from a single record (see page 633)
- Retrieving data from two or more records (see page 634)
- Retrieving data from a record joined to itself (see page 636)

Examples to this point are all based on ASF-generated tables. However, you can also use the SELECT statement to retrieve data from CA IDMS/DB database records. This portion of the section tells you how to retrieve data from:

- One CA IDMS/DB record
- Two or more CA IDMS/DB records by using sets
- A CA IDMS/DB bill-of-materials data structure

### Retrieving data from a single record

**Comparing a record to a table**

Retrieving information from a CA IDMS/DB database record is comparable to retrieving rows from a table. The CA IDMS/DB record type (for example, the EMPLOYEE record) is like a table (for example, the EMP table). The EMPLOYEE record occurrences are like the rows of EMP table.

**Signing on to a subschema**

To access CA IDMS/DB database records, you have to first sign on to a subschema. A subschema is a view of the database. That is, it describes a subset of the database records that a CA IDMS/DB database contains.

To signon to a subschema, issue a `SIGNON` command:

```
signon ss=empss01
```

**Coding the SELECT statement**

To retrieve information from a single CA IDMS/DB database record, code the SELECT statement like this:
1. Enter record field names instead of column names following the SELECT keyword: emp-last-name-0415

2. Enter the record name instead of the table name following the FROM keyword: employee

Finding record names

If you do not know the record or the record field names that you need, issue these commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HELP RECORDS</td>
<td>Lists the records defined to the subschema you have accessed</td>
</tr>
<tr>
<td>HELP RECORD=record-name</td>
<td>Lists the fields associated with the named record</td>
</tr>
</tbody>
</table>

Example

List each office's code number, in ascending order, and city:

```sql
select office-code-0450, office-city-0450 from office order by office-code-0450 !
display
```

OFFICE REPORT
01/27/99

<table>
<thead>
<tr>
<th>OFFICE-CODE-0450</th>
<th>OFFICE-CITY-0450</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>005</td>
<td>GLASSTER</td>
</tr>
<tr>
<td>008</td>
<td>WESTON</td>
</tr>
<tr>
<td>012</td>
<td>CAMBRIDGE</td>
</tr>
</tbody>
</table>

END OF REPORT

Retrieving data from two or more records

How CA OLQ interprets a set relationship

CA IDMS/DB database records relate to each other through set&$. RB. relationships or through data values. A CA IDMS/DB set links occurrences of one record type with associated occurrences of another record type. One record type is the owner of the set. The other record type is a member of the set. For example, the OFFICE-EMPLOYEE set associates each employee with a particular office. The OFFICE record is the owner and the EMPLOYEE record is the member.

When you retrieve data from two or more tables, you join the tables on a common value. For example, you join the DEPT and EMP tables by equating department IDs: where dept.deptid=emp.deptid. Similarly, when you retrieve data from two or more records, you join the records by using a set relationship. For example, you join the OFFICE and EMPLOYEE records with the OFFICE-EMPLOYEE set relationship: where office-employee. The figure below illustrates how CA OLQ interprets a set relationship between the OFFICE and EMPLOYEE database records:
IDMSDB--Retrieving data from two or more records

Coding the SELECT statement

To retrieve data from two or more CA IDMS/DB records, code the SELECT statement like this:

1. Enter record field names instead of column names following the SELECT keyword. If the same field name appears in more than one record, qualify the fields with the record name: `employee.emp-id-0415`.

2. Enter the record names instead of the table names following the FROM keyword: `office, employee`.

3. Enter the set names following the WHERE keyword. Separate set names by AND. You can also include other WHERE criteria. Separate additional WHERE criteria from set names by using AND, also: `where (office-employee and dept-employee) and (dept-id-0410 = 4000)`.

Example 1 -- Retrieving data from 2 records

List all employees who work in the Boston office. The SELECT statement shown below selects EMPLOYEE and OFFICE records in the OFFICE-EMPLOYEE set having an office code of BOSTON:

```
select emp-last-name-0415 as &xq.employee name', office-city-0450 as &xq.office' from employee, office where office-employee and office-city-0450 = &xq.boston' ! display
```
Example 2 -- Retrieving data from 3 records

List the department and office location of each employee. The SELECT statement shown below:

1. Selects fields from the DEPARTMENT, EMPLOYEE, and OFFICE records where:
   - The DEPT-EMPLOYEE set associates DEPARTMENT and EMPLOYEE record occurrences
   - The OFFICE-EMPLOYEE set associates OFFICE and EMPLOYEE record occurrences

2. Orders the retrieved rows by employee name within each department

```
select dept-id-0410 as department, emp-last-name-0415 as employee, office-city-0450 as office
from department, employee, office
where dept-employee and office-employee
order by dept-id-0410, emp-last-name-0415
```

Retrieving data from a record joined to itself

By using a nested structure
Like tables, records can participate in nested structures. For example, employees who are supervisors have employees who are staff members. Likewise, employees who are staff can report to more than one supervisor.

This type of set relationship is called a bill-of-materials structure. The data structure diagram in Sample Tables and Database shows a bill-of-materials structure between the EMPLOYEE and STRUCTURE records:

- One set is MANAGES. It associates supervisors with staff.
- The other set is REPORTS-TO. It associates each employee with one or more supervisors.

The STRUCTURE record exists only to facilitate these set relationships.

Assign alias record names

This figure shows how CA OLQ interprets a bill-of-materials structure relationally, by using alias names for the EMPLOYEE record:

- The SUPERVISOR alias contains occurrences of supervisors. The MANAGES set relates each supervisor to employees who are staff.
- The WORKER alias contains occurrences of staff. The REPORTS-TO set relates each staff member to employees who are supervisors.

CA OLQ uses the concept illustrated in the figure shown under Retrieving data from two or more records (see page 634) to interpret these set relationships:

```
EMPLOYEE

MANAGES     REPORTS TO

STRUCTURE

SUPERVISOR      WORKER

MANAGES     REPORTS TO

EMPLOYEE
```

CA OLQ relates the supervisor IDs to worker record occurrences and relates worker worker IDs to manager record occurrences in two tables.
Employee 0075 manages Poznanski and Achebe
Employee 0075 works for Sarem and Romans

Coding the SELECT statement

As with a reflexive table join, the SELECT statement is unique in that you assign alias names to the same record. To code the SELECT statement, follow these steps:

1. Qualify each record field name with an alias record name: `supervisor.emp-last-name-0415`.

2. For each bill-of-materials navigation, assign a unique alias to the record by coding:
   a. The record name
   b. A blank
   c. The alias
   Separate each record and its alias from another with a comma: `employee supervisor, employee worker, structure`.

3. Name the sets that participate in the bill-of-materials following the WHERE clause by coding:
   a. A qualified set name. A qualified set name is the set name followed by a period and the alias record name: `where reports-to.worker`.
   b. An AND logical operator.
   c. A second qualified set name.

Example -- Listing managers and their staff

Retrieve each project leader and the staff working on the project. The SELECT statement assigns these aliases to the EMPLOYEE record: SUPERVISOR and WORKER. STRUCTURE is the name of the CA IDMS/DB record that facilitates this bill-of-materials data structure.

The alias table names qualify record field names that appear following the SELECT keyword and set names that appear in the WHERE clause:

```
select supervisor.emp-last-name-0415 as &xq.project leader', worker.emp-last-name-0415 as &xq.staff' from employee supervisor, employee worker, structure where manages.
```

EMPLOYEE/EMPLOYEE REPORT

<table>
<thead>
<tr>
<th>PROJECT LEADER</th>
<th>STAFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>BANK</td>
<td>ZEDI</td>
</tr>
</tbody>
</table>

EMPLOYEE/EMPLOYEE REPORT
Retrieving Data From Tables and Records

Signing on

To join information from a table and a database record, sign on to the subschema that contains the database record and sign on to the table. Use a view id in the SIGNON statement to keep each signon active and to qualify records and tables that share the same name:

```
signon table emp view emptab !signon ss=empss01 view empssc&RB.
```

Comparing view IDs to alias names

A view ID applies when you sign on to more than one subschema. It qualifies records or tables that have the same name in different subschemas.

An alias applies to records and tables in the SELECT statement. It qualifies fields (columns) and sets that have the same name in different records or tables.

Coding the SELECT statement

Once you have signed on to the subschemas, code the SELECT statement like this:

1. Enter column names and/or record field names following the SELECT keyword: lastname, dept-name-0410.

2. Enter table and/or record names following the FROM keyword. If the record and table share the same name, qualify them with the view id you assigned to the subschema at signon: emptab.emp, empssc.department.

3. Equate values in the WHERE clause. If you are retrieving information from a table and a database record, compare a column to a field: deptid = dept-id-0410

Example -- Joining a table and a record

Join table EMP and database record DEPARTMENT to list all employees and their associated departments by:

1. Signing on to the EMPSS01 subschema in dictionary TSTDICT.

2. Signing on to the EMP table in dictionary ASFDICT.
3. Select the employee's name from table EMP and the department's name from record DEPARTMENT where a table row and record occurrence have the same department ID value. The ORDER BY clause instructs CA OLQ to display the rows alphabetically by employees names.

```sql
signon ss empss01 dictname tstdict view empssc
   !
select lastname as employee, dept-name-0410 as department
from emp, department where deptid = dept-id-0410 order by lastname display
```

<table>
<thead>
<tr>
<th>EMPLOYEE</th>
<th>DEPARTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANGELO</td>
<td>PUBLIC RELATIONS</td>
</tr>
<tr>
<td>BANK</td>
<td>PUBLIC RELATIONS</td>
</tr>
<tr>
<td>DOUGH</td>
<td>INTERNAL SOFTWARE</td>
</tr>
<tr>
<td>GALLWAY</td>
<td>INTERNAL SOFTWARE</td>
</tr>
<tr>
<td>GARFIELD</td>
<td>INTERNAL SOFTWARE</td>
</tr>
<tr>
<td>GRANGER</td>
<td>INTERNAL SOFTWARE</td>
</tr>
<tr>
<td>HEAROWITZ</td>
<td>INTERNAL SOFTWARE</td>
</tr>
<tr>
<td>HENDON</td>
<td>EXECUTIVE ADMINISTRATION</td>
</tr>
<tr>
<td>JACKSON</td>
<td>PUBLIC RELATIONS</td>
</tr>
<tr>
<td>JACOBI</td>
<td>INTERNAL SOFTWARE</td>
</tr>
<tr>
<td>JENSEN</td>
<td>INTERNAL SOFTWARE</td>
</tr>
<tr>
<td>LITERATA</td>
<td>INTERNAL SOFTWARE</td>
</tr>
<tr>
<td>MCDougALL</td>
<td>PUBLIC RELATIONS</td>
</tr>
<tr>
<td>PAPAZEUS</td>
<td>EXECUTIVE ADMINISTRATION</td>
</tr>
</tbody>
</table>

**Test Your Knowledge**

Using the sample tables in Appendix A, code a SELECT statement for each of the queries listed below. num=B.Answers, contains one possible answer. Remember, there can be more than one way to achieve the same result when you use the SELECT statement.

1. List the average salary for employees in the Boston office
2. List all employees in the Springfield office who are programmer/analysts
3. List the average salary of the managers in table DEPT
4. List all employees who are either programmer/analysts, paste-up artists, or a brainstorming manager

Using the data structure diagram in code a SELECT statement for these database record queries:

1. List each employee's job title and salary
2. For each job assigned to more than one employee, list the number of employees assigned to the job and their average salaries
3. List each employee's manager
Sample Tables and Database

**BOSTON Table**

<table>
<thead>
<tr>
<th>EMPID</th>
<th>LASTNAME</th>
<th>HIREDATE</th>
<th>OFFICECODE</th>
<th>TOWN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0120</td>
<td>ANGELO</td>
<td>090879</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0007</td>
<td>BANK</td>
<td>043078</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0069</td>
<td>BLOOMER</td>
<td>050580</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0119</td>
<td>BOWER</td>
<td>121477</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0081</td>
<td>FITZHugh</td>
<td>091981</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0045</td>
<td>FONRAD</td>
<td>041480</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0053</td>
<td>GARDNER</td>
<td>061581</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0030</td>
<td>HENDON</td>
<td>112173</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0100</td>
<td>HUTTON</td>
<td>090777</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0158</td>
<td>JACKSON</td>
<td>070777</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0011</td>
<td>JENSON</td>
<td>092980</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0051</td>
<td>JOHNSON</td>
<td>032377</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0049</td>
<td>KAHALLY</td>
<td>092979</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0067</td>
<td>KIMBALL</td>
<td>091978</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0106</td>
<td>KING</td>
<td>081680</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0074</td>
<td>KRAAMER</td>
<td>040481</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0127</td>
<td>MCDougall</td>
<td>060780</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0101</td>
<td>NICEMAN</td>
<td>050680</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0091</td>
<td>ORGRATZI</td>
<td>101080</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0149</td>
<td>PENMAN</td>
<td>090877</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0013</td>
<td>PEOPLES</td>
<td>012801</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0048</td>
<td>TERNER</td>
<td>052682</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0466</td>
<td>ANDALE</td>
<td>061582</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0457</td>
<td>ARM</td>
<td>012365</td>
<td>002</td>
<td>BOSTON</td>
</tr>
</tbody>
</table>

**DEPT Table**

<table>
<thead>
<tr>
<th>DEPTID</th>
<th>DEPTNAME</th>
<th>MGRID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>PERSONNEL</td>
<td>0013</td>
</tr>
<tr>
<td>2000</td>
<td>ACCOUNTING AND PAYROLL</td>
<td>0011</td>
</tr>
<tr>
<td>3100</td>
<td>INTERNAL SOFTWARE</td>
<td>0003</td>
</tr>
<tr>
<td>3200</td>
<td>COMPUTER OPERATIONS</td>
<td>0004</td>
</tr>
<tr>
<td>4000</td>
<td>PUBLIC RELATIONS</td>
<td>0007</td>
</tr>
<tr>
<td>5100</td>
<td>BRAINSTORMING</td>
<td>0015</td>
</tr>
<tr>
<td>5200</td>
<td>THERMOREGULATION</td>
<td>0349</td>
</tr>
<tr>
<td>5300</td>
<td>BLUE SKIES</td>
<td>0321</td>
</tr>
<tr>
<td>6666</td>
<td>EXECUTIVE ADMINISTRATION</td>
<td>0030</td>
</tr>
</tbody>
</table>

**EMP Table**

<table>
<thead>
<tr>
<th>EMPID</th>
<th>FIRSTNAME</th>
<th>LASTNAME</th>
<th>DEPTID</th>
<th>MGRID</th>
<th>SALARY</th>
<th>JOBID</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>JOHN</td>
<td>RUPEE</td>
<td>6666</td>
<td>0030</td>
<td>76000.00</td>
<td>3001</td>
</tr>
<tr>
<td>0003</td>
<td>Jennifer</td>
<td>Garfield</td>
<td>3100</td>
<td>0030</td>
<td>65000.00</td>
<td>3001</td>
</tr>
<tr>
<td>0007</td>
<td>Monte</td>
<td>Bank</td>
<td>4000</td>
<td>0030</td>
<td>80000.00</td>
<td>4001</td>
</tr>
<tr>
<td>0019</td>
<td>Julie</td>
<td>Jensen</td>
<td>3100</td>
<td>0003</td>
<td>37000.00</td>
<td>3025</td>
</tr>
<tr>
<td>0020</td>
<td>James</td>
<td>Jacobi</td>
<td>3100</td>
<td>0003</td>
<td>55000.00</td>
<td>3011</td>
</tr>
<tr>
<td>0021</td>
<td>Ralph</td>
<td>Tyro</td>
<td>3100</td>
<td>0003</td>
<td>20000.00</td>
<td>3027</td>
</tr>
<tr>
<td>0024</td>
<td>Jane</td>
<td>Dough</td>
<td>3100</td>
<td>0003</td>
<td>33000.00</td>
<td>3025</td>
</tr>
<tr>
<td>0027</td>
<td>Vladimir</td>
<td>Hearowitz</td>
<td>3100</td>
<td>0003</td>
<td>33000.00</td>
<td>3025</td>
</tr>
<tr>
<td>EMPID</td>
<td>PHONE</td>
<td>SSN</td>
<td>STARTDATE</td>
<td>BIRTHDATE</td>
<td>CLASS</td>
<td>SEX</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>-----</td>
<td>-----------</td>
<td>-----------</td>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td>0001</td>
<td>6173421515</td>
<td>013445656</td>
<td>750223</td>
<td>450818</td>
<td>72</td>
<td>M</td>
</tr>
<tr>
<td>0003</td>
<td>6173321967</td>
<td>021994516</td>
<td>770121</td>
<td>450818</td>
<td>71</td>
<td>F</td>
</tr>
<tr>
<td>0007</td>
<td>6173321933</td>
<td>022446676</td>
<td>780430</td>
<td>500101</td>
<td>72</td>
<td>M</td>
</tr>
<tr>
<td>0019</td>
<td>61732241955</td>
<td>033456718</td>
<td>820929</td>
<td>400730</td>
<td>43</td>
<td>F</td>
</tr>
<tr>
<td>0020</td>
<td>6173348912</td>
<td>018813465</td>
<td>811111</td>
<td>401101</td>
<td>63</td>
<td>M</td>
</tr>
<tr>
<td>0021</td>
<td>6174459191</td>
<td>019893456</td>
<td>801221</td>
<td>551225</td>
<td>21</td>
<td>M</td>
</tr>
<tr>
<td>0024</td>
<td>6174458155</td>
<td>022337878</td>
<td>760808</td>
<td>530229</td>
<td>42</td>
<td>M</td>
</tr>
<tr>
<td>0025</td>
<td>6173349634</td>
<td>031896154</td>
<td>810909</td>
<td>560425</td>
<td>42</td>
<td>F</td>
</tr>
<tr>
<td>0026</td>
<td>6173341212</td>
<td>011234545</td>
<td>800527</td>
<td>580222</td>
<td>42</td>
<td>M</td>
</tr>
<tr>
<td>0027</td>
<td>6173349155</td>
<td>014567777</td>
<td>811010</td>
<td>571006</td>
<td>42</td>
<td>M</td>
</tr>
<tr>
<td>0030</td>
<td>6173346767</td>
<td>019996919</td>
<td>801707</td>
<td>520430</td>
<td>42</td>
<td>M</td>
</tr>
<tr>
<td>0035</td>
<td>6174561277</td>
<td>022887770</td>
<td>780907</td>
<td>350304</td>
<td>72</td>
<td>N</td>
</tr>
<tr>
<td>0036</td>
<td>6174319909</td>
<td>010004560</td>
<td>790223</td>
<td>401229</td>
<td>33</td>
<td>F</td>
</tr>
</tbody>
</table>

**JOBCLASS Table**

<table>
<thead>
<tr>
<th>CLASS</th>
<th>MINSALARY</th>
<th>MAXSALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>12000.00</td>
<td>18000.00</td>
</tr>
<tr>
<td>21</td>
<td>18000.00</td>
<td>20000.00</td>
</tr>
<tr>
<td>33</td>
<td>37000.00</td>
<td>39000.00</td>
</tr>
<tr>
<td>42</td>
<td>33000.00</td>
<td>35000.00</td>
</tr>
<tr>
<td>43</td>
<td>35500.00</td>
<td>40000.00</td>
</tr>
<tr>
<td>51</td>
<td>38000.00</td>
<td>57000.00</td>
</tr>
<tr>
<td>63</td>
<td>52500.00</td>
<td>62500.00</td>
</tr>
<tr>
<td>71</td>
<td>60000.00</td>
<td>70000.00</td>
</tr>
<tr>
<td>72</td>
<td>76000.00</td>
<td>90000.00</td>
</tr>
<tr>
<td>81</td>
<td>85000.00</td>
<td>105000.00</td>
</tr>
<tr>
<td>93</td>
<td>200000.00</td>
<td>250000.00</td>
</tr>
</tbody>
</table>

**JOBLIST Table**

<table>
<thead>
<tr>
<th>JOBID</th>
<th>TITLE</th>
<th>EMPID</th>
<th>CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>MGR PERSONNEL</td>
<td>0013</td>
<td>72</td>
</tr>
<tr>
<td>1023</td>
<td>RECRUITER/INTERVWR</td>
<td>0091</td>
<td>43</td>
</tr>
<tr>
<td>1051</td>
<td>PERSONNEL CLERK</td>
<td>0051</td>
<td>11</td>
</tr>
<tr>
<td>1053</td>
<td>PERSONNEL CLERK</td>
<td>0081</td>
<td>11</td>
</tr>
<tr>
<td>2001</td>
<td>MGR ACCTNG/PAYROLL</td>
<td>0011</td>
<td>72</td>
</tr>
<tr>
<td>2023</td>
<td>ACCOUNTANT</td>
<td>0067</td>
<td>62</td>
</tr>
<tr>
<td>2025</td>
<td>FINANCIAL ANALYST</td>
<td>0100</td>
<td>62</td>
</tr>
<tr>
<td>2051</td>
<td>AP CLERK</td>
<td>0101</td>
<td>12</td>
</tr>
<tr>
<td>2053</td>
<td>AR CLERK</td>
<td>0106</td>
<td>12</td>
</tr>
<tr>
<td>2055</td>
<td>PAYROLL CLERK</td>
<td>0069</td>
<td>13</td>
</tr>
</tbody>
</table>
### SPRINGFIELD Table

<table>
<thead>
<tr>
<th>EMPID</th>
<th>LASTNAME</th>
<th>STARTYEAR</th>
<th>OFFICECODE</th>
<th>CITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0004</td>
<td>CRANE</td>
<td>051477</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0024</td>
<td>DOUGH</td>
<td>080876</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0032</td>
<td>FERNDALE</td>
<td>090979</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0329</td>
<td>FINN</td>
<td>061679</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0029</td>
<td>GALLWAY</td>
<td>101081</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0003</td>
<td>GARFIELD</td>
<td>012177</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0028</td>
<td>GRANGER</td>
<td>052780</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0027</td>
<td>HEARWITZ</td>
<td>090981</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0020</td>
<td>JACOBI</td>
<td>111181</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0019</td>
<td>JENSEN</td>
<td>092982</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0016</td>
<td>KLWELLEN</td>
<td>010678</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0031</td>
<td>LIPSCH</td>
<td>042981</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0035</td>
<td>LITERATA</td>
<td>090980</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0023</td>
<td>O’HEARN</td>
<td>050478</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>EMPID</td>
<td>LASTNAME</td>
<td>STARTDATE</td>
<td>OFFICECODE</td>
<td>CITY</td>
</tr>
<tr>
<td>-------</td>
<td>-----------</td>
<td>-----------</td>
<td>------------</td>
<td>--------</td>
</tr>
<tr>
<td>0466</td>
<td>ANDALE</td>
<td>061582</td>
<td>008</td>
<td>WESTON</td>
</tr>
<tr>
<td>0457</td>
<td>ARM</td>
<td>012365</td>
<td>008</td>
<td>WESTON</td>
</tr>
<tr>
<td>0467</td>
<td>BREEZE</td>
<td>022980</td>
<td>008</td>
<td>WESTON</td>
</tr>
<tr>
<td>0469</td>
<td>KASPAR</td>
<td>091476</td>
<td>008</td>
<td>WESTON</td>
</tr>
<tr>
<td>0341</td>
<td>MUNYON</td>
<td>031482</td>
<td>008</td>
<td>WESTON</td>
</tr>
<tr>
<td>0471</td>
<td>PAPAZEUS</td>
<td>090778</td>
<td>008</td>
<td>WESTON</td>
</tr>
<tr>
<td>0355</td>
<td>TIME</td>
<td>050675</td>
<td>008</td>
<td>WESTON</td>
</tr>
<tr>
<td>0458</td>
<td>WAGNER</td>
<td>070883</td>
<td>008</td>
<td>WESTON</td>
</tr>
<tr>
<td>0349</td>
<td>WILCO</td>
<td>111179</td>
<td>008</td>
<td>WESTON</td>
</tr>
<tr>
<td>0119</td>
<td>BOWER</td>
<td>121477</td>
<td>008</td>
<td>WESTON</td>
</tr>
<tr>
<td>0081</td>
<td>FITZHUGH</td>
<td>091981</td>
<td>008</td>
<td>WESTON</td>
</tr>
<tr>
<td>0004</td>
<td>CRANE</td>
<td>051477</td>
<td>008</td>
<td>WESTON</td>
</tr>
<tr>
<td>0024</td>
<td>DOUGH</td>
<td>088876</td>
<td>008</td>
<td>WESTON</td>
</tr>
<tr>
<td>0032</td>
<td>FERNDALE</td>
<td>090979</td>
<td>008</td>
<td>WESTON</td>
</tr>
<tr>
<td>0329</td>
<td>FINN</td>
<td>061679</td>
<td>008</td>
<td>WESTON</td>
</tr>
</tbody>
</table>
EMPLOYEE Database Data Structure Diagram

Answers

Chapter 2 Answers

Query 1

List the salary range for each job class:

```sql
select class, maxsalary - minsalary as &xq.salary range' from jobclass
```

<table>
<thead>
<tr>
<th>CLASS</th>
<th>SALARY RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>6000.00</td>
</tr>
<tr>
<td>21</td>
<td>2000.00</td>
</tr>
</tbody>
</table>
Query 2

List all employees who have both a:

- Manager with ID 0007 or 0003
- Job that begins with the number 3

```
select lastname, mgrid, jobid from emp where mgrid in (0003, 0007) and jobid like '&xq.3___'
```

Query 3

Identify all employees whose sex code was entered incorrectly (that is, is not M or F):

```
select firstname, lastname, sex from emp where sex not in ('m', 'f')
```

Query 4

List the number of years employees have from January 1, 1988 until they reach retirement age (65):

```
select firstname, lastname, birthdate, (65 - next-int-eqlo(datedif(880101,birthdate)/365)) as 'years until retirement'
from emp
```
Chapter 3 Answers

Query 1

For each manager in table EMP, determine the number of staff reporting to the manager and the average salary of the staff members:

```
select mgrid, count(*) as &number of staff', avg(salary) as &average salary'
from emp
```

```
group by mgrid
```

```
EMP REPORT
mm/dd/yy
```

```
MGRID NUMBER OF STAFF AVERAGE SALARY
0003 8 35375.00
0007 5 29200.00
0030 6 106833.33
```

END OF REPORT

Query 2

List the number of jobs greater than or equal to 3 that are assigned to classes 10 and 50. Display the report in order by the number of jobs:

```
select count(*) as &number of jobs', class from joblist
```

```
where class between 10 and 50
```

```
group by class
```

```
having count(*) >= 3
```

```
order by 1
```

```
JOBLIST REPORT
mm/dd/yy
```

```
NUMBER OF JOBS CLASS
3 11
3 33
4 12
5 42
6 21
6 43
```

END OF REPORT

Query 3

For departments 6666, 4000, and 3100, list the minimum and maximum salaries within the department, provided the average departmental salary in greater than $37,800:

```
select deptid, min(salary) as &minimum salary', max(salary) as &maximum salary'
from emp
```

```
where deptid in (3100, 4000, 6666)
```

```
group by deptid having avg(salary) > 37800
```

```
EMP REPORT
mm/dd/yy
```

- 1 -
Query 4

List information on employees earning less than the average salary:

```
select lastname, salary from emp where salary < (select avg(salary) from emp)
```

<table>
<thead>
<tr>
<th>LASTNAME</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>JENSEN</td>
<td>37000.00</td>
</tr>
<tr>
<td>JACOBI</td>
<td>55000.00</td>
</tr>
<tr>
<td>TYRO</td>
<td>20000.00</td>
</tr>
<tr>
<td>DOUGH</td>
<td>33000.00</td>
</tr>
<tr>
<td>HEAROWITZ</td>
<td>33000.00</td>
</tr>
<tr>
<td>GRANGER</td>
<td>34500.00</td>
</tr>
<tr>
<td>GALLWAY</td>
<td>33000.00</td>
</tr>
<tr>
<td>LITERATA</td>
<td>37500.00</td>
</tr>
<tr>
<td>ANGELO</td>
<td>18000.00</td>
</tr>
<tr>
<td>MCDougall</td>
<td>18000.00</td>
</tr>
<tr>
<td>PENMAN</td>
<td>39000.00</td>
</tr>
<tr>
<td>JACKSON</td>
<td>34000.00</td>
</tr>
<tr>
<td>ZEDI</td>
<td>37000.00</td>
</tr>
</tbody>
</table>

Chapter 4 Answers

Table Query 1

List the average salary for employees in the Boston office:

```
select count(*) as &xq.number of boston employees', avg(salary) as &xq.average salary'
from emp, boston where emp.empid = boston.empid
```

<table>
<thead>
<tr>
<th>NUMBER OF BOSTON EMPLOYEES</th>
<th>AVERAGE SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>74142.85</td>
</tr>
</tbody>
</table>

Table Query 2

List all employees in the Springfield office who are programmer/analysts:

```
select emp.lastname, joblist.title, springfield.city from emp, joblist, springfield
where emp.empid = springfield.empid and emp.empid = joblist.empid and joblist.title = &xq.programmer/anaylst'
```

<table>
<thead>
<tr>
<th>LASTNAME</th>
<th>TITLE</th>
<th>CITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOUGH</td>
<td>PROGRAMMER/ANALYST</td>
<td>SPRINGFIELD</td>
</tr>
</tbody>
</table>
Table Query 3

List the average salary of the managers in table DEPT:

\[
\text{select avg(salary) as 'average manager salary' from emp, dept where emp.empid = dept.mgrid}
\]

EMP/DEPARTMENT REPORT

AVERAGE MANAGER SALARY

128333.33

END OF REPORT

Table Query 4

List all employees who are either programmer/analysts, paste-up artists, or a brainstorming manager:

\[
\text{select emp.lastname, joblist.title from emp, joblist where joblist.title in ('programmer/analyst', 'paste-up artist', 'mgr brainstorming') and joblist.empid = emp.empid}
\]

JOBLIST/EMPLOYEE REPORT

LASTNAME           TITLE

DOUGH             PROGRAMMER/ANALYST
HEAROWITZ         PROGRAMMER/ANALYST
GRANGER           PROGRAMMER/ANALYST
GALLWAY           PROGRAMMER/ANALYST
ANGELO            PASTE-UP ARTIST
MCDOUGALL         PASTE-UP ARTIST
PAPAZEUS          MGR BRAINSTORMING

END OF REPORT

Record Query 1

List each employee's job title and salary:

\[
\text{select employee.emp-last-name-0415, job.job-title-0440, emposition.salary-amount-0420 from employee, emposition, job where employee.emp-emposition and job-emposition}
\]

JOB/EMPOSITION/EMPLOYEE REPORT

EMP-LAST-NAME-0415 TITLE-0440 SALARY-AMOUNT-0420

O’HEARN  PROGRAMMER/ANALYST  38000.00
TYRO    PROGRAMMER/ANALYST  28000.00
WILCO   MGR THERMOREGULATION  80000.00
GARFIELD MGR INTERNL SOFTWARE  65000.00
RUPEE   MGR INTERNL SOFTWARE  76000.00
JOHNSON PERSONNEL CLERK  13500.00
FITZMUG PERSONNEL CLERK  13000.00
BLOOMER PAYROLL CLERK  15000.00
ARM    STURM/DRANG ADMIN  46000.00
WAGNER STURM/DRANG ADMIN  47000.00
ORGRATZI RECRUITER/INTERVWR  39000.00
BANK    MGR PUBLIC RELATIONS  80000.00
Record Query 2

For each job assigned to more than one employee, list the number of employees assigned to the job and their average salaries:

```
select job.title-0440, count(*) as &xq.number of jobs', avg(emposition.salary-amount-0420) as &xq.average salary' from employee, emposition, job
where emp-emposition and
job-emposition
group by job.title-0440
having count(*) > 1
```

```
JOB/EMPOSITION REPORT

mm/dd/yy

TITLE-0440  NUMBER OF JOBS  AVERAGE SALARY

COMPUTER OPERATOR  3  20333.33
DATA ENTRY CLERK  4  19937.50
DATABASE ADMIN.  2  55000.00
DOCUMENTATION SPEC  2  41250.00
MGR BRAINSTORMING  3  83333.33
MGR COMPUTER OPS  3  68333.33
MGR INTERNL SOFTWARE  2  70500.00
PASTE-UP ARTIST  2  17500.00
PERSONNEL CLERK  2  12500.00
PR WRITER  2  38000.00
PROGRAMMER TRAINEE  2  29000.00
PROGRAMMER/ANALYST  6  35500.00
RAINMAKER  3  46166.66
```

Record Query 3

List each employee's manager:

```
select worker.last-name-0415 as &xq.staff', supervisor.last-name-0415 as &xq.project leader' from employee worker, employee supervisor, structure
where reports-to.worker
and manages.supervisor
order by worker.emp-last-name-0415
```

```
EMPLOYEE/EMPLOYEE REPORT

mm/dd/yy

STAFF  PROJECT LEADER

ANDALE  MAKER
ANGELO  BANK
ARM  MAKER
BANK  BOWER
BANK  WILDER
BLOOMER  JENSON
BOWER  BANK
BREEZE  MAKER
CLOTH  WILCO
CLOUD  WILDER
CLOUD  MOON
CRANE  RUPEE
CROW  MAKER
```

- 1 -
Chapter 5 Answers

Query 1

List the jobs for which employees earn more than $65,000:

```
select * from joblist where exists (select * from emp where emp.jobid=joblist.jobid and salary > 65000)
```

```
JOBLIST REPORT
   mm/dd/yy
EMPID   CLASS   JOBID   TITLE
  0003     71     3001  MGR INTERNL SOFTWARE
  0007     72     4001  MGR PUBLIC RELATIONS
  0015     72     5001  MGR BRAINSTORMING
  0471     72     5001  MGR BRAINSTORMING
  0030     93     9001  PRESIDENT
  0472     81     9005  DIR CORP CONFUSION

END OF REPORT
```

Query 2

List openings for jobs that command salaries in the range $35,000 to $40,000:

```
select class, jobid, title from joblist
where not exists (select * from emp where emp.jobid=joblist.jobid)
and exists (select * from jobclass where jobclass.class=joblist.class
and minsalary >= 35000 and maxsalary <= 40000)
```

```
JOBLIST REPORT
   mm/dd/yy
CLASS   JOBID   TITLE
   43     1023  RECRUITER/INTERVWR
   33     4023  PHOTOGRAPHER
   43     5023  RAINDANCE CONSULTANT
   43     5035  HUMIDITY CONTROL CLK

END OF REPORT
```

Query 3

List all employees in the Springfield office who have this job title: "Programmer/Analyst":

```
select * from springfield
where exists (select * from employee where employee.empid = springfield.empid
and exists (select * from joblist
where employee.empid = joblist.empid
and joblist.title = "Programmer/Analyst"))
```

```
SPRINGFIELD REPORT
   mm/dd/yy
EMPID   LASTNAME   STARTYEAR   OFFICECODE   CITY
  0024   DOUGH     080876      001         SPRINGFIELD
  0029   GALLWAY   101081      001         SPRINGFIELD
  0028   GRANGER   092780      001         SPRINGFIELD
  0027   HEAROWITZ 090981      001         SPRINGFIELD

END OF REPORT
```

Query 4

List all departments that have no associated employee:
select * from dept where not exists (select * from emp where emp.deptid = dept.deptid)

DEPT REPORT

<table>
<thead>
<tr>
<th>DEPTID</th>
<th>DEPTNAME</th>
<th>MGRID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>PERSONNEL</td>
<td>0013</td>
</tr>
<tr>
<td>2000</td>
<td>ACCOUNTING AND PAYROLL</td>
<td>0011</td>
</tr>
<tr>
<td>3200</td>
<td>COMPUTER OPERATIONS</td>
<td>0004</td>
</tr>
<tr>
<td>5100</td>
<td>BRAINSTORMING</td>
<td>0015</td>
</tr>
<tr>
<td>5200</td>
<td>THERMOREGULATION</td>
<td>0349</td>
</tr>
<tr>
<td>5300</td>
<td>BLUE SKIES</td>
<td>0321</td>
</tr>
</tbody>
</table>

END OF REPORT

Chapter 6 Answers

Query 1

List employees in the Boston and Springfield offices, including commuters. Display the report in order of last name:

select * from boston
union
select * from springfield
order by 2

BOSTON REPORT

<table>
<thead>
<tr>
<th>EMPID</th>
<th>LASTNAME</th>
<th>HIREDATE</th>
<th>OFFICECODE</th>
<th>TOWN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0466</td>
<td>ANDALE</td>
<td>061582</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0120</td>
<td>ANGELO</td>
<td>090879</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0457</td>
<td>ARM</td>
<td>012365</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0007</td>
<td>BANK</td>
<td>043078</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0007</td>
<td>BANK</td>
<td>043078</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0069</td>
<td>BLOOMER</td>
<td>050580</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0069</td>
<td>BLOOMER</td>
<td>050580</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0119</td>
<td>BOWER</td>
<td>121477</td>
<td>002</td>
<td>BOSTON</td>
</tr>
<tr>
<td>0467</td>
<td>BREEZE</td>
<td>022980</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0004</td>
<td>CRANE</td>
<td>051477</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0024</td>
<td>DOUGH</td>
<td>080876</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0032</td>
<td>FERNDALE</td>
<td>090979</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>0329</td>
<td>FINN</td>
<td>061679</td>
<td>001</td>
<td>SPRINGFIELD</td>
</tr>
</tbody>
</table>

Query 2

List the average salaries of employees in the Boston and Springfield offices:

select avg(salary) as 'AVERAGE SALARY', town as CITY from boston, emp
where boston.empid = emp.empid group by town
union
select avg(salary), city from springfield, emp
where springfield.empid = emp.empid group by city

BOSTON/EMP REPORT

<table>
<thead>
<tr>
<th>AVERAGE SALARY</th>
<th>CITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>49500.00</td>
<td>SPRINGFIELD</td>
</tr>
<tr>
<td>74142.85</td>
<td>BOSTON</td>
</tr>
</tbody>
</table>

Query 3
List all the employees in the Springfield and Weston offices whose jobid is 3001 or 5001:

```
select lastname, city from weston, emp where weston.empid=emp.empid and jobid in (3001, 5001)
union
select lastname, city from springfield, emp where springfield.empid=emp.empid and jobid in (3001, 5001)
```

```
EMP/WESTON REPORT

LASTNAME          | JOBID | CITY
-----------------|-------|------
GARFIELD          | 3001  | SPRINGFIELD
PAPAZEUS          | 5001  | WESTON
RUPEE             | 3001  | SPRINGFIELD

END OF REPORT
```

Final Query

```
Contents

- Glossary (see page 653)
```

For each job, list the number of employees greater than 1 who each the minimum salary for the job's class. Display the report in order of job title:

```
select joblist.title, count(*) as number of employees, avg(salary) as salary
from joblist, emp where exists (select * from jobclass where emp.class=jobclass.class and emp.salary = jobclass.minsalary) and joblist.empid=emp.empid group by title having count(*) > 1 order by title
```

```
JOBLIST/EMP REPORT

TITLE               | NUMBER OF EMPLOYEES | SALARY
-------------------|---------------------|-------
PASTE-UP ARTIST    | 2                   | 18000.00
PROGRAMMER/ANALYST | 3                   | 33000.00

END OF REPORT
```

Glossary

Additional selection criteria

Logical expressions, logical record keywords, and criteria expressions for subscripted fields that you use to tell CA OLQ which rows of data to retrieve for your report. Additional selection criteria are specified in the Additional selection criteria field of the Selection criteria screen.

Aggregate function

A function that performs a predefined operation on a group of report rows. Examples of aggregate functions are: average, high value, low value, count, and total.

ASF (Automatic System Facility)

A tool in CA-IDMS/DB used to create and manage tables. Once you have created a table using CA OLQ, you can use ASF to modify the table definition.

ASF dictionary
An alternate data dictionary used by ASF. You must be using the ASF dictionary when you are creating data tables.

ASF table

A presentation of data as a series of rows and columns from a table associated with the IDMSR schema.

Batch

Batch processing means that the user doesn’t have to interact with the computer system in order to perform a function. Usually, a batch job is set up in advance (such as when you fill out your Batch Processing screen). Once the job has started running, you cannot intervene except to suspend execution.

Built-in function

A function that performs a predefined string, arithmetic, date/time, or trigonometric calculation on your report rows. Examples of built-in functions are: substring, Gregorian date, cosine, and square root.

Code table

A table defined in the data dictionary that contains corresponding pairs of values. One column in the table lists internal code values that are used to efficiently store the data in the database. The other column in the table lists external values that are used in programs or reports.

For example, a STATE code table could represent the STATE column as 01 in the internal (encoded) expression and Alabama in the external (decoded) expression.

Column

A vertical division in a table. A column represents a category of information. For example, employee last name.

Column alignment

An editing feature that determines how report data columns align under the column headers. Options are left, right, and center.

Column header

A header at the top of each column of report data.

COMPUTE statement

A CA OLQ syntax statement used to perform calculations in the menu facility. Any time you specify a built-in or aggregate function, CA OLQ creates a COMPUTE statement. You can also provide your own COMPUTE statement.

Current report
The report you’re working on in an active CA OLQ session. If you retrieve a saved report, CA OLQ clears out the current report.

**Data dictionary**

The storage facility used by CA products as a central source for data definitions, modules, and runtime information. Qfile definitions are stored in the data dictionary.

**Database view**

Another term for subschema.

**Destination**

When you print a report, you specify an output destination where the report is to be printed. Usually, the destination is a file associated with a printer.

**Display sequence**

A numeric sequence listed on CA OLQ report formatting screens. The numbers in the display sequence correspond to the order in which report columns are displayed.

**Element**

The smallest significant unit of data in a CA-IDMS/DB database. Record elements correspond to columns in a table. For example, DEPT-ID-0410 is an element.

**External picture**

A code that defines the way your column value is formatted in your report. The external picture is used to add punctuation (for example, commas) and special characters (for example, dollar signs) to your column data display.

**Group by all**

A report total including all rows in your report. Group by all means the same thing as report total.

**Group computation**

A calculation that CA OLQ performs on a report group.

**Group field**

A report column whose value is used to divide your report rows into groups. For example, you could list all of the company’s employees grouped according to which department they work in. In this case, DEPARTMENT NAME is the group field.

**Input file**

A file that contains input into a batch program.

**Integrated Data Dictionary (IDD)**
The CA product used to access definitions stored in the data dictionary.

Interactive

A way of performing a function in which the computer system requires the user to provide input and then responds to that input. An example would be CA OLQ menu mode. Another term to describe interactive processing is online.

Interrupt count

The maximum number of records CA OLQ will retrieve when building a report. If the number of records that meet the selection criteria for that report exceeds the interrupt count, CA OLQ suspends data retrieval and issues a message asking you if you want to continue to retrieve records.

Job control language

A language used to define the special requirements of your batch program to the system. Job control language (JCL) statements name input and output files, the name of your program, and your output destination.

Join

A relational operation through which two or more tables are combined. Tables are joined based on columns that the tables have in common.

Join criteria

A logical expression that compares like columns in two or more tables.

Output destination

Any type of device that receives the report that you have created as a result of your batch job. Output destinations can be a printer, a terminal, or a log.

Output file

A file that contains the results of your batch program.

Page header

A title at the top of each page of your report.

Page footer

A title at the bottom of each page of your report.

Project

A relational operation through which only particular columns of a table are accessed.

Qfile definition

The CA OLQ syntax statements stored in the data dictionary when you create your qfile.
Record

A group of related elements. Records correspond to rows in a table. For example, DEPT-NAME-0410, DEPT-ID-0410, and DEPT-HEAD-ID-0410 are all grouped into the DEPARTMENT record.

Report group

A set of report rows such that each row contains the same value of the group field. For example, the personnel department is a report group with DEPARTMENT NAME as the group field. Each row in this group contains Personnel in the DEPARTMENT NAME field.

Report subtotal

A computation applied to a report group. For example, if you grouped your employees by department, you could create report subtotals that compute the average salary in each department.

Report total

A computation that includes all of the rows in your report. For example, you could compute the total sales revenue earned by all of your employees. Note that a report total does not have to be a sum. It can also be an average, a counter, a high value, or a low value.

Row

A horizontal row in a table. A row represents one data occurrence. For example, information on each employee.

Retention period

The number of days your saved report file is kept in your directory. After the retention period has expired, the report file is deleted.

Saved report

A report file maintained in your user directory that contains a copy of a current CA OLQ report.

Select

A relational operation through which only particular rows of a table are accessed.

Selection criteria

Logical expressions that you use to tell CA OLQ which rows of data to retrieve for your report. Selection criteria are specified in the Selection criteria field of the Column Select screen.

Separator character

A character used to separate group computations from the rest of the report.

Sort

A way to order report rows. CA OLQ sorts the rows in your report based on the value of the sort field that you specify. Rows can be sorted in ascending or descending order.
Sparse/Full option

A CA OLQ editing feature that determines how column values that repeat in consecutive rows are displayed:

- **Sparse** displays only the first of a repeating set of column values.
- **Full** displays all occurrences of the repeating value.

SQL table

A presentation of data as a series of rows and columns from a table associated with an SQL schema.

Subschema

A view of the database that contains a subset of the records, elements, sets, and areas that make up the entire database. A subschema usually views data that is functionally related.

For example, the personnel department uses a subschema that views employee information such as salary, date of hire, and personal information. All of the information is held in the same database, but the personnel department views only the information that it needs.

Summary computation

Another term for group computation.

Summary line

A report line that displays a group computation.
Other CA IDMS Reporting Facilities

Online reporting facilities

Users can display information online with the following tools:

- CA OLQ (see page 659)
- IDMS DC/UCF OnLine PLOG (OLP) (see page 659)
- DC/UCF DCMT DISPLAY commands (see page 660)
- IDMS DC/UCF OPER WATCH commands (see page 660)
- IDD DDDL DISPLAY commands (see page 661)
- CA IDMS Performance Monitor (see page 661)

Each item is discussed separately as follows.

**CA OLQ**

Generate reports on dictionary data

CA OLQ can be used to generate reports on information contained in a DC/UCF dictionary. The following screen shows an example of a CA OLQ-generated report:

```
CURRENT REPORT
MAP-NAME-098 PREP-BY-098 PROG-NAME-051 DATE-CREATED-051
EMPMAP MAPR10.0 EMPINQ mm/dd/yy
JKDMAP1 JKD D6 mm/dd/yy
EMPJOB DEH EMP08BD mm/dd/yy
JKDTEST1 JKD JKDDIAL1 mm/dd/yy
JKDTEST1 JKD JKDDIAL mm/dd/yy
JKDMAP JKD D4 mm/dd/yy
JKDMAP JKD D2 mm/dd/yy
END OF REPORT - PAGE 1
```

CA OLQ reports supplement information supplied by dictionary (DREPORT), IDMS DC/UCF (CREPORT), and CA ADS (AREPORT) reports.

⚠️ **Note:** For more information on CA OLQ, see the *CA OLQ Reference section*.

**IDMS DC/UCF OnLine PLOG (OLP)**

Displays current contents of the log area
OLP displays the current contents of the DDLDCLOG area of the dictionary. OLP can be used to display system messages, system trace information, and snap dumps.

The following screen shows information displayed by OLP.

```
FROM   ON    TO    ON    COL PRT SKIP LOG TYPES           ROLL STATUS
hh:mm:ss yyyy-mm-dd hh:mm:ss yyyy-mm-dd 010 OFF 2345 (WT/TR/DU/   ) 040
hh:mm   IDMS DC258003 Vnn USER KYM SIGNED ON LTERM LT12002 AT hh:mm:ss.ht yy.ddd
hh:mm   IDMS DC402009 Vnn DICTNAME CORPDICT HAS BEEN SET
hh:mm   IDMS DC258003 Vnn USER BXP SIGNED ON LTERM LT12003 AT hh:mm:ss.ht yy.ddd
hh:mm   IDMS DC402009 Vnn DICTNAME MIDADICT HAS BEEN SET
hh:mm   IDMS DC258003 Vnn USER ALP SIGNED ON LTERM LT12004 AT hh:mm.ss.ht yy.ddd
hh:mm   IDMS DC402009 Vnn DICTNAME CORPDICT HAS BEEN SET
hh:mm   IDMS DC402009 Vnn DICTNAME MIDADICT HAS BEEN SET
hh:mm   IDMS DC258003 Vnn USER RPMP SIGNED OFF LTERM LT12002 AT hh:mm:ss.ht yy.ddd
hh:mm   IDMS DC402009 Vnn DICTNAME CORECNT HAS BEEN SET
hh:mm   IDMS DC258003 Vnn USER GAF SIGNED ON LTERM LT12005 AT hh:mm.ss.ht yy.ddd
hh:mm   IDMS DC402009 Vnn DICTNAME CORPDICT HAS BEEN SET
```

Note: For more information on this reporting tool, see the Administrating section.

DC/UCF DCMT DISPLAY commands

Display definitions and runtime statistics

DC/UCF DCMT DISPLAY commands display definitions and runtime statistics associated with DC/UCF systems.

DCMT commands complement information supplied by IDMS DC/UCF system (CREPORT) and statistics (SREPORT) reports.

Note: For more information on DCMT commands, see the CA IDMS System Tasks and Operator Reference section.

IDMS DC/UCF OPER WATCH commands

Display dynamic system runtime statistics

IDMS DC/UCF OPER WATCH commands display dynamic system runtime statistics associated with IDMS DC/UCF systems.

OPER WATCH commands complement information supplied by IDMS DC/UCF system (CREPORT) and statistics (SREPORT) reports.
Note: For more information on OPER WATCH commands, see the CA IDMS System Tasks and Operator Reference section.

### IDD DDDL DISPLAY commands

**Display entity occurrences defined to dictionary**

DDDL can be used in command mode or menu mode to display entity occurrences defined to the data dictionary. The following screen shows the contents displayed by submitting the following command to IDD:

```
DISPLAY MAP NAME IS SYBMAP AS SYNTAX.
```

```
ADD
  MAP NAME IS SYBMAP VERSION IS 1
  WITHIN PANEL SYBMAP-OLMPANEL VERSION IS 1
    DATE CREATED IS mm/dd/yy
    DATE LAST UPDATED IS mm/dd/yy
    PREPARED BY SYB
    REVISED BY SYB
    PUBLIC ACCESS IS ALLOWED FOR ALL
    DATE LAST COMPILED IS mm/dd/yy
    TIME LAST COMPILED IS mm/dd/yy
    MAP RELEASE I.D. IS R2
    NUMBER OF NON LITERAL FIELDS IS 1
    NUMBER OF RECORDS IN MAP IS 1
    PFLD NAME TO SET CURSOR AT IS OLMPF-0003
    ASSOCIATED WITH RECORD SYBREC VERSION 1

```

Information display using DDDL in command mode or menu mode complements information supplied by data dictionary (DREPORT), IDMS DC/UCF system (CREPORT), and CA ADS (AREPORT) reports.

Note: For more information on this reporting tool, see the CA IDMS IDD DDDL Reference section.

### CA IDMS Performance Monitor

**Display system statistics**

The CA IDMS Performance Monitor is a performance and tuning tool you can use to monitor hardware and software utilization in a DC/UCF system.

**Three components**

The CA IDMS Performance Monitor consists of three components:
- Realtime monitor -- Displays specific system-resource statistics at the time of the request
- Interval monitor -- Displays system-wide, wait-time statistics for a unit of time
- Application monitor -- Displays statistics about resource usage by individual program

⚠️ **Note:** For more information on the online component of CA IDMS Performance Monitor, see the *CA IDMS Performance Monitor Using section*.

### Batch reporting facilities

You can display information with the following batch reporting tools:

- CA IDMS/DB utilities
- CA IDMS Performance Monitor reports

The utilities are described below. A brief overview of the CA IDMS Performance Monitor appears earlier in this section. For more information on the batch component of the Performance Monitor, see the *CA IDMS Performance Monitor System Administering section*.

#### Summary of utilities reports

**Utilities** are batch facilities that perform system services and frequently generate report output. The table below lists CA IDMS-supplied utilities that supplement CA IDMS reports.

<table>
<thead>
<tr>
<th>Utility</th>
<th>Information in Report Output</th>
</tr>
</thead>
</table>
| ADSORPTS(1) | CA ADS dialog and application statistics:  
Dialog reports -- summary dialog information, process module source statements, dialog-associated records, contents of fixed dialog block (FDB)  
Application reports -- application task codes, global records, functions, responses |
| ADSOBPLG    | Runtime CA ADS Batch application and dialog statistics (also called CA ADS Batch Print Log Utility in the *CA ADS Batch Using section.*) |
| IDMSDBAN(2) | Database structure statistics:  
Area statistics  
Set statistics  
Record statistics  
Set analysis data for four types of sets |
| IDMSRPTS(2) | Data dictionary definitions for schemas, subschemas, physical databases, and nonschema created entities:  
Schema reports -- areas, files, subschemas, record types, and sets defined in a specific schema  
Subschema reports -- record types, sets, areas, logical records, including paths and program activity, in a specified subschema  
Schema-independent reports -- module, IDD-built record, protocol, user, and q-file |
<table>
<thead>
<tr>
<th>Utility</th>
<th>Information in Report Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>definitions</td>
<td></td>
</tr>
<tr>
<td>Physical database definition reports -- segment and DMCL listings; physical database definitions</td>
<td></td>
</tr>
<tr>
<td>PRINT INDEX(2)</td>
<td>Structure of system-owned indexes and indexed sets</td>
</tr>
<tr>
<td>PRINT JOURNAL (2)</td>
<td>Journal file statistics</td>
</tr>
<tr>
<td></td>
<td>Status of disk journal files</td>
</tr>
<tr>
<td></td>
<td>Journal page utilization</td>
</tr>
<tr>
<td>PRINT LOG(2)</td>
<td>System statistics logged to DDLDCLOG area of the data dictionary</td>
</tr>
<tr>
<td>PRINT PAGE(2)</td>
<td>Database page content in hexadecimal/decimal format</td>
</tr>
<tr>
<td>PRINT SPACE(2)</td>
<td>Space utilization in one or more areas or segments</td>
</tr>
<tr>
<td>RHDCMPUT</td>
<td>Definition, screen image, and source code of map and panel occurrences (also called the Mapping Batch Utility in the IDMS DC/UCF Mapping Facility)</td>
</tr>
<tr>
<td>ROLLBACK and ROLLFORWARD (2)</td>
<td>Journal record statistics:</td>
</tr>
<tr>
<td></td>
<td>Checkpoints and ENDJ checkpoint statistics for each transaction</td>
</tr>
<tr>
<td></td>
<td>Before and after database record images in hexadecimal/decimal format</td>
</tr>
</tbody>
</table>
Standard Dictionary Reports -- DREPORTS

Standard dictionary reports provide summary, detail, and cross-reference information on the contents of the dictionary, including information on certain entity relationships.

Dictionary reports can do the following:

- Help the MIS department organize and control data by providing an up-to-date source of information on each entity in the dictionary and the relationships between entities
- Help the programmer by providing information such as the names and synonyms of records and their associated elements; task codes and their associated programs; file retention periods; and standardized routines
- Help the DCA manage the communication network by providing information such as the relationship of lines and terminals, the names of programs executed by a task, and the names of users authorized to initiate a task
- Help the systems analyst monitor system performance by providing information on how the existing systems function, whether the systems are being used as originally designed, and the effect a new system could have on an existing system.

For more information, see the following topics:
- Information Obtained from the DDLDML Area (see page 664)
- Dictionary Report Categories (see page 665)
- Types of Reports (see page 665)
- Producing Dictionary Reports (see page 666)
- Basic Entity Reports (see page 669)
- Site-Specific Entity Reports (see page 700)
- Teleprocessing Entity Reports (see page 705)
- Cross-Reference Reports (see page 726)
- Special-Purpose Report Modules (see page 740)

Information Obtained from the DDLDML Area

Standard dictionary reports document information maintained in the DDLDML area of the dictionary. The records in the DDLDML area of the dictionary are defined and modified by various CA IDMS system software components, such as the Data Dictionary Definition Language (DDDL) compiler and the system generation compiler. Information that is displayed in the fields of the dictionary reports is taken from the corresponding fields in the dictionary records. Note, however, that not all fields in the records are displayed in the standard reports.
Dictionary Report Categories

Dictionary reports are grouped into the following categories for the purpose of discussion:

- **Basic entity reports** document the standard data processing entities: system, user, program, module, file, record, and element.

- **Site-specific entity reports** document entities that are defined to meet the needs of a specific site: class, attribute, and user-defined entities. The system-supplied classes, LANGUAGE and MODE, are included in this category.

- **Teleprocessing entity reports** document entities typical of online systems: destination, line, logical terminal, map, panel, physical terminal, queue, and task.

- **Cross-reference reports** document relationships between specific entities.

- **Special-purpose reports** are used to perform administrative functions, such as punching module source to an output file.

Types of Reports

Three types of reports are typically available for each of the basic entity and teleprocessing entity reports:

<table>
<thead>
<tr>
<th>Type</th>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum</td>
<td>053 - 068</td>
<td>List all occurrences of a particular entity type with a minimal amount of supplementary information on each occurrence, such as the version number, description, and the dates the occurrence was defined and updated.</td>
</tr>
<tr>
<td>Deta</td>
<td>001 - 019</td>
<td>List every occurrence of a particular entity type with in-depth descriptions of the entity and its relationships with other dictionary entities. You can also run a detail report with a SELECT or BYPASS parameter to report on entities that meet a specified test condition. For example, instead of running a detail report that lists all modules, you could use a SELECT parameter to report on only those entities that were created by a particular user and updated on a particular date.</td>
</tr>
<tr>
<td>Key</td>
<td></td>
<td>Are identical to the detail reports in format but list only the entity occurrences specified as key values in the KEY parameter. More than one KEY parameter can be specified for a particular report; each KEY parameter must be specified on a separate line. The syntax for the KEY parameter is described later in this section. The key field names used in the KEY parameter are presented in Key Field Names for Key Reports table.</td>
</tr>
</tbody>
</table>
Producing Dictionary Reports

Contents
- Syntax (see page 666)
- Parameters (see page 666)
- Key Field Names for Key Reports (see page 667)
- Examples (see page 668)

A dictionary report is produced by submitting a job that includes the standard Job Control Language (JCL) for CA Culprit for CA IDMS report writers and report-specific control statements. JCL for z/OS, z/VSE and z/VM operating systems is shown in Appendixes A through D, respectively.

Syntax

Col 2

```
DATABASE DICTNAME=dictionary-name
DBNAME=database-name
DICTNODE=dictionary-node-name
DBNODE=database-node-name
```

Col 2

```
PARAM= LIST
     NOLIST
     EJECT
```

Col 1

dREPORT= dreport-number

Col 2

```
KEY key-field-name 'key-field-value'
```

Col 2

```
SESelect BUFFER
     BYPass record-name level-number IN PATH path-id
```

WHEN boolean-expression

Parameters

General syntax rules for the CA IDMS reports are presented in Section1:. Syntax rules specific to the DREPORTs are described below:

- **dREPORT dreport-number**

  Identifies the report to be run. DREPORT identifies the report as a dictionary report; the D of DREPORT is optional. If D is specified, DREPORT must begin in column 1; if D is not specified, REPORT must begin in column 2.
Dictionary report numbers are shown in DREPORT Listing table and in Tables 2-2 through 2-6 in this section. Multiple reports can be requested in the same job run with the following exceptions:

- KEY reports cannot be requested in the same run with detail, summary, cross-reference, or special-purpose reports.
- DREPORTs 009 and 010 cannot be run together.

- **KEY**
  Identifies the key report being requested. KEY must be entered in columns 2-4.

- **key-field-name**
  Identifies the key field name of the entity type being reported on. Valid key field names for each entity type are listed in DREPORT Key fields table.

- **'key-value'**
  Identifies the entity occurrence of the type specified in key-field-name; key-value must be enclosed in single quotation marks. If the quoted value is smaller than the field length shown in Key Field Names for Key Reports table, CA Culprit for CA IDMS pads the value with spaces on the right; if the value is shorter, CA Culprit for CA IDMS truncates the value to the specified length.

### Key Field Names for Key Reports

<table>
<thead>
<tr>
<th>DREPORT Module</th>
<th>DREPORT Name</th>
<th>Name of Key Field</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>038</td>
<td>Attribute/Record Report</td>
<td>ATTR-NAME-093</td>
<td>40</td>
</tr>
<tr>
<td>039</td>
<td>Attribute/Element Report</td>
<td>ATTR-NAME-093</td>
<td>40</td>
</tr>
<tr>
<td>051</td>
<td>Module Text to Card Utility</td>
<td>MOD-NAME-067</td>
<td>32</td>
</tr>
<tr>
<td>052</td>
<td>Module Text to File Utility</td>
<td>MOD-NAME-067</td>
<td>32</td>
</tr>
<tr>
<td>071</td>
<td>Class Report</td>
<td>CLASS-NAME-092</td>
<td>20</td>
</tr>
<tr>
<td>072</td>
<td>Attribute Report</td>
<td>ATTR-NAME-093</td>
<td>40</td>
</tr>
<tr>
<td>073</td>
<td>System Report</td>
<td>SYS-NAME-041</td>
<td>8</td>
</tr>
<tr>
<td>074</td>
<td>User Report</td>
<td>USER-NAME-047</td>
<td>32</td>
</tr>
<tr>
<td>075</td>
<td>Program Report</td>
<td>PROG-NAME-051</td>
<td>8</td>
</tr>
<tr>
<td>076</td>
<td>Module Report</td>
<td>MOD-NAME-067</td>
<td>32</td>
</tr>
<tr>
<td>077</td>
<td>File Report</td>
<td>SA-NAM-018</td>
<td>32</td>
</tr>
<tr>
<td>078</td>
<td>Record Report</td>
<td>SR-NAM-036</td>
<td>32</td>
</tr>
<tr>
<td>079</td>
<td>Element Report</td>
<td>INQ-NAM-058</td>
<td>32</td>
</tr>
<tr>
<td>081</td>
<td>Task Report</td>
<td>TASK-NAME-025</td>
<td>8</td>
</tr>
<tr>
<td>082</td>
<td>Queue Report</td>
<td>QUEUE-NAME-030</td>
<td>16</td>
</tr>
<tr>
<td>083</td>
<td>Destination Report</td>
<td>DEST-NAME-028</td>
<td>8</td>
</tr>
<tr>
<td>084</td>
<td>Logical Terminal Report</td>
<td>LTRM-NAME-106</td>
<td>8</td>
</tr>
<tr>
<td>085</td>
<td>Physical Terminal Report</td>
<td>PTRM-NAME-074</td>
<td>8</td>
</tr>
</tbody>
</table>
**Examples**

**Example 1**

These control statements can be used to produce a detailed report on the file occurrence ORDER TRANSACTIONS. The report modules used to run the report are found in the default dictionary; data for the report is taken from the DOCUNWK dictionary.

```plaintext
DATABASE DBNAME=DOCUNWK
DREPORT=077
  KEY SA-NAM-018 'ORDER TRANSACTIONS'
```

**Example 2**

These control statements can be used to produce a file summary report (DREPORT 057) with a parameter listing and a module detail and file detail report without a listing. The report modules used to run the report are in the default dictionary; data for the reports is taken from the DOCUTEST dictionary.

```plaintext
DATABASE DBNAME=DOCUTEST
PARAM=LIST
DREPORT=57
  PARAM=NOLIST
  DREPORT=6,7
```

**Example 3**

These control statements can be used to produce a class report that lists information on the LANGUAGE and DIVISION class occurrences. The key field name for the class report is CLASS-NAME-092; the key values for the report are LANGUAGE and DIVISION. The report modules used to run the report and the data for the report are in the default dictionary.

```plaintext
DREPORT=071
  KEY CLASS-NAME-092 'LANGUAGE'
  KEY CLASS-NAME-092 'DIVISION'
```

**Example 4**

These control statements can be used to produce a module report that lists the source code for dictionary report modules 1, 2, and 3. The report modules used to run the reports and the data for the reports are in the CULPDICT dictionary.

```plaintext
DATABASE DICTNAME=CULPDICT DBNAME=CULPDICT
DREPORT=076
  KEY MOD-NAME-067 'DREPORT 001'
  KEY MOD-NAME-067 'DREPORT 002'
  KEY MOD-NAME-067 'DREPORT 003'
```

**Example 5**

---

<table>
<thead>
<tr>
<th>DREPORT Module</th>
<th>DREPORT Name</th>
<th>Name of Key Field</th>
<th>Field Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>086</td>
<td>Line Report</td>
<td>LINE-NAME-109</td>
<td>8</td>
</tr>
<tr>
<td>087</td>
<td>Panel Report</td>
<td>PANEL-NAME-118</td>
<td>32</td>
</tr>
<tr>
<td>088</td>
<td>Map Report</td>
<td>MAP-NAME-098</td>
<td>8</td>
</tr>
<tr>
<td>089</td>
<td>User-Defined Entity Report</td>
<td>CLASS-NAME-092</td>
<td>20</td>
</tr>
</tbody>
</table>
These control statements can be used to produce a module detail report that describes only those modules created by user CMH and updated on a specified date. The report modules required to run the report are found in the CULPDIRCT dictionary; data for the report is taken from the default dictionary.

```
DATABASE DICTNAME=CULPDIRCT
DREPORT=006
SELECT MOD-NAME-067 WHEN PREP-BY-067 EQ 'CMH' AND
  *DATE-LU-067 EQ 'mm/dd/yy'
```

What Follows

The remainder of this section describes the standard dictionary reports. The reports are presented by category and entity type. For example, the MODULE summary, MODULE detail, and MODULE key reports are presented together under Basic Entity Reports (see page 669). Tables 2-2 through 2-6 list the dictionary reports by category; DREPORT Listing table in Appendix D lists the DREPORTs by number.

## Basic Entity Reports

### Contents

- **System Reports** *(DREPORTs 003, 053, 073)* (see page 670)
  - System Summary Report *(DREPORT 053)* (see page 670)
  - System Detail and Key Reports *(DREPORTs 003 and 073)* (see page 671)
- **User Reports** *(DREPORTs 004, 054, 074)* (see page 673)
  - User Summary Report *(DREPORT 054)* (see page 673)
  - User Detail and Key Reports *(DREPORTs 004 and 074)* (see page 674)
- **Program Reports** *(DREPORTs 005, 055, 075)* (see page 678)
  - Program Summary Report *(DREPORT 055)* (see page 678)
  - Program Detail and Key Reports *(DREPORTs 005 and 075)* (see page 679)
- **Module Reports** *(DREPORTs 006, 056, 076)* (see page 682)
  - Module Summary Report *(DREPORT 056)* (see page 682)
  - Module Detail and Key Reports *(DREPORTs 006 and 076)* (see page 683)
- **File Reports** *(DREPORTs 007, 057, 077)* (see page 686)
  - File Summary Report *(DREPORT 057)* (see page 686)
  - File Detail and Key Reports *(DREPORTs 007 and 077)* (see page 686)
- **Record Reports** *(DREPORTs 008, 038, 058, 078)* (see page 689)
  - Record Summary Report *(DREPORT 058)* (see page 689)
  - Record Detail and Key Report *(DREPORTs 008 and 078)* (see page 690)
  - Record/Attribute Key Report *(DREPORT 038)* (see page 693)
- **Element Reports** *(DREPORTs 009, 010, 039, 059, 079)* (see page 694)
  - Element Summary Report *(DREPORT 059)* (see page 694)
  - Element Detail and Key Reports *(DREPORTs 009 and 079)* (see page 695)
  - Inactive Element Detail Report *(DREPORT 010)* (see page 698)
  - Attribute/Element Key Report *(DREPORT 039)* (see page 699)
The basic entity reports provide information on seven entity types: system, user, program, module, file, record, and element. The following Basic Entity Reports table lists the basic entity reports in order of presentation in this section.

<table>
<thead>
<tr>
<th>DREPORT Module</th>
<th>DREPORT Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>003</td>
<td>System Report -- Detail</td>
</tr>
<tr>
<td>053</td>
<td>System Report -- Summary</td>
</tr>
<tr>
<td>073</td>
<td>System Report -- Key(1)</td>
</tr>
<tr>
<td>004</td>
<td>User Report -- Detail</td>
</tr>
<tr>
<td>054</td>
<td>User Report -- Summary</td>
</tr>
<tr>
<td>074</td>
<td>User Report -- Key(1)</td>
</tr>
<tr>
<td>005</td>
<td>Program Report -- Detail</td>
</tr>
<tr>
<td>055</td>
<td>Program Report -- Summary</td>
</tr>
<tr>
<td>075</td>
<td>Program Report -- Key(1)</td>
</tr>
<tr>
<td>006</td>
<td>Module Report -- Detail</td>
</tr>
<tr>
<td>056</td>
<td>Module Report -- Summary</td>
</tr>
<tr>
<td>076</td>
<td>Module Report -- Key(1)</td>
</tr>
<tr>
<td>007</td>
<td>File Report -- Detail(3)</td>
</tr>
<tr>
<td>057</td>
<td>File Report -- Summary(3)</td>
</tr>
<tr>
<td>077</td>
<td>File Report -- Key(1)</td>
</tr>
<tr>
<td>008</td>
<td>Record Report -- Detail</td>
</tr>
<tr>
<td>038</td>
<td>Attribute/Record Report -- Key(1)</td>
</tr>
<tr>
<td>058</td>
<td>Record Report -- Summary</td>
</tr>
<tr>
<td>078</td>
<td>Record Report -- Key(1)</td>
</tr>
<tr>
<td>009</td>
<td>Element Report -- Detail(2)</td>
</tr>
<tr>
<td>010</td>
<td>Inactive Element Report -- Detail(2)</td>
</tr>
<tr>
<td>039</td>
<td>Attribute/Element Report -- Summary</td>
</tr>
<tr>
<td>059</td>
<td>Element Report -- Summary</td>
</tr>
<tr>
<td>079</td>
<td>Element Report -- Key(1)</td>
</tr>
</tbody>
</table>

Notes:

- KEY reports cannot be run with summary or detail reports.
- DREPORTs 009 and 010 cannot be requested in the same run.
- DREPORTs 007 and 057 cannot be run with any other reports.

**System Reports (DREPORTs 003, 053, 073)**

System reports (DREPORTs 003, 053, and 073) provide information on system occurrences that have been defined to the dictionary. Systems are represented in the dictionary as occurrences of the SYS041 record.

**System Summary Report (DREPORT 053)**

Contents
The system summary report provides the following information on systems occurrences:

- System name and version number
- Subsystems associated with the system (indented under the system name
- Associated description
- Date the system occurrence was added and date updated (if updated)

Sample DREPORT 053:

```
REPORT NO. 53 mm/dd/yy PAGE 1
DREPORT 053 SYSTEM REPORT - SUMMARY

************************************************************************************
********************************************
---- DATE ----
SYSTEM UPDATED CREATED DESCRIPTION
************************************************************************************
********************************************
DCSYSTEM mm/dd/yy VER 1
DCSYSTEM mm/dd/yy VER 105
DCSYSTEM mm/dd/yy mm/dd/yy VER 9969
LHNSYSTEM mm/dd/yy mm/dd/yy VER 1
LHNTEST mm/dd/yy mm/dd/yy VER 1
QATAWDICT mm/dd/yy mm/dd/yy VER 1 QA TAW SYSTEMNAME
TESTSYSTEM mm/dd/yy mm/dd/yy VER 1
```

System Detail and Key Reports (DREPORTs 003 and 073)

Contents

The system detail report provides detailed information on all system occurrences; the system key report provides similar information for selected system occurrences. The fields and format of these two reports are the same.

The following report shows sample output for DREPORT 073. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=073
KEY SYS-NAME-041 'LHNSYSTEM'

Sample DREPORT 073:

```
REPORT NO. 73 mm/dd/yy PAGE 1
DREPORT 073 SYSTEM REPORT

************************************************************************************
```

Field Descriptions

The format of DREPORTs 003 and 073 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

- **SYSTEM NAME**
  Identifies the name and version number of the system occurrence being described.

- **DESCRIPTION**
  Identifies text associated with the system occurrence.

- **DATE UPDATED/CREATED**
  Identifies the date the system occurrence was last updated and the date the system occurrence was defined to the dictionary.

- **PREPARED BY/REVISED BY**
  Identifies the user who defined the system occurrence to the dictionary and the user who last updated it.
PUBLIC ACCESS
Identifies the level of access allowed for unregistered users.

Class/attribute associations
Identifies class/attribute associations that have been defined for the system occurrence. In this sample report, the attribute DOCUMENT within the class DIVISION is associated with LHNSYSTEM.

Associated entities
Lists associated entities for the system occurrence with any associated text. The following entity types can be associated with a system occurrence: destination, line, logical terminal, map, module, physical terminal, process, program, q-file, queue, table, or task.

USER
Identifies a user who is registered to access the entity and any responsibility code and text associated with the user. If no users are registered for the entity, the USER literal does not appear; if no responsibility code is associated with the user, the RESPONSIBLE FOR literal does not appear.

SAME AS
Identifies the system from which this system definition was copied (if copied).

User-defined nests
Identifies user-defined nests that relate one system occurrence to another. In this sample report, ASSOCIATED TEST SYSTEM is a relational key that relates LHNSYSTEM to TESTSYSTEM.

COMMENT
Identifies comment text associated with the system occurrence through the COMMENTS clause of the DDDL SYSTEM statement.

User-defined comments
Identifies user-defined comments defined through the DEFINITION/comment-key clause of the DDDL SYSTEM statement. In this sample report, DELETION DATE is a comment key.

User Reports (DREPORTs 004, 054, 074)
User reports (DREPORTs 004, 054, and 074) provide information on user occurrences that have been defined to the dictionary. Users are represented in the dictionary as occurrences of the USER-047 record.

User Summary Report (DREPORT 054)
Contents
The user summary report includes the following information on user occurrences:

- User name
- Associated description
- Date defined and date updated (if updated)
Sample DREPORT 054:

REPORT NO. 54
n
DREPORT 054

DATA DICTIONARY REPORTER

REL nn.

PAGE 1

USER REPORT - SUMMARY

************************************************************************************
********************************************
---- D A T E ----

USER NAME DESCRIPTION UPDATED CREATED

************************************************************************************
********************************************

BIARI01 mm/dd/yy

CORP mm/dd/yy

CULL DBA mm/dd/yy

ENK mm/dd/yy

EQA mm/dd/yy

FQA mm/dd/yy

HANEL01 mm/dd/yy

IQA mm/dd/yy

JFD mm/dd/yy

JLK mm/dd/yy

LHN SYSTEM ADMINISTRATOR mm/dd/yy

MJH mm/dd/yy

MQA mm/dd/yy

PAGTO01 mm/dd/yy

PROFUSEA mm/dd/yy

PROFUSER mm/dd/yy

QAE mm/dd/yy

SQA mm/dd/yy

UQA mm/dd/yy

VQA mm/dd/yy

WMF mm/dd/yy

User Detail and Key Reports (DREPORTs 004 and 074)

Contents

The user detail report provides detailed information on all user occurrences; the user key report provides similar information for selected user occurrences. The fields and format of these two reports are the same.
The following report shows sample output for DREPORT 074. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=074
KEY USER-NAME-047 'LHN'

Sample DREPORT 074:

REPORT NO. 74 mm/dd/yy PAGE 1
DREPORT 074 USER REPORT

************************************************************************************
********************************************
---- D A ----
USER NAME UPDATED
CREATED
************************************************************************************

LHN mm/dd
/mm/dd/yy
PREPARED BY LHN
REVISED BY LHN
DESCRIPTION SYSTEM ADMINISTRATOR
FULL NAME LAURIE NELSON
AUTHORITIES:

PASSWORD NO AUTHORITY IDD
DC
CULPRIT NO AUTHORITY ELEMENT NO AUTHORITY
DESTINATION NO AUTHORITY FILE NO AUTHORITY
OLQ NO AUTHORITY MODULE NO AUTHORITY
LINE NO AUTHORITY PROCESS NO AUTHORITY
ADS NO AUTHORITY
LOAD MODULE NO AUTHORITY
MAP NO AUTHORITY QFILE NO AUTHORITY
IDMS MESSAGE NO AUTHORITY
SCHEMA NO AUTHORITY TABLE NO AUTHORITY
PANEL NO AUTHORITY
SUBSCHEMA NO AUTHORITY PROGRAM NO AUTHORITY
PTERM NO AUTHORITY ENTRY POINT NO AUTHORITY
DMCL NO AUTHORITY
QUEUE CLASS & ATTRIBUTE RECORD NO AUTHORITY
CLASS TASK NO AUTHORITY
ATTRIBUTE NO AUTHORITY

DEFAULT OPTIONS
NO OLQ HEADER
INTERRUPT
WHOLE
FULL
NO FILLER
ECHO
HEADER
ALL
COMMENTS
NO PATH STATUS
NO CODE TABLE
NO EXTERNAL PIC
TERSE

&&1.
QFILE ALLOWED
SORT ALLOWED
Sample DREPORT 074 - continued:

REPORT NO.  74   DATA DICTIONARY REPORTER  REL  nn.
 n   mm/dd/yy   PAGE  2   USER REPORT

************************************************************************************
********************************************
---- T E ----
USER NAME  UPDATED
CREATED
************************************************************************************

******************************************************************************

SYSTEM            LHNTEST            VER  1
AUTHORIZED TO     ALL AUTHORITY
PROGRAM            EMPSS01            VER  1
AUTHORIZED TO     UPDATE
ACCESS TO SYSTEM  LHNSSYSTEM        VER  1
ACCESS TO SUBSCHEMA EMPSS01
OF SCHEMA          EMPSCHM          VER  1
RECORD            EMPOSITION        VER  1
RESPONSIBLE FOR    CREATION AND UPDATE AND DELETION
RECORD            EMPOSITION        VER  1
AUTHORIZED TO     UPDATE
ELEMENT            LHN-ELEM          VER  1
RESPONSIBLE FOR    CREATION AND UPDATE AND DELETION
AUTHORIZED TO     ALL AUTHORITY
ELEMENT            DEPT-NAME         VER  1
RESPONSIBLE FOR    CREATION AND UPDATE AND DELETION
AUTHORIZED TO     UPDATE
DESTINATION        TESTDEST          VER  1
RESPONSIBLE FOR    CREATION AND UPDATE AND DELETION
AUTHORIZED TO     ALL AUTHORITY
LINE               VTAM234           VER  1
RESPONSIBLE FOR    CREATION AND UPDATE AND DELETION
COMMENT
00000100 USER WAS CREATED FOR TEST PURPOSES
DEFINITION
00000100 DELETION DATE

Field Descriptions

The format of DREPORTs 004 and 074 depends on the order in which information was defined to the dictionary. A description of the fields in the sample user report follows:

- **USER NAME**
  Identifies the name of the user occurrence.

- **DATE UPDATED/CREATED**
  Identifies the date the user occurrence was last updated and the date the user occurrence was defined to the dictionary.
- **PREPARED BY/REVISED BY**
  Identifies the user who defined the user occurrence to the dictionary and the user who last updated it.

- **DESCRIPTION**
  Identifies text associated with the user occurrence.

- **FULL NAME**
  Identifies a name associated with the user occurrence (typically the user’s full name).

- **PASSWORD ASSIGNED**
  Indicates that a password is assigned to the user.

- **AUTHORITIES**
  Identifies the user’s authority to access secured entities, perform secured operations, or assign or revoke equal authority for another user.

- **DEFAULT OPTIONS**
  Identifies values that CA OLQ uses as default options when the user signs on.

- **CA OLQ options**
  Identifies additional CA OLQ authorizations.

- **OVERRIDES ALLOWED**
  Indicates whether the user can define and update CA Culprit for CA IDMS record layouts and files definitions.

- **IDD SIGNON ALLOWED**
  Indicates whether the user can access the DDDL compiler.

- **IDB ALLOWED**
  Indicates that the user can access IDB.

- **ASF ALLOWED**
  Indicates that the user can access ASF.

- **WITHIN USER**
  Identifies a relationship between this user occurrence and another user occurrence.

- **SAME AS**
  Identifies the user occurrence from which this user occurrence was copied (if copied).

- **User-defined nests**
  Identify user-defined nests that relate one user occurrence to another. In this sample report, the user-defined nest SUPERVISOR relates user LHN to user JFD.

- **Class/attribute associations**
  Identifies class/attribute associations that have been defined for the user occurrence. In this sample report, the attribute ASF within class PRODUCT and the attribute mm/dd/yy within class DATE-OF-HIRE are associated with user LHN.
• **Associated entity occurrences**
  Identifies associated entity occurrences for the user occurrence. Responsibility codes and text associated with each relationship are also listed.

• **ACCESS TO**
  Indicates that the user is authorized to access a CA Culprit for CA IDMS file, a qfile, a signon qfile, a subschema, or a system.

• **COMMENT**
  Identifies comments associated with the user occurrence through the COMMENTS clause of the DDDL USER statement.

• **User-defined comments**
  Identifies user-defined comments that were defined through the DEFINITION/comment-key clause of the DDDL USER statement are listed in this field. In this sample report, DELETION DATE is a comment key.

### Program Reports (DREPORTs 005, 055, 075)

Program reports (DREPORTs 005, 055, and 075) provide information on program occurrences that have been defined to the dictionary. Programs are represented in the dictionary as occurrences of the PROG-051 record.

### Program Summary Report (DREPORT 055)

**Contents**

The program summary report includes the following information on program occurrences:

• Program name and version number

• Subordinate programs (if any)

• Associated description

• Date defined, date updated (if updated), and date compiled (if compiled). Program occurrences that were created or updated by the DC/UCF system generation compiler do not have these dates associated with them.

• Language associated with the program

**Sample DREPORT 055:**

```
REPORT NO. 55
n
DREPORT 055

DATA DICTIONARY REPORTER REL nn.

PROGRAM REPORT - SUMMARY

************************************************************************************
*******************************************************
****** A T E *******
PROGRAM DESCRIPTION COMPILED UP

--- D
```

16-Jan-2018  678/898
## Program Detail and Key Reports (DREPORTs 005 and 075)

### Contents

The program detail report provides detailed information on all program occurrences; the program key report provides similar information on selected program occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 075. The DREPORT and KEY parameters used to create this sample report are:

- **DREPORT=075**
- **KEY PROG-NAME-051 ‘EMPSS01’**

### Sample DREPORT 075:

<table>
<thead>
<tr>
<th>REPORT NO.</th>
<th>DREPORT 75</th>
<th>DATA DICTIONARY REPORTER</th>
<th>REL nn.</th>
<th>PROGRAM REPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>mm/dd/yy</td>
<td>mm/dd/yy</td>
<td>mm/dd/yy</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>ESTIMATED</th>
<th>TIMES</th>
<th>D A T E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LINES</td>
<td>COMPILED</td>
<td>COMPILE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| EMPSSAD | VER 1 | mm/dd/yy |
| EMPSSLR | VER 1 | mm/dd/yy |
| EMPSSLR1| VER 1 | mm/dd/yy |
| EMPSS01 | VER 1 | mm/dd/yy |
| EMPSS01A| VER 1 | mm/dd/yy |
| EMPSS02 | VER 1 | mm/dd/yy |
| ERPT01M | VER 1 | mm/dd/yy |
| GOOD01D | VER 1 | ADS DIALOG |
| CTRL01D | VER 1 | ADS DIALOG |
| GWGAPP01| VER 1 | ADS DIALOG |
| GWGDIA01| VER 1 | ADS DIALOG |
| GWGDIA02| VER 1 | ADS DIALOG |
| GWGDIA03| VER 1 | ADS DIALOG |
| GWGMAP01| VER 1 | mm/dd/yy |
| HELP908 | VER 1 | mm/dd/yy |
| H981924 | VER 1 | mm/dd/yy |
| IDMS    | VER 1 | mm/dd/yy |
| IDMSCOBI| VER 1 | mm/dd/yy |
| IDMSDCCI| VER 1 | mm/dd/yy |
Field Descriptions

The format of DREPORTs 005 and 075 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

- **PROGRAM**
  Identifies the program name and version number.

- **ESTIMATED LINES**
  Identifies the estimated number of source code lines in the program.

- **TIMES COMPILED**
  Identifies the number of times that the program has been compiled. The DML processors automatically update the compile-time statistics if the activity log is on and the dictionary is in UPDATE mode at run time.

- **DATE COMPILED/UPDATED/CREATED**
  Identifies the date the program occurrence was last compiled or updated and the date the program occurrence was defined to the dictionary.

- **PREPARED BY/REVISED BY**
  Identifies the user who defined the program to the dictionary and the user who last updated it.

- **PUBLIC ACCESS**
  Identifies the level of access allowed for unregistered users.

- **DESCRIPTION**
  Identifies text associated with the program occurrence.

- **Class/attribute associations**
  Identifies class/attribute associations that have been defined for the program occurrence. In this sample report, the attribute ASSEMBLER within the class LANGUAGE and the attribute DOCUMENT within class DIVISION are associated with EMPSS01.

- **SAME AS**
  Identifies the program occurrence from which this program occurrence was copied (if copied).
• **SYSTEM**
  Identifies a system associated with the program and any user-supplied text for the program /system relationship.

• **USER**
  Identifies a user who is registered to access the program occurrence and any responsibility codes and associated text defined for the user. If no users are registered, the USER literal does not appear; if no responsibility codes have been assigned, the RESPONSIBLE FOR literal does not appear.

• **SUBPROGRAM**
  Identifies a subprogram called by the program.

• **CALLED BY**
  Identifies a program that calls this program as a subprogram.

• **ENTRY**
  Identifies an entry point for the program.

• **User-defined nests**
  Identifies user-defined nests that relate this program occurrence and another program occurrence. In this sample report, the relational key ASSOCIATED PROGRAM relates EMPSS02 to EMPSS01.

• **MODULE**
  Identifies a module used by the program and any user-supplied text for this program/module relationship.

• **RECORD COPIED**
  Identifies the records and record synonyms that are copied into the program.

• **INPUT FILE**
  Identifies the file opened by the program.

• **SUBSCHEMA OF SCHEMA**
  Identifies a subschema (and associated schema) used by the program.

• **Associated logical records, records, sets, and areas**
  Specifies associated logical records, records, sets, and areas that the program uses or is expected to use are listed in this field. The field can also show the usage mode for the named area and a count of the number of times the function is performed.

• **COMMENT**
  Identifies commentary text provided for this program through the COMMENTS clause of the DDDL PROGRAM statement.

• **MAP USED**
  Identifies the map used by the program.
- **User-defined comments**
  Identifies user-defined comments that were defined through the `DEFINITION/comment-key` clause of the DDDL PROGRAM statement. In this sample report, DELETION DATE is a comment key.

- **ENTRY POINT**
  Identifies the program occurrence as an entry point.

- **WITHIN PROGRAM**
  Identifies the program that uses the program occurrence as an entry point.

**Module Reports (DREPORTs 006, 056, 076)**

Module reports (DREPORTs 006, 056, and 076) provide information on module occurrences that have been defined to the dictionary. Modules, processes, functions, q-files, and edit/code tables are represented in the dictionary as occurrences of the MODULE-067 record.

**Module Summary Report (DREPORT 056)**

**Contents**

The module summary report provides the following information on module occurrences:

- Entity name and version number
- Associated description
- Associated language
- Date defined and date updated (if updated)

**Sample DREPORT 056:**

```
REPORT NO. 56 DATA DICTIONARY REPORTER REL nn.
m/dd/yy PAGE 8 MODULE REPORT - SUMMARY
DREPORT 056

*********************************************************************************
********************************************
---- DATE ----
MODULE NAME VER DESCRIPTION LANGUAG
E UPDATED CREATED
*********************************************************************************
********************************************
EMP-COVER-INS- 1 OLQ OLQ
PLAN mm/dd/yy mm/dd/yy
EMP-DEPT- 1 OLQ OLQ
REPORT mm/dd/yy mm/dd/yy
EMPQFILE 1 OLQ OLQ
mm/dd/yy mm/dd/yy
ERPT01P mm/dd/yy
```

16-Jan-2018 682/898
Module Detail and Key Reports (DREPORTs 006 and 076)

Contents

The module detail report provides detailed information on all module occurrences; the module key report provides similar information on selected module occurrences. The fields and format of these two reports are the same.

The following report shows sample output for DREPORT 076. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=076
KEY MOD-NAME-067 'EMPLOYEE'

Sample DREPORT 076:
Field Descriptions

The format of DREPORTs 006 and 076 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:
- **MODULE NAME**
  Identifies the name and version number of the module occurrence.

- **DATE UPDATED/CREATED**
  Identifies the date the module occurrence was last updated and the date the module occurrence was defined to the dictionary.

- **LANGUAGE**
  Identifies the language associated with the module occurrence.

- **PREPARED BY/REVISED BY**
  Identifies the user who defined the module occurrence to the dictionary and the user who last updated it.

- **DESCRIPTION**
  Identifies the text associated with the module occurrence.

- **PUBLIC ACCESS**
  Identifies the level of access allowed for unregistered users.

- **MODE**
  Identifies the operating mode for the module occurrence. MODE is a system-supplied class.

- **USER**
  Identifies a user who is registered to access this entity and any responsibility code and text associated with the user. If no users are registered for this entity, the USER literal does not appear; if no responsibility code is associated with the user, the RESPONSIBLE FOR literal does not appear.

- **SYSTEM**
  Identifies the system that includes the module and any user-supplied text associated with the module/system relationship.

- **SAME AS**
  Identifies the module occurrence from which this module occurrence was copied (if copied).

- **PROGRAM**
  Identifies the program associated with the module and any user-supplied text for the relationship.

- **User-defined nests**
  Identify user-defined nests that relate the module occurrence and another module occurrence. In this sample report, the relational key ASSOCIATED PRODUCTION PROCESS is used to relate EMPLOYEE to process EMP-DEPT-REPORT.

- **COMMENT**
  Identifies commentary text supplied for this module occurrence through the COMMENTS clause of the DDDL MODULE/PROCESS/QFILE/TABLE statement.

- **User-defined comments**
  Identify user-defined comments that were defined through the DEFINITION/comment-key clause of the DDDL MODULE/PROCESS/QFILE/TABLE statement. In this sample report, DELETION DATE is a comment key.
• **MODULE SOURCE**  
  Lists the source code stored within the module occurrence. The numbers are supplied by the DDDL compiler. If the module occurrence is a q-file, Q-FILE SOURCE appears on the report.

• **CODE TABLE**  
  Identifies the module occurrence as a code table and describes the code table format.

### File Reports (DREPORTs 007, 057, 077)

File reports (DREPORTs 007, 057, and 077) provide information on IDD files. IDD files are represented in the dictionary as occurrences of the SA-018 record.

#### File Summary Report (DREPORT 057)

**Contents**

The following information is included in the file summary report:

- IDD file name and version number
- Associated text
- Label option
- Date defined and date updated (if updated)

**Sample DREPORT 057:**

```
REPORT NO. 57               DATA DICTIONARY REPORTER REL nn.
DREPORT 057 mm/dd/yy PAGE 1            FILE REPORT - SUMMARY

************************************************************************************
********************************************
---- DATE ----
FILE NAME VER DESCRIPTION
LABELS UPDATED CREATED
************************************************************************************

TESTFILE mm/dd/yy mm/dd/yy
```

**File Detail and Key Reports (DREPORTs 007 and 077)**

**Contents**

The file detail report provides detailed information on all IDD files; the file key report provides similar information. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 077. The DREPORT and KEY parameters used to create this sample report are:
Sample DREPORT 077:

DREPORT=077
KEY SA-NAM-018 ‘TESTFILE’

REPORT NO. 77 mm/dd/yy PAGE 1
FILE REPORT

************************************************************************************
*******************************************************
RECORD BLOCK RECORD ---
- D A T E ----
FILE NAME SIZE SIZE FORMAT LABELS UPD
ATED CREATED
************************************************************************************
*******************************************************
TESTFILE VER 1 132 132 F NOT SPECIFIED mm/dd/yy
PREPARED BY LHN
REVISED BY LHN
DESCRIPTION TEST FILE FOR DOCUMENTATION
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY
USER LHN RESPONSIBLE FOR CREATION AND UPDATE AND DELETION
FILE-TYPE PS
DEVICE-TYPE 3380
DIVISION DOCUMENT
PROGRAM LHNPROG VER 1 I-O FILE
RECORD LHN-REC VER 1
FILE SYNONYM TEST-FILE VER 1
RECORD SYNONYM LHN-REC VER 1
FILE SYNONYM LHNFILE VER 1
COMMENT 00000100 THIS IS A DOCUMENTATION TEST FILE
DEFINITION 00000100 DELETION DATE

Field Descriptions

The format of DREPORTs 007 and 077 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

- **FILE NAME**
  Identifies the name and version number of the IDD file occurrence.

- **RECORD SIZE**
  Identifies the maximum record size for this file occurrence.

- **BLOCK SIZE**
  Identifies the size (in bytes) of the record blocks on this file.

- **RECORD FORMAT**
  Identifies whether the record format is fixed (F), undefined (U), variable (V), or variable spanned (S).

- **LABELS**
  Indicates whether the file labels are NOT SPECIFIED, STANDARD, NON-STANDARD, or OMITTED.
• **DATE UPDATED/CREATED**
  Identifies the date the IDD file was last updated and the date the file occurrence was defined to the dictionary.

• **PREPARED BY/REVISED BY**
  Identifies the user who defined the IDD file to the dictionary and the user who last updated it.

• **DESCRIPTION**
  Identifies text associated with the IDD file occurrence.

• **FILE-TYPE**
  Identifies the type of IDD file: PS, IS, CARD, VS, or UM.

• **VSAM-TYPE**
  Identifies the type of VSAM file (if VSAM file): KS, ES, or RS.

• **Class/attribute associations**
  Identify class/attribute associations that have been defined for the IDD file. In this sample report, the attribute DOCUMENT within class DIVISION is associated with TESTFILE.

• **USER**
  Identifies a user who is registered to access the file occurrence and indicates any responsibility code and associated text defined for the user. If no users are registered, the USER literal does not appear; if no responsibility codes are associated with a user, the RESPONSIBLE FOR literal does not appear for that user.

• **SAME AS**
  Identifies the file occurrence from which this file occurrence was copied (if copied).

• **PROGRAM**
  Identifies a program that uses this file occurrence for input/output.

• **EXTERNAL NAME**
  Identifies the ddname or file-id of the file used as input to the program, if defined.

• **RELATED FILE**
  Identifies a relationship between this file occurrence and another.

• **RECORD**
  Identifies a record that is associated with this file.

• **FILE SYNONYM**
  Identifies a file synonym for the file occurrence.

• **RECORD SYNONYM**
  Identifies record synonym names associated with the last file or file synonym listed.

• **COMMENT**
  Identifies commentary text supplied for this file occurrence through the COMMENTS clause of the DDDL FILE statement.
User-defined comments
Identifies user-defined comments that were defined through the DEFINITION/comment-key clause of the DDDL FILE statement. In this sample report, DELETION DATE is a comment key.

User-defined nests
Identifies user-defined nests that relate this file occurrence to another file occurrence.

Record Reports (DREPORTs 008, 038, 058, 078)

Record reports (DREPORTs 008, 038, 058, and 078) provide information on record occurrences that have been defined to the dictionary. Elements that are associated with the record are also listed. Record occurrences are represented in the dictionary as occurrences of the SR-036 record.

Types of Record Reports

A summary report, two key reports, and a detail report are provided for record entities. One key report uses the record name as the key value; the other uses an attribute as the key value.

Record Summary Report (DREPORT 058)

Contents

The record summary report provides the following information on record occurrences:

- Record name and version number
- Associated description
- Record length
- Record storage method
- Estimated number of times the record occurs in files or databases
- Date defined and date updated (if updated)

Sample DREPORT 058:

<table>
<thead>
<tr>
<th>REPORT NO. 58</th>
<th>mm/dd/yy</th>
<th>PAGE 3</th>
<th>DATA DICTIONARY REPORTER</th>
<th>REL nn.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DREPORT 058</td>
<td></td>
<td></td>
<td>RECORD REPORT - SUMMARY</td>
<td></td>
</tr>
</tbody>
</table>

*****************************************************************************
*****************************************************************************

RECORD

<table>
<thead>
<tr>
<th>RECORD NAME</th>
<th>DATE</th>
<th>DESCRIPTION</th>
<th>OCCURRENCES</th>
<th>UPDATED</th>
<th>CREATED</th>
</tr>
</thead>
<tbody>
<tr>
<td>POSITION</td>
<td>100</td>
<td>mm/dd/yy</td>
<td>28</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>EXPERTISE</td>
<td>1</td>
<td></td>
<td></td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

*****************************************************************************
*****************************************************************************
Record Detail and Key Report (DREPORTs 008 and 078)

Contents

The record detail report provides detailed information on all record occurrences; the record key report provides similar information on selected record occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 078. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=078
KEY SR-NAM-036 'LHNREC'

Sample DREPORT 078:

<table>
<thead>
<tr>
<th>RECORD</th>
<th>RECORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>LENGTH</td>
</tr>
<tr>
<td>LHNREC</td>
<td>VER 1</td>
</tr>
</tbody>
</table>

- PREPARED BY: LHN
- REVISED BY: LHN
- USER LHN RESPONSIBLE FOR CREATION AND UPDATE AND DELETION
Field Descriptions

The format of DREPORTs 008 and 078 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

- **RECORD NAME**
  Identifies the name and version number of the record occurrence.

- **RECORD LENGTH**
  Identifies the record length in bytes.

- **BUILDER**
  Identifies the compiler last used to add or update the record occurrence. Possible values are D (DDDL compiler), S (schema compiler), C (mapping compiler), and X (IDMSDIRL).

- **RECORD TYPE**
  Identifies the record storage mode, if defined.

- **OCCURRENCES**
  Identifies the actual or estimated number of times the record occurs in files or databases.

- **DATE UPDATED/CREATED**
  Identifies the date the record occurrence was last updated and the date the record occurrence was added to the dictionary.

- **PREPARED/REVISED BY**
  Identifies the name of the user who defined the record occurrence and the name of the user who last updated it.

- **DESCRIPTION**
  Identifies text associated with the record occurrence.

- **PUBLIC ACCESS**
  Identifies the level of access allowed for unregistered users.

- **USER**
  Identifies a user who is registered to access the record occurrence and identifies the responsibility code and any associated text defined for the user. If no users are registered, the USER literal does not appear; if no responsibility codes have been defined for the user, the RESPONSIBLE FOR literal does not appear.
- **Class/attribute associations**
  Identifies class/attribute associations that have been defined for the record occurrence are listed in this field. In this sample report, the attribute DOCUMENT within class DIVISION is associated with LHNREC.

- **SAME AS**
  Identifies the record occurrence from which this record occurrence was copied (if copied).

- **MODE**
  Identifies the operating mode for the program in which the record is used.

- **LANGUAGE**
  Identifies the language associated with the record occurrence.

- **User-defined nests**
  Identify user-defined nests that relate the record occurrence to another record occurrence. In this sample report, the relational key ASSOCIATED RECORD relates record LHNREC to record TEST-REC.

- **PRIMARY FILE**
  Identifies the file associated with the record.

- **COMMENT**
  Identifies commentary text supplied for the record occurrence through the COMMENTS clause of the DDDL RECORD statement.

- **User-defined comments**
  Identify user-defined comments that were defined through the DEFINITION/OLQ HEADER/CULPRIT HEADER/comment-key clause of the DDDL RECORD statement are listed in this field. In this sample report, DELETION DATE is a comment key.

- **RECORD**
  Identifies a primary record and provides a description of the elements associated with the record. If defined, the language associated with the record and the programs, files, and subschemas associated with the record are displayed.

- **IN SCHEMA/AREA**
  Identifies the name and version number of the schema and the area in which the record occurs.

- **RECORD SYNONYM**
  Identifies a record synonym and provides a description of the elements associated with the record synonym. The associated language is also displayed, if defined.

- **SUBSCHEMA**
  Identifies the subschema in which this record is used and provides a description of the elements associated with the subschema.

- **SUBSCHEMA VIEW**
  Indicates that the record is used in a view and provides a description of the elements (fields) used in the view.
Record/Attribute Key Report (DREPORT 038)

Contents

The record/attribute key report provides information on all records associated with a specified attribute. The entries that can appear on this report are identical to those described for the record key report (DREPORT 078) above.

The following report shows sample output for DREPORT 038. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=038
KEY ATTR-NAME-093 ‘COBOL’

Note that the KEY parameter specifies an attribute name rather than a record name.

Sample DREPORT 038:

<table>
<thead>
<tr>
<th>RECORD NO.</th>
<th>DATA DICTIONARY REPORTED</th>
<th>REL nn.</th>
</tr>
</thead>
<tbody>
<tr>
<td>38</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

--- DATE ---

<table>
<thead>
<tr>
<th>RECORD NAME</th>
<th>LENGTH</th>
<th>BUILDER</th>
<th>TYPE</th>
<th>OCCURREN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

************************************************************************************

*****************************************************

RECORD               RECORD

---- D A T E ----

RECORD NAME          LENGTH  BUILDER  TYPE  OCCURREN

************************************************************************************

*****************************************************

KEY ATTRIBUTE: COBOL

<table>
<thead>
<tr>
<th>DB-STATISTICS</th>
<th>VER 1 100 C</th>
</tr>
</thead>
<tbody>
<tr>
<td>mm/dd/yy</td>
<td>mm/dd/yy</td>
</tr>
<tr>
<td>PREPARED BY</td>
<td>PUBLIC</td>
</tr>
<tr>
<td>REVISED BY</td>
<td>MJH</td>
</tr>
<tr>
<td>LANGUAGE</td>
<td>COBOL</td>
</tr>
<tr>
<td>COPYED IN PROGRAM</td>
<td>LNTN01D</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LANGUAGE</th>
<th>COPYED IN PROGRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>DATE-TODAY</th>
<th>DISPLAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>TIME-TODAY</th>
<th>DISPLAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>PAGES-READ</th>
<th>COMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td></td>
<td>S9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>PAGES-WRITTEN</th>
<th>COMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td></td>
<td>S9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>PAGES-Requested</th>
<th>COMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td></td>
<td>S9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>CALC-TARGET</th>
<th>COMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td></td>
<td>S9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>CALC-OVERFLOW</th>
<th>COMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td></td>
<td>S9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>VIA-TARGET</th>
<th>COMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td></td>
<td>S9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>VIA-OVERFLOW</th>
<th>COMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td></td>
<td>S9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>LINES-REQUESTED</th>
<th>COMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td></td>
<td>S9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>RECS-CURRENT</th>
<th>COMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td></td>
<td>S9</td>
</tr>
</tbody>
</table>
Element Reports (DREPORTs 009, 010, 039, 059, 079)

Element reports (DREPORTs 009, 010, 039, 059, and 079) provide information on element occurrences that have been defined to the dictionary. Elements are represented in the dictionary as occurrences of the INQ-058 record. One summary report, two detail reports, and two key reports are provided for the ELEMENT entity type.

Element Summary Report (DREPORT 059)

Contents

The element summary report provides the following information on element occurrences:

- Element name and version number
- The builder code for the element
- Element length
- Element picture
- Element usage mode
- Whether the JUSTIFY option is on
- Whether the BLANK ON ZERO option is in effect
- Whether the SYNC option is on
- The date defined and date updated (if updated)

**Sample DREPORT 059:**

**Contents**

The element detail report provides detailed information on all element occurrences; the element key report provides similar information on selected element occurrences. The fields and format of these two reports are the same.
The report below shows sample output for DREPORT 079. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=079
KEY INQ-NAM-058 'DEPT-NAME'

Sample DREPORT 079:

REPORT NO. 79 DATA DICTIONARY REPORTER REL nn.

DREPORT 079 ELEMENT REPORT

************************************************************************************
********************************************
BUILD ---- D A T E ----
ELEMENT NAME CODE UPDATED CREATED
************************************************************************************
********************************************
DEPT-NAME VER 1
PREPARED BY DPD
REVISED BY LHN
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY
DEFINITION
100 DELETION DATE
COMMENT 100 THIS ELEMENT IS THE DEPARTMENT HEAD IN THE DEPARTMENT RECORD
ASSOCIATED ELEMENT
NAME

PICTURE X(45) DISPLAY LEN= 45
ELEMENT SYNONYM DEPT-NAME-0410 VER 1 IDD BUILT
RECORD NAME DEPARTMENT
RECORD NAME DEPARTMENT VER 1 IN SCHEMA EMPSCHM
VER 1 RECORD NAME DEPARTMENT VER 1 IN SCHEMA TEST
VER 1 ELEMENT SYNONYM DEPTNAME
RECORD SYNONYM DEPARTMT VER 1
ELEMENT SYNONYM DPNAME
RECORD SYNONYM DEPT VER 1

DEPT-NAME VER 100
PREPARED BY DPD
REVISED BY DPD

Field Descriptions

The format of the DREPORTs 009 and 079 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

- **ELEMENT NAME**
  Identifies the name and version of the element occurrence being described.
- **BUILD CODE**
  Identifies the compiler that last updated the element occurrence. Possible values are D (DDDL compiler) and S (schema compiler).

- **DATE UPDATED/CREATED**
  Identifies the date the element occurrence was last updated and the date the element occurrence was added to the dictionary.

- **PREPARED BY/REVISED BY**
  Identifies the user who defined the element occurrence to the dictionary and the user who last updated it.

- **DESCRIPTION**
  Identifies text associated with the element occurrence.

- **PUBLIC ACCESS**
  Identifies the level of access allowed for unregistered users.

- **Class/attribute associations**
  Identifies class/attribute associations that have been defined for the element occurrence. In this sample report, the attribute DOCUMENT within class DIVISION is associated with DEPT-NAME.

- **USER**
  Identifies a user who is registered to access the element occurrence and any responsibility codes and text associated with the user. If no users are registered, the USER literal does not appear; if no responsibility codes are assigned to a user, the RESPONSIBLE FOR literal does not appear.

- **SAME AS**
  Identifies the element occurrence from which this element occurrence was copied (if copied).

- **User-defined nests**
  Identifies user-defined nests that relate the element occurrence to another element occurrence. In this sample report, the relational key ASSOCIATED ELEMENT relates DEPT-NAME to version 100 of DEPT-NAME.

- **COMMENT**
  Identifies commentary text associated with the element occurrence through the COMMENTS clause of the DDDL ELEMENT statement.

- **User-defined comments**
  Identifies user-defined comments that were defined through the DEFINITION/comment-key clause of the DDDL ELEMENT statement are listed in this field. In this sample report, DELETION DATE is a comment key.

- **SUBORDINATE ELEMENT**
  Identifies a subordinate element of a group element.

- **PRIMARY GROUP**
  Identifies the element as a primary group element and identifies the element usage and length.

- **ELEMENT/RECORD NAME**
  Identify the element and the record that contains the element or element synonym.
WITHIN GROUP
Identifies the group element that includes this subordinate element.

PICTURE/RANGE/VALUE
Identify the characteristics of the element.

Inactive Element Detail Report (DREPORT 010)

Contents

The inactive element detail report lists all element occurrences that are not associated with a record occurrence. The entries that can appear on this report are identical to those described for the active element key report (DREPORT 079) above, with the exception of the fields that describe the element /record associations.

Sample DREPORT 010:

REPORT NO. 10 DATA DICTIONARY REPORTER REL nn.
DREPORT 010 ELEMENT REPORT

************************************************************************************
********************************************
BUILD ---- DATE ----
ELEMENT NAME CODE UPDATED CREATED
************************************************************************************

ADXXTBIF-
MESSAGE          VER 1
PREPARED BY     RSB
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY
PICTURE         X(64)
DISPLAY         LEN= 64

DATE-
ELEM          VER 1
PREPARED BY     GCH
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY
PICTURE         9(8)
DISPLAY         LEN= 8

DC-ATTN-
INT          VER 1
PREPARED BY     PUBLIC
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY
VALUE '4525'
VALUE '4625'
WITHIN GROUP  ERROR-
STATUS       VER 1
PRIMARY GROUP  COND

ERROR-
STATUS        VER 1
PREPARED BY     PUBLIC
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY
Attribute/Element Key Report (DREPORT 039)

Contents

The attribute/element key report provides information on all elements that are associated with a specified attribute, regardless of the class association. The entries that can appear on this report are identical to those described for the element key report (DREPORT 079) above.

The report below shows sample output for DREPORT 039. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=039
KEY ATTR-NAME-093 'DOCUMENT'

Note that these KEY parameters specify attribute names rather than element names.

Sample DREPORT 039:

REPORT NO. 39                      DATA DICTIONARY REPORTER    REL nn.
DREPORT 039                      ELEMENT REPORT

************************************************************************************
********************************************
BUILD ---- D A T E ----
ELEMENT NAME

CODE         UPDATED        CREATED

************************************************************************************
********************************************
KEY ATTRIBUTE: DOCUMENT

DEPT-NAME

VER 1
D mm/dd/yy mm/dd/yy
PREPARED BY DPD
REvised BY LHN
DIVISION DOCUMENT
USER LHN

DEFINITION

100 DELETION DATE

COMMENT

100 THIS ELEMENT IS THE DEPARTMENT HEAD IN THE DEPARTMENT RECORD

ASSOCIATED ELEMENT

DEPT-NAME IN THE DEPARTMENT RECORD

PICTURE

X(45)

DISPLAY LEN= 45

ELEMENT SYNONYM DEPT-NAME-0410

RECORD NAME DEPARTMENT

VER 1 IDD BUILT

RECORD NAME DEPARTMENT

VER 1 IN SCHEMA EMPSCHM

VER 1

RECORD NAME DEPARTMENT

VER 1 IN SCHEMA TEST

VER 1

ELEMENT SYNONYM DPNAME

RECORD SYNONYM DEPARTMT

VER 1

ELEMENT SYNONYM DEPT

RECORD SYNONYM DEPT

VER 1

16-Jan-2018
Site-Specific Entity Reports

Contents

- Class Reports (DREPORTs 001 and 071) (see page 700)
- Attribute Reports (DREPORTs 002 and 072) (see page 702)
- User-Defined Entity Reports (DREPORTs 019 and 089) (see page 704)

The site-specific entity reports provide information on three entity types: class, attribute, and user-defined entities. These entities are typically used to classify entities and to establish relationships between entities beyond the standard relationships provided through IDD.

Classes Supplied at Installation

Two classes, LANGUAGE and MODE, are supplied at installation. Other classes can be defined to further describe your particular environment.

⚠️ Note: For more information on classes, attributes, and user-defined entities, see the CA IDMS IDD DDDL Reference section.

Site-Specific Reports

The following Site-Specific Entity Reports table lists the site-specific reports in order of presentation in this section.

<table>
<thead>
<tr>
<th>DREPORT Module</th>
<th>DREPORT Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>Class Report -- Detail</td>
</tr>
<tr>
<td>071</td>
<td>Class Report -- Key(1)</td>
</tr>
<tr>
<td>002</td>
<td>Attribute Report -- Detail</td>
</tr>
<tr>
<td>072</td>
<td>Attribute Report -- Key(1)</td>
</tr>
<tr>
<td>019</td>
<td>User-Defined Entity Report -- Detail</td>
</tr>
<tr>
<td>089</td>
<td>User-Defined Entity Report -- Key(1)</td>
</tr>
</tbody>
</table>

Class Reports (DREPORTs 001 and 071)

Contents

Class reports provide information on class occurrences and their associated attributes. Classes are represented in the dictionary as occurrences of the CLASS-092 record; attributes are represented as occurrences of the ATTRIBUTE-093 record. There is no summary report for the class entity type; the detail report is concise and can be used in its place.

The class detail report provides information for all class occurrences defined to the dictionary; the class key report provides similar information for selected class occurrences. The fields and format of these two reports are the same.
Sample DREPORT 001:

FIELD DESCRIPTIONS

The format of DREPORTs 001 and 071 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

- **CLASS/ATTRIBUTE**
  Identifies the class and/or attribute being described.
DATE UPDATED/CREATED
Identifies the date the class occurrence was last updated and the date the class occurrence was defined to the dictionary.

ATTRI
Indicates whether attributes are defined automatically (A) or manually (blank) and whether attributes are singular (S) or plural (blank).

DELETION LOCK
Indicates whether the deletion lock for this class occurrence is ON or OFF.

CLASS TYPE
Indicates whether this class is a CLASS (default) or a user-defined ENTITY.

PREPARED BY/REVISED BY
Identifies the user who defined the class occurrence to the dictionary and the user who last updated it.

COMMENT
Identifies commentary text associated with the class occurrence through the COMMENTS clause of the DDDL CLASS statement.

User-defined comments
Identifies user-defined comments that were defined through the DEFINITION/comment-key clause of the DDDL CLASS statement. In this sample report, DELETION DATE is a comment key.

Attributes
Identifies attributes associated with the class occurrence.

Attribute Reports (DREPORTs 002 and 072)

Contents

Attribute reports provide information on attributes defined to the dictionary. Attributes are represented in the dictionary as occurrences of the ATTRIBUTE-093 record. There is no summary report for the attribute entity type; the class detail report may be used in its place.

The attribute detail report provides detailed information on all attribute occurrences; the attribute key report provides similar information on selected attribute occurrences. The fields and format of these two reports are the same.

The following report shows sample output for DREPORT 072. The DREPORT and KEY parameters used to create this report are:

DREPORT=072
KEY ATTR-NAME-093 ‘DIVISION’

Sample DREPORT 072:

```
REPORT NO. 72
n
n
DREPORT 072

mm/dd/yy PAGE 1

DATA DICTIONARY REPORTER REL nn.

ATTRIBUTE REPORT
```
DIVISION

DOCUMENT
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY

USER
SYSTEM LHN
PROGRAM EMPSS01
ELEMENT LHN-ELEM
ELEMENT DEPT-NAME
LINE VTAM234
PHYSICAL TERMINAL TESTERM
LOGICAL TERMINAL JESRDR
DESTINATION TESTDEST
QUEUE TEST
QUEUE TESTQUEUE
TASK TESTBYE
PANEL TEST-PANEL
MAP TEST-MAP

mm/dd/yy OFF
VER 1
VER 1
VER 1
VER 1
VER 1
VER 1
VER 1
VER 1
VER 1
VER 1
VER 1
VER 1
VER 1
VER 1

Field Descriptions

The format of DREPORTs 002 and 072 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

- **CLASS/ATTRIBUTE**
  Identifies the class and attribute being described.

- **DATE UPDATED/CREATED**
  Identifies the date this attribute occurrence was last updated and the date the attribute occurrence was defined to the dictionary.

- **ATTRI A/S**
  Indicates whether attributes in this class are defined automatically (A) or manually (blank) and whether attributes in this class are singular (S) or plural (blank).

- **DELETION LOCK**
  Indicates whether the deletion lock for the attribute is ON or OFF.

- **PREPARED/REVISED BY**
  Identifies the user who defined the attribute occurrence and the user who last updated it.

- **PUBLIC ACCESS**
  Identifies the level of access allowed for unregistered users.

- **Associated entity occurrences**
  Identifies associated entity occurrences that have been defined for the attribute occurrence.

- **COMMENT**
  Identifies commentary text associated with the attribute occurrence through the COMMENTS clause of the DDDL ATTRIBUTE statement.
User-defined comments
Identifies user-defined comments that were defined through the DEFINITION/comment-key clause of the DDDL ATTRIBUTE statement.

User-Defined Entity Reports (DREPORTs 019 and 089)

Contents

User-defined entity reports provide information on user-defined entity occurrences that have been defined to the dictionary. User-defined entities are represented in the dictionary as occurrences of the CLASS-092 record. The user-defined entity detail report provides detailed information on all user-defined entity occurrences; the user-defined entity key report provides similar information on selected user-defined entity occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 089. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=089
KEY CLASS-NAME-092 'DIVISION'

Sample DREPORT 089:

Field Descriptions

The format of DREPORTs 098 and 089 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:
• **Class name**  
Identifies the class name of the user-defined entity being described.

• **DATE UPDATED/CREATED**  
Identifies the date the user-defined entity occurrence was last updated and the date the user-defined entity occurrence was defined to the dictionary.

• **DELETION LOCK**  
Indicates whether the deletion lock is ON or OFF.

• **Attribute name**  
Identifies the attribute associated with this user-defined entity.

• **PREPARED BY/REVISED BY**  
Identifies the user who defined the user-defined entity occurrence to the dictionary and the user who last updated it.

• **Associated entities**  
Identifies associated entities for the user-defined entity.

• **Class/attribute associations**  
Identifies class/attribute associations that have been defined for the attribute occurrence. In this sample report, the attributes ASF and OLQ within class PRODUCT are associated with attribute DOCUMENT.

• **User-defined nests**  
Identifies user-defined nests that relate one user-defined entity occurrence to another. In this sample report, the relational key ASSOCIATED ATTRIBUTE relates attribute SYSDOC within class DOCTYPE to DOCUMENT.

• **COMMENT**  
Identifies commentary text associated with the user-defined entity occurrence through the COMMENTS clause of the DDDL user-defined entity statement.

• **User-defined comments**  
Identifies user-defined comments that were defined through the DEFINITION/comment-key clause of the DDDL user-defined entity statement. In this sample report, DELETION DATE is a comment key.

### Teleprocessing Entity Reports

**Contents**

- **Task Reports (DREPORTs 011, 061, 081)** (see page 707)
  - Task Summary Report (DREPORT 061) (see page 707)
  - Task Detail and Key Reports (DREPORTs 011 and 081) (see page 708)
- **Queue Reports (DREPORTs 012, 062, 082)** (see page 709)
  - Queue Summary Report (DREPORT 062) (see page 709)
  - Queue Detail and Key Reports (DREPORTs 012 and 082) (see page 710)
- **Destination Reports (DREPORTs 013, 063, 083)** (see page 712)
The teleprocessing entity reports provide information on the following entity types: task, queue, destination, logical terminal, physical terminal, line, panel, and map. These entity types correspond to the standard components for online systems.

Summary of Teleprocessing Entity Reports

The following table lists the teleprocessing entity reports in the order of presentation in this section.

<table>
<thead>
<tr>
<th>DREPORT Module</th>
<th>DREPORT Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>011</td>
<td>Task Report -- Detail</td>
</tr>
<tr>
<td>061</td>
<td>Task Report -- Summary</td>
</tr>
<tr>
<td>081</td>
<td>Task Report -- Key(1)</td>
</tr>
<tr>
<td>012</td>
<td>Queue Report -- Detail</td>
</tr>
<tr>
<td>062</td>
<td>Queue Report -- Summary</td>
</tr>
<tr>
<td>082</td>
<td>Queue Report -- Key(1)</td>
</tr>
<tr>
<td>013</td>
<td>Destination Report -- Detail</td>
</tr>
<tr>
<td>063</td>
<td>Destination Report -- Summary</td>
</tr>
<tr>
<td>083</td>
<td>Destination Report -- Key(1)</td>
</tr>
<tr>
<td>014</td>
<td>Logical Terminal Report -- Detail</td>
</tr>
<tr>
<td>064</td>
<td>Logical Terminal Report -- Summary</td>
</tr>
<tr>
<td>084</td>
<td>Logical Terminal Report -- Key(1)</td>
</tr>
<tr>
<td>015</td>
<td>Physical Terminal Report -- Detail</td>
</tr>
<tr>
<td>065</td>
<td>Physical Terminal Report -- Summary</td>
</tr>
<tr>
<td>085</td>
<td>Physical Terminal Report -- Key(1)</td>
</tr>
<tr>
<td>016</td>
<td>Line Report -- Detail</td>
</tr>
<tr>
<td>066</td>
<td>Line Report -- Summary</td>
</tr>
<tr>
<td>086</td>
<td>Line Report -- Key(1)</td>
</tr>
</tbody>
</table>
CA IDMS - 19.0

<table>
<thead>
<tr>
<th>DREPORT Module</th>
<th>DREPORT Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>017</td>
<td>Panel Report -- Detail</td>
</tr>
<tr>
<td>067</td>
<td>Panel Report -- Summary</td>
</tr>
<tr>
<td>087</td>
<td>Panel Report -- Key(1)</td>
</tr>
<tr>
<td>018</td>
<td>Map Report -- Detail</td>
</tr>
<tr>
<td>068</td>
<td>Map Report -- Summary</td>
</tr>
<tr>
<td>088</td>
<td>Map Report -- Key(1)</td>
</tr>
</tbody>
</table>

**Note:** Key reports cannot be run with summary or detail reports.

**Task Reports (DREPORTs 011, 061, 081)**

Task reports (DREPORTs 011, 061, and 081) provide information on the task occurrences defined to the dictionary. Tasks are represented in the dictionary as occurrences of the TASK-025 record.

**Task Summary Report (DREPORT 061)**

**Contents**

The task summary report provides the following information on task occurrences:

- Task name and version number
- Associated text
- Date defined and date updated (if updated)

**Sample DREPORT 061:**

```
REPORT NO. 61          DATA DICTIONARY REPORTER  REL nn.
DREPORT 061            TASK REPORT SUMMARY

*************************************************************************************
********************************************
---- DATE ----
TASK UPDATED             CREATED
*************************************************************************************
********************************************

BYE VER 1                mm/dd/yy   mm/dd/yy
CLIST VER 1              mm/dd/yy   mm/dd/yy
CLOD VER 1               mm/dd/yy   mm/dd/yy
COBINPUT VER 1           mm/dd/yy
```
Task Detail and Key Reports (DREPORTs 011 and 081)

Contents

The task detail report provides detailed information on all task occurrences; the task key report provides similar information on selected task occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 081. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=081
KEY TASK-NAME-025 'TESTBYE'

Sample DREPORT 081:

REPORT NO. 81 DATA DICTIONARY REPORTER REL nn.
DREPORT 081 TASK REPORT

************************************************************************************
********************************************
---- DATE ----
TASK UPDATED CREATED
************************************************************************************
********************************************

TESTBYE VER 1

PREPARED BY LHN
REVISED BY LHN
DESCRIPTION TEST TASK
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY
USER LHN RESPONSIBLE FOR CREATION AND UPDATE AND DELETION
WITHIN SYSTEM LHNSYSTEM

INVOKES PROGRAM RHDCBYE VER 1

DIVISION DOCUMENT

COMMENT
00000100 THIS IS A TEST TASK OCCURRENCE

Field Descriptions

The format of DREPORTs 011 and 081 depends on the order in which information was defined to the dictionary. A description of the fields on the sample report follows:

- TASK
  Identifies the name and version number of the task occurrence being described.
• **DATE UPDATED/CREATED**
  Identifies the date the task occurrence was last updated (if updated) and the date the task occurrence was defined to the dictionary. Task occurrences that are defined through the DC/UCF system generation compiler do not have these dates associated with them.

• **PREPARED BY/REVISED BY**
  Identifies the user who defined the task occurrence to the dictionary and the user who last updated it.

• **DESCRIPTION**
  Identifies text associated with the task occurrence.

• **PUBLIC ACCESS**
  Identifies the level of access allowed for unregistered users.

• **USER**
  Identifies a user who is registered to access the task occurrence and identifies any responsibility codes and associated text defined for the user. If no users are registered for the task, the USER literal does not appear; if no responsibility codes are associated with a user, the RESPONSIBLE FOR literal does not appear.

• **WITHIN SYSTEM...INVOKES PROGRAM**
  Identifies a system associated with the task occurrence and the initial program invoked by the task.

• **Class/attribute associations**
  Identifies class/attribute associations that have been defined for the task. In this sample report, the attribute DOCUMENT within class DIVISION is associated with TESTBYE.

• **COMMENT**
  Identifies commentary text associated with the task through the COMMENTS clause of the DDDL TASK statement.

• **User-defined comments**
  Identifies user-defined comments that were defined through the DEFINITION/comment-key clause of the DDDL TASK statement.

Queue Reports (DREPORTs 012, 062, 082)

Queue reports (DREPORTs 012, 062, and 082) provide information on queue occurrences that have been defined to the dictionary. Queues are represented in the dictionary as occurrences of the QUEUE-030 record.

Queue Summary Report (DREPORT 062)

**Contents**

The queue summary report provides the following information on queue occurrences:

* Queue name and version number*
Queue Detail and Key Reports (DREPORTs 012 and 082)

Contents

The queue detail report provides detailed information on all queue occurrences; the queue key report provides similar information on selected queue occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 082. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=82
KEY QUEUE-NAME-030 'TESTQUEUE'

Sample DREPORT 082:

<table>
<thead>
<tr>
<th>REPORT NO. 82</th>
<th>mm/dd/yy</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DREPORT 082</td>
<td>mm/dd/yy</td>
<td>1</td>
</tr>
</tbody>
</table>

QUEUE REPORT

************************************************************************************

--- DATE ---

TESTQUEUE

PREPARED BY LHN
DESCRIPTION
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY
USER LHN
RESPONSIBLE FOR CREATION AND UPDATE AND DELETION
USER JFD

16-Jan-2018
Field Descriptions

The format of DREPORTs 012 and 082 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

- **QUEUE**
  Identifies the name and version number of the queue occurrence being described.

- **DATE UPDATED/CREATED**
  Identifies the date on which the queue occurrence was last updated and the date the queue occurrence was defined to the dictionary. Queue occurrences that were defined through the DC/UCF system generation compiler do not have these dates associated with them.

- **PREPARED BY/REVISED BY**
  Identifies the user who defined the queue occurrence and the user who last updated it.

- **DESCRIPTION**
  Identifies text associated with the queue occurrence.

- **PUBLIC ACCESS**
  Identifies the level of access allowed for unregistered users.

- **USER**
  Identifies a user who is registered to access the queue occurrence and any responsibility code and associated text defined for the user. If no users are registered, the USER literal does not appear; if no responsibility codes have been defined for a user, the RESPONSIBLE FOR literal does not appear.

- **WITHIN SYSTEM**
  Identifies a system that is associated with the queue occurrence.

- **Class/attribute associations**
  Identifies class/attribute associations that have been defined for the queue occurrence. In this sample report, the attribute DOCUMENT within class DIVISION is associated with TESTQUEUE.

- **COMMENT**
  Identifies commentary text associated with the queue occurrence through the COMMENTS clause of the DDDL QUEUE statement.

- **User-defined comments**
  Identifies user-defined comments that were defined though the DEFINITION/comment-key clause of the DDDL QUEUE statement. In this sample report, DELETION DATE is a comment key.
Destination Reports (DREPORTs 013, 063, 083)

Destination reports (DREPORTs 013, 063, and 083) provide information on destination occurrences that have been defined to the dictionary. Destinations are represented in the dictionary as occurrences of the DEST-028 record.

Destination Summary Report (DREPORT 063)

Contents

The destination summary report provides the following information on destination occurrences:

- Destination name and version number
- Associated description
- Date defined and date updated (if updated)

Sample DREPORT 063:

```
REPORT NO. 63 DATA DICTIONARY REPORTER REL nn.
DREPORT 063 DESTINATION REPORT SUMMARY

************************************************************************************
********************************************
---- DATE ----
DESTINATION UPDATED CREATED
************************************************************************************
********************************************
TESTDEST VER 1 TEST DESTINATION
```

Destination Detail and Key Reports (DREPORTs 013 and 083)

Contents

The destination detail report provides detailed information on all destination occurrences; the destination key report provides similar information on selected destination occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 083. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=083
KEY DEST-NAME-028 'TESTDEST'

Sample DREPORT 083:

```
REPORT NO. 83 DATA DICTIONARY REPORTER REL nn.
DREPORT 083 DESTINATION REPORT
```
Field Descriptions

The format of DREPORTs 013 and 083 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

- **DESTINATION**
  Identifies the name and version number of the destination being described.

- **DATE UPDATED/CREATED**
  Identifies the date the destination occurrence was last updated and the date the destination occurrence was defined to the dictionary.

- **PREPARED BY/REVISED BY**
  Identifies the user who defined the destination occurrence to the dictionary and the user who last updated it.

- **DESCRIPTION**
  Identifies text associated with this destination occurrence.

- **PUBLIC ACCESS**
  Identifies the level of access allowed to unregistered users.

- **USER**
  Identifies a user who is registered to access this destination occurrence and any responsibility code and associated text defined for the user. If no users are registered, the USER literal does not appear; if no responsibility codes are assigned to a user, the RESPONSIBLE FOR literal does not appear.

- **WITHIN SYSTEM**
  Identifies a system associated with this destination occurrence.

- **Class/attribute associations**
  Identifies class/attribute associations that have been defined for this destination occurrence. In this sample report, the attribute DOCUMENT within class DIVISION is associated with TESTDEST.
• **COMMENT**
  Identifies commentary text associated with this destination through the COMMENTS clause of the DDDL DESTINATION statement.

• **User-defined comments**
  Identifies user-defined comments that were defined through the DEFINITION/comment-key clause of the DDDL DESTINATION statement.

Logical Terminal Reports (DREPORTs 014, 064, 084)

Logical terminal reports (DREPORTs 014, 064, and 084) provide information on logical terminal occurrences that have been defined to the dictionary. Logical terminals are represented in the dictionary as occurrences of the LTRM-106 record.

Logical Terminal Summary Report (DREPORT 064)

**Contents**

The logical terminal summary report provides the following information on logical terminal occurrences:

- Logical terminal name and version number
- Associated description
- Date defined and date updated (if updated)

**Sample DREPORT 064:**

```
REPORT NO. 64          DATA DICTIONARY REPORTER REL nn.
  064             mm/dd/yy   PAGE 1                       LOGICAL TERMINAL REPORT SUMMARY
DREPORT            mm/dd/yy

------------------------------------------------------------------------------------------------------------------------
---- DATE ----
LOGICAL TERMINAL  UPDATED     CREATED
------------------------------------------------------------------------------------------------------------------------
CONSOLE     VER 1
  mm/dd/yy    mm/dd/yy
JESRDR      VER 1 TEST LTERM FOR DOCUMENTATION
  mm/dd/yy
UCFLTB1     VER 1
  mm/dd/yy
UCFLTB2     VER 1
  mm/dd/yy
```
Logical Terminal Detail and Key Reports (DREPORTs 014 and 084)

Contents

The logical terminal detail report provides detailed information on logical terminal occurrences; the logical terminal key report provides similar information on selected logical terminal occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 084. The DREPORT and KEY parameter used to create this sample report are:

DREPORT=084
KEY LTRM-NAME-106 ‘JESRDR’

Sample DREPORT 084:

REPORT NO. 84 REPORT NO. 84
DATA DICTIONARY REPORTER REL nn. PAGE 1
DREPORT 084 LOGICAL TERMINAL REPORT

************************************************************************************
********************************************
---- DATE ----
LOGICAL TERMINAL
UPDATED CREATED

************************************************************************************
********************************************

JESRDR VER 1
mm/dd/yy
PREPARED BY LHN
DESCRIPTION TEST LTERM FOR DOCUMENTATION
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY
USER LHN RESPONSIBLE FOR CREATION AND UPDATE AND DELETION
WITHIN SYSTEM LHNSYSTEM VER 1
DIVISION DOCUMENT
COMMENT
00000100 THIS IS A TEST LOGICAL TERMINAL DEFINITION
00000100 DELETION DATE

Field Descriptions

The format of DREPORTs 014 and 084 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

- **LOGICAL TERMINAL**
  Identifies the name and version number of the logical terminal being described.

- **DATE UPDATED/CREATED**
  Identifies the date the logical terminal occurrence was last updated and the date the logical terminal occurrence was defined to the dictionary.

- **PREPARED BY/REVISED BY**
  Identifies the user who defined the logical terminal occurrence to the dictionary and the user who last updated it.
- **DESCRIPTION**  
  Identifies text associated with the logical terminal occurrence.

- **PUBLIC ACCESS**  
  Identifies the level of access allowed for unregistered users.

- **USER**  
  Identifies a user who is registered for access to the logical terminal occurrence and any responsibility code and associated text defined for the user. If no users are registered for this occurrence, the USER literal does not appear; if no responsibility codes have been defined for the user, the RESPONSIBLE FOR literal does not appear.

- **WITHIN SYSTEM...PHYSICAL TERMINAL**  
  Identifies a system and the physical terminals associated with the logical terminal occurrence.

- **Class/attribute associations**  
  Identifies class/attribute associations that have been defined for the logical terminal occurrence. In this sample report, the attribute DOCUMENT within class DIVISION is associated with JESRDR.

- **COMMENT**  
  Identifies commentary text associated with the logical terminal through the COMMENTS clause of the DDDL LOGICAL-TERMINAL statement.

- **User-defined comments**  
  Identifies user-defined comments that were defined through the DEFINITION/comment-key clause of the DDDL LOGICAL-TERMINAL statement. In this sample report, DELETION DATE is a comment key.

---

**Physical Terminal Reports (DREPORTs 015, 065, 085)**

Physical terminal reports (DREPORTs 015, 065, and 085) provide information on physical terminal occurrences that have been defined to the dictionary. Physical terminals are represented in the dictionary as occurrences of the PTRM-074 record.

**Physical Terminal Summary Report (DREPORT 065)**

**Contents**

The physical terminal summary report provides the following information on physical terminal occurrences:

- Physical terminal name and version number
- Associated text
- Date defined and date updated (if updated)

**Sample DREPORT 065:**

<table>
<thead>
<tr>
<th>REPORT NO. 65</th>
<th>mm/dd/yy</th>
<th>PAGE</th>
<th>DATA DICTIONARY REPORTER REL nn.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DREPORT 065</td>
<td></td>
<td>1</td>
<td>PHYSICAL TERMINAL REPORT SUMMARY</td>
</tr>
</tbody>
</table>
Physical Terminal Detail and Key Reports (DREPORTs 015 and 085)

Contents

The physical terminal detail report provides detailed information on physical terminal occurrences; the physical terminal key report provides similar information on selected physical terminal occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 085. The DREPORT and KEY parameters used to create this sample report are shown below:

DREPORT=085
KEY PTRM-NAME-074 ‘TESTTERM’

Sample DREPORT 085:

REPORT NO. 85 mm/dd/yy PAGE 1
DREPORT 085

************************************************************************************
********************************************
---- D A T E ----
PHYSICAL TERMINAL
UPDATED CREATED
************************************************************************************

TESTERM VER 1 mm/dd/yy
PREPARED BY LHN
REVISED BY LHN
DESCRIPTION TEST PHYSICAL TERMINAL FOR DOCUMENTATION
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY
USER LHN RESPONSIBLE FOR CREATION AND UPDATE AND DELETION
WITHIN SYSTEM LHNSYSTEM VER 1
Field Descriptions

The format of DREPORTs 015 and 085 depends on the order in which the information was defined to the dictionary. A description of the fields in the sample report follows:

- **PHYSICAL TERMINAL**
  Identifies the name and version number of the physical terminal being described.

- **DATE UPDATED/CREATED**
  Identifies the date the physical terminal occurrence was last updated and the date the physical terminal occurrence was defined to the dictionary. Physical terminal occurrences that are defined through the DC/UCF system generation compiler do not have these dates associated with them.

- **PREPARED BY/REVISED BY**
  Identifies the user who defined the physical terminal occurrence to the dictionary and the user who last updated it.

- **DESCRIPTION**
  Identifies text associated with the physical terminal occurrence.

- **PUBLIC ACCESS**
  Identifies the level of access allowed for unregistered users.

- **USER**
  Identifies a user who is registered to access the physical terminal occurrence and any responsibility code and associated text defined for the user. If no users are registered, the USER literal does not appear; if no responsibility codes are assigned, the RESPONSIBLE FOR literal does not appear.

- **WITHIN SYSTEM ...LINE**
  Identifies a system associated with the physical terminal and any associated lines.

- **Class/attribute associations**
  Identifies class/attribute associations that have been defined for the physical terminal occurrence. In this sample report, the attribute DOCUMENT within class DIVISION is associated with TESTTERM.

- **COMMENT**
  Identifies commentary text associated with the physical terminal through the COMMENTS clause of the DDDL PHYSICAL-TERMINAL statement.

- **User-defined comments**
  Identifies user-defined comments that were defined through the DEFINITION(comment-key) clause of the DDDL PHYSICAL-TERMINAL statement are listed in this field. In this sample report, DELETION DATE is a comment key.
Line Reports (DREPORTs 016, 066, 086)

Line reports (DREPORTs 016, 066, and 086) provide information on line occurrences that have been defined to the dictionary. Lines are represented in the dictionary as occurrences of the LINE-109 record.

Line Summary Report (DREPORT 066)

Contents

The line summary report provides the following information on line occurrences:

- Line name and version number
- Associated description
- Date defined and date updated (if updated)

Sample DREPORT 066:

```
REPORT NO.  66       DATA DICTIONARY REPORTER REL nn.
DREPORT 066       LINE REPORT SUMMARY

************************************************************************************
********************************************
---- DATE ----
LINE
UPDATED CREATED
************************************************************************************

CONSOLE         VER 1
                mm/dd/yy     mm/dd/yy
JESRDR          VER 1
                                mm/dd/yy
UCFLINE         VER 1
                mm/dd/yy     mm/dd/yy
VTAMLIN         VER 1
                mm/dd/yy     mm/dd/yy
```

Line Detail and Key Reports (DREPORTs 016 and 086)

Contents

The line detail report provides detailed information on all line occurrences defined to the dictionary; the line key report provides similar information on selected line occurrences. The fields and format of these two reports are the same.
The report below shows sample output for DREPORT 086. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=86
KEY LINE-NAME-109 'VTAM234'

Sample DREPORT 086:

```plaintext
REPORT NO. 86 mm/dd/yy PAGE 1
DREPORT 086 LINE REPORT

************************************************************************************
********************************************
---- DATE ----
LINE UPDATED CREATED
************************************************************************************

VTAM234 VER 1 mm/dd/yy mm/dd/yy
PREPARED BY LHN REVISED BY LHN
DESCRIPTION TEST LINE OCCURRENCE
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY
USER LHN RESPONSIBLE FOR CREATION AND UPDATE AND DELETION
WITHIN SYSTEM LHNSYSTEM VER 1
DIVISION DOCUMENT
COMMENT 00000100 THIS IS A TEST LINE OCCURRENCE
```

Field Descriptions

The format of DREPORTs 016 and 086 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

- **LINE**
  Identifies the name and version number of the line being described.

- **DATE UPDATED/CREATED**
  Identifies the date the line occurrence was last updated and the date the line occurrence was defined to the dictionary.

- **PREPARED BY/REVISED BY**
  Identifies the user who defined the line occurrence to the dictionary and the user who last updated it.

- **DESCRIPTION**
  Identifies text associated with the line occurrence.

- **PUBLIC ACCESS**
  Identifies the level of access allowed for unregistered users.
Panel Reports (DREPORTs 017, 067, 087)

Panel reports (DREPORTs 017, 067, and 087) provide information on panel occurrences that have been defined to the dictionary. Panels are represented in the dictionary as occurrences of the PANEL-118 record.

Panel Summary Report (DREPORT 067)

Contents

The panel summary report provides the following information on panel occurrences:

- Panel name and version number
- Associated description
- Date defined and date updated (if updated)

Sample DREPORT 067:

```
REPORT NO. 67                     DATA DICTIONARY REPORTER   REL nn.
DREPORT 067 mm/dd/yy PAGE 1      PANEL REPORT SUMMARY

************************************************************************************
********************************************
---- DATE ----
PANEL UPDATED CREATED

************************************************************************************
```

- - - - - D A T E - - - - -

- - - - - - - - - - -

- - - - - - - - - - -

- - - - - - - - - - -

- - - - - - - - - - -

- - - - - - - - - - -

- - - - - - - - - - -

- - - - - - - - - - -

- - - - - - - - - - -
Panel Detail and Key Reports (DREPORTs 017 and 087)

Contents

The panel detail report provides detailed information on all panel occurrences; the panel key report provides similar information on selected panel occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 087. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=087
KEY PANEL-NAME-118 'TEST-PANEL'

Sample DREPORT 087:

<table>
<thead>
<tr>
<th>REPORT NO.</th>
<th>DATA DICTIONARY REPORTER</th>
<th>REL nn.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DREPORT 087</td>
<td>PANEL REPORT</td>
<td></td>
</tr>
</tbody>
</table>

************************************************************************************
********************************************
---- DATE ----
PANEL

---------- D A T E ----------

UPDATED CREATED

************************************************************************************

TEST-PANEL

<table>
<thead>
<tr>
<th>PREPARED BY</th>
<th>REVISED BY</th>
<th>DESCRIPTION</th>
<th>PUBLIC ACCESS</th>
<th>USER</th>
<th>RESPONSIBLE FOR CREATION AND UPDATE AND DELETION</th>
</tr>
</thead>
<tbody>
<tr>
<td>LHN</td>
<td>LHN</td>
<td>TEST PANEL FOR DOCUMENTATION</td>
<td>ALLOWED FOR ALL AUTHORITY</td>
<td>LHN</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MAP</th>
<th>WITHIN SYSTEM</th>
<th>DIVISION</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST-MAP</td>
<td>LHNSYSTEM</td>
<td>DOCUMENT</td>
<td></td>
</tr>
</tbody>
</table>

VER 1
Field Descriptions

The format of DREPORTs 017 and 087 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

- **PANEL**
  Identifies the name and version number of the panel being described. The -OLMPANEL suffix indicates that the panel occurrence was defined to the dictionary by the OLM compiler. The suffix -AUTOPANEL indicates that a panel was defined to the dictionary through the AUTOPANEL option of the DC/UCF mapping compiler.

- **DATE UPDATED/CREATED**
  Identifies the date the panel occurrence was last updated and the date the panel occurrence was defined to the dictionary. Panel occurrences that are defined through the DC/UCF system generation compiler do not have these dates associated with them.

- **PREPARED BY/REVISED BY**
  Identifies the user who defined the panel occurrence to the dictionary and the user who last updated it.

- **DESCRIPTION**
  Identifies text associated with the panel occurrence.

- **PUBLIC ACCESS**
  Identifies the level of access allowed for unregistered users.

- **USER**
  Identifies a user who is registered to access the panel occurrence and any responsibility codes and associated text defined for the user. If no users are registered, the USER literal does not appear; if no responsibility codes are assigned to the user, the RESPONSIBLE FOR literal does not appear.

- **MAP...WITHIN SYSTEM**
  Identifies a map associated with the panel occurrence and the system associated with the map.

- **Class/attribute associations**
  Identifies class/attribute associations that have been defined for the panel occurrence. In this sample report, the attribute DOCUMENT within class DIVISION is associated with TEST-PANEL.

- **COMMENT**
  Identifies commentary text associated with the panel occurrence through the COMMENTS clause of the DDDL PANEL statement.

- **User-defined comments**
  Identifies user-defined comments that were defined through the DEFINITION/comment-key clause of the DDDL PANEL statement. In this sample report, DELETION DATE is a comment key.
Map Reports (DREPORTs 018, 068, 088)

Map reports (DREPORTs 018, 068, and 088) provide information on map occurrences that have been defined to the dictionary. Maps are represented in the dictionary as occurrences of the MAP-098 record.

Map Summary Report (DREPORT 068)

Contents

The map summary report provides the following information on map occurrences:

- Map name and version number
- Associated description
- Date defined and date updated (if updated)

Sample DREPORT 068:

```
REPORT NO. 68 DATA DICTIONARY REPORTER REL nn.
DREPORT 068 MAP REPORT SUMMARY

************************************************************************************
********************************************
---- DATE ----
MAP UPDATED CREATED
************************************************************************************

ABCD01M VER 1 mm/dd/yy mm/dd/yy
ABIF01M VER 1 mm/dd/yy mm/dd/yy
ADDS01M VER 1 mm/dd/yy mm/dd/yy
ADM01M VER 1 mm/dd/yy mm/dd/yy
ADM01M VER 1 mm/dd/yy mm/dd/yy
ADOLQ01M VER 1 mm/dd/yy mm/dd/yy
ADOL01M VER 1 mm/dd/yy mm/dd/yy
ADPT01M VER 1 mm/dd/yy mm/dd/yy
ADRP01M VER 1 mm/dd/yy mm/dd/yy
ADRP01M VER 1 DEFAULT RESPONSE PTF 87-06-1041 mm/dd/yy mm/dd/yy
ADSL01M VER 1 mm/dd/yy mm/dd/yy
ADXXMBIF VER 1 mm/dd/yy mm/dd/yy
```
Map Detail and Key Reports (DREPORTs 018 and 088)

Contents

The map detail report provides detailed information on all map occurrences; the map key report provides similar information on selected map occurrences. The fields and format of these two reports are the same.

The report below shows sample output for DREPORT 088. The DREPORT and KEY parameters used to create this sample report are:

DREPORT=88
KEY MAP-NAME-098 ‘TEST-MAP’

Sample DREPORT 088:

REPORT NO. 88

DATA DICTIONARY REPORTER REL nn.

DREPORT 088 MAP REPORT

************************************************************************************
********************************************
---- DATE ----
MAP UPDATED CREATED
************************************************************************************

TEST- MAP VER 1

PREPARED BY LHN
DESCRIPTION THIS IS A TEST MAP FOR DOCUMENTATION
PUBLIC ACCESS ALLOWED FOR ALL AUTHORITY USER LHN RESPONSIBLE FOR CREATION AND UPDATE AND DE
LETION
PROGRAM EMPSS01 VER 1
WITHIN SYSTEM LHNSYSTEM VER 1
DIVISION DOCUMENT
COMMENT
DEFINITION
DELETION DATE

Field Descriptions

The format of DREPORTs 018 and 088 depends on the order in which information was defined to the dictionary. A description of the fields in the sample report follows:

- **MAP**
  Identifies the name and version number of the map being described.

- **DATE UPDATED/CREATED**
  Identifies the date the map occurrence was last updated and the date the map occurrence was defined to the dictionary.
PREPARED BY/REVISED BY
Identifies the user who defined the map occurrence to the dictionary and the user who last updated it.

DESCRIPTION
Identifies text associated with the map occurrence.

PUBLIC ACCESS
Identifies the level of access allowed for unregistered users.

USER
Identifies a user who is registered to access the map occurrence and any responsibility codes and associated text defined for the user. If no users are registered, the USER literal does not appear; if no responsibility codes are assigned to the user, the RESPONSIBLE FOR literal does not appear.

PROGRAM
Identifies a program that uses the map occurrence.

WITHIN SYSTEM
Identifies a system associated with the map occurrence.

Class/attribute associations
Identify class/attribute associations that have been defined for the map occurrence. In this sample report, the attribute DOCUMENT within class DIVISION is associated with TEST-MAP.

RECORD
Identifies a record that contains record elements used in the map occurrence.

COMMENT
Identifies commentary text associated with the map occurrence through the COMMENTS clause of the DDDL MAP statement.

User-defined comments
Identifies user-defined comments that were defined through the DEFINITION/comment-key clause of the DDDL MAP statement. In this sample report, DELETION DATE is a comment key.

Cross-Reference Reports

Contents
- File/Record Cross-Reference Report (DREPORT 020) (see page 727)
- File Synonym Cross-Reference Report (DREPORT 021) (see page 729)
- Record Synonym Cross-Reference Report (DREPORT 022) (see page 729)
- Element Synonym Cross-Reference Report (DREPORT 023) (see page 730)
- Element Description Report (DREPORT 024) (see page 731)
- Element Designator Report (DREPORT 025) (see page 733)
- File Activity Report (DREPORT 026) (see page 734)
- IDMS Set Activity Report (DREPORT 027) (see page 735)
- IDMS Record Activity Report (DREPORT 028) (see page 736)
The cross-reference reports provide information on:

- Files and areas and their associated records (DREPORT 020)
- Files and their associated synonyms (DREPORT 021)
- Records and their associated synonyms (DREPORT 022)
- Elements and their associated synonyms (DREPORT 023)
- Elements with their associated descriptions (DREPORT 024)
- Element designators/attribute relationships (DREPORT 025)
- Program use of files, sets, records, and areas (DREPORTs 026, 027, 028, 029)
- Program/element relationships (DREPORT 030)

Summary of Cross-Reference Reports

The table below lists the cross-reference reports in order of presentation in this section.

<table>
<thead>
<tr>
<th>DREPORT Number</th>
<th>DREPORT Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>020</td>
<td>File/Record Cross-Reference Report(1)</td>
</tr>
<tr>
<td>021</td>
<td>File Synonym Cross-Reference Report(1)</td>
</tr>
<tr>
<td>022</td>
<td>Record Synonym Cross-Reference Report</td>
</tr>
<tr>
<td>023</td>
<td>Element Synonym Cross-Reference Report</td>
</tr>
<tr>
<td>024</td>
<td>Element Description Report</td>
</tr>
<tr>
<td>025</td>
<td>Element Designator Report</td>
</tr>
<tr>
<td>026</td>
<td>File Activity Report(1)</td>
</tr>
<tr>
<td>027</td>
<td>IDMS Set Activity Report</td>
</tr>
<tr>
<td>028</td>
<td>IDMS Record Activity Report</td>
</tr>
<tr>
<td>029</td>
<td>IDMS Area Activity Report</td>
</tr>
<tr>
<td>030</td>
<td>Element/Program Cross-Reference Report</td>
</tr>
</tbody>
</table>

(1) DREPORTs 020, 021, and 026 cannot be run with any other reports.

File/Record Cross-Reference Report (DREPORT 020)

Contents

The File/Record Cross-Reference report lists all IDD file/record relationships defined to the dictionary. The file/record relationships are defined through the RECORD SYNONYM FOR FILE SYNONYM clause or the WITHIN FILE clause of the DDDL RECORD statement.
Sample DREPORT 020:

<table>
<thead>
<tr>
<th>FILE NAME</th>
<th>RECORD LENGTH</th>
<th>BLOCK SIZE</th>
<th>LABELS</th>
<th>DATE UPDATED/CREATED</th>
<th>DESCRIPTION</th>
<th>PRIMARY RECORD/RECORD</th>
<th>FILE NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>TESTFILE</td>
<td>VER 1</td>
<td>mm/dd/yy</td>
<td>32</td>
<td>mm/dd/yy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>mm/dd/yy</td>
<td>132</td>
<td>mm/dd/yy</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Descriptions

A description of the fields in the sample report follows:

- **FILE NAME**
  Identifies the primary name and version number of an IDD file and displays a description if one is defined for this file occurrence.

- **RECORD LENGTH**
  Identifies the maximum record length for the file.

- **BLOCK SIZE**
  Identifies the block size of the file, if defined.

- **LABELS**
  Identifies any labels defined for the file.

- **DATE UPDATED/CREATED**
  Identifies the date the file occurrence was last updated and the date the file occurrence was defined to the dictionary.

- **Description**
  Identifies text associated with the file occurrence.

- **PRIMARY RECORD/RECORD**
  Identifies the names and version numbers of primary records associated with the file.

- **FILE NAME**
  Identifies the name and version number of a file associated with the named primary record and any text associated with this file occurrence.

- **Associated records and elements**
  Identifies records and elements that are associated with an IDD file occurrence.
File Synonym Cross-Reference Report (DREPORT 021)

Contents

The File Synonym Cross-Reference report lists all IDD file occurrences in the data dictionary with their associated file synonym names. File synonyms are defined through the FILE NAME SYNONYM clause of the DDDL FILE statement or through the schema compiler.

Sample DREPORT 021:

<table>
<thead>
<tr>
<th>REPORT NO.</th>
<th>mm/dd/yy</th>
<th>PAGE</th>
<th>DATA DICTIONARY REPORTER</th>
<th>REL nn.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DREPORT 021</td>
<td></td>
<td>1</td>
<td>FILE SYNONYM CROSS REFERENCE REPORT</td>
<td></td>
</tr>
</tbody>
</table>

************************************************************************************
********************************************
FILE SYNONYM NAME | PRIMARY FILE NAME
************************************************************************************
********************************************
TESTFILE | VER 1 | **SAME**
LHNFILE | VER 1 | TESTFILE

Field Descriptions

A description of the fields in the sample report follows:

- **FILE SYNONYM NAME**
  Identifies the name and version number of a file synonym.

- **PRIMARY FILE NAME**
  Identifies the primary file name and version number for this file synonym or displays **SAME** if the file synonym name is the primary file name.

Record Synonym Cross-Reference Report (DREPORT 022)

Contents

The Record Synonym Cross-Reference report lists all record occurrences defined to the dictionary with their associated record synonym names. Record synonyms are defined through the RECORD NAME SYNONYM clause of the DDDL RECORD statement and through the schema compiler.

Sample DREPORT 022:

<table>
<thead>
<tr>
<th>REPORT NO.</th>
<th>mm/dd/yy</th>
<th>PAGE</th>
<th>DATA DICTIONARY REPORTER</th>
<th>REL nn.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DREPORT 022</td>
<td></td>
<td>1</td>
<td>RECORD SYNONYM CROSS REFERENCE REPORT</td>
<td></td>
</tr>
</tbody>
</table>

************************************************************************************
********************************************
RECORD SYNONYM NAME | PRIMARY RECORD NAME | SCHEMA
************************************************************************************
********************************************
AD21D008-TEST-RECORD | VER 1 | **SAME**
Element Synonym Cross-Reference Report (DREPORT 023)

Field Descriptions

A description of the fields in the sample report follows:

- **RECORD SYNONYM NAME**
  Identifies the record synonym name and version number.

- **PRIMARY RECORD NAME**
  Identifies the primary name and version number for this synonym or displays **SAME** if the record synonym name is the primary record name.

- **SCHEMA**
  Identifies the schema that includes the record, if defined.

Element Synonym Cross-Reference Report (DREPORT 023)

Contents

The Element Synonym Cross-Reference report lists element occurrences defined to the dictionary with their associated element synonym names. Element synonyms are defined through the ELEMENT NAME SYNONYM clause of the DDDL ELEMENT or RECORD ELEMENT statements and through the schema compiler.

Sample DREPORT 023:

REPORT NO. 23               DATA DICTIONARY REPORTER   REL nn.
DREPORT 023 mm/dd/yy    PAGE  19        ELEMENT SYNONYM CROSS REFERENCE REPORT

************************************************************************************
********************************************
ELEMENT SYNONYM NAME          PRIMARY ELEMENT NAME
GROUP SYNONYM NAME

************************************************************************************

DENTIST-ZIP-LAST-FOUR **SAME** VER 1
Field Descriptions

A description of the fields in the sample report follows:

- **ELEMENT SYNONYM NAME**
  Identifies the name and version number of an element synonym.

- **PRIMARY ELEMENT NAME**
  Identifies the primary element name associated with this synonym name or displays **SAME** if the element synonym name is the primary element name.

- **GROUP SYNONYM NAME**
  Identifies the group synonym name associated with the element synonym, if defined.

Element Description Report (DREPORT 024)

Contents

The Element Description report lists element descriptions and all elements associated with each description. This report sorts elements by description. Element description text is defined to the dictionary through the ELEMENT DESCRIPTION IS clause of the DDDL ELEMENT statement.

Sample DREPORT 024:

```
REPORT NO. 24 mm/dd/yy PAGE 1
DATA DICTIONARY REPORTER REL nn.
ELEMENT DESCRIPTION REPORT

************************************************************************************
DESCRIPTION /DATA ELEMENT NAME VERSION LGTH PICTURE USAGE ELEMEN
SYNONYM NAME
************************************************************************************

CALCULATED BONUS
```
FIELD DESCRIPTIONS

A description of the fields in the sample report follows:

- **DESCRIPTION/DATA ELEMENT NAME**
  Identifies an element description and all elements with that description.

- **VERSION**
  Identifies the version number of the element.
Element Designator Report (DREPORT 025)

Contents

The Element Designator report lists all attributes within the element designator class and describes all elements associated with each element designator. Element designator is a system-provided class that allows you to classify similar elements for report purposes. Element/element designator relationships are defined through INCLUDE class IS attribute clause of the DDDL ELEMENT statement, where the class is ELEMENT DESIGNATOR.

Sample DREPORT 025:

<table>
<thead>
<tr>
<th>DESIGNATOR</th>
<th>DATA ELEMENT NAME</th>
<th>VERSION</th>
<th>LGTH</th>
<th>PICTURE</th>
<th>USAGE</th>
<th>ELEMENT SYNONYM NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOCUMENTATION</td>
<td>LHN-DEPT-NAME</td>
<td>VER 1 45 X</td>
<td>DISPLAY LHN-DEPT-NAME</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Descriptions

A description of the fields in the sample report follows:

- **DESIGNATOR**
  Identifies the name of an attribute associated with the element designator class.

- **DATA ELEMENT NAME**
  Identifies the names and version numbers of data elements associated with this element designator.

- **LGTH**
  Identifies the length in bytes of the element.
• PICTURE
  Identifies the picture clause associated with this data element.

• USAGE
  Identifies the usage mode for the element.

• ELEMENT SYNONYM NAME
  Identifies a synonym name associated with the element by the ELEMENT NAME SYNONYM clause of the DDDL ELEMENT or RECORD ELEMENT statements, or by the schema compiler.

File Activity Report (DREPORT 026)

Contents

The File Activity report provides information on how IDD files are used by programs. These file /program relationships are defined through the INPUT/I-O/OUTPUT clause of the DDDL PROGRAM statement and through the DML processors if the activity log is on and the dictionary is in UPDATE mode at run time.

Sample DREPORT 026:

FILE NAME          PROGRAM          USAGE   REFERENCED   SYNONYM NAME
TESTFILE            VER 1           LHNPROG VER 1 I-  

Field Descriptions

A description of the fields in the sample report follows:

• FILE NAME
  Identifies the primary name and version number of a file that is opened by a program.

• PROGRAM
  Identifies the name and version number of a program that accesses this file.

• USAGE
  Indicates whether the program opens the named file for input, output, or input/output operations.

• REFERENCED
  Identifies the number of OPEN statements for this file and usage within the named program.

• SYNONYM NAME
  Identifies the name used by the program to reference the file.
IDMS Set Activity Report (DREPORT 027)

Contents

The IDMS Set Activity report provides information on how programs reference sets. These set/program relationships are defined through the SET clause of the DDDL PROGRAM statement or through the DML processors if the activity log is on and the dictionary is in UPDATE mode at run time.

Sample DREPORT 027:

```
REPORT NO. 27 DATA DICTIONARY REPORTER REL nn.
DREPORT 027 DATA DICTIONARY REPORTER REL nn.
DREPORT 027 IDMS SET ACTIVITY REPORT

************************************************************************************
************************************************************************************
---- DATE ----
SCHEMA VER SUBSCHEMA SET PROGRAM VER USAGE
TIMES COMPILED CREATED
************************************************************************************
************************************************************************************
EMPSCHM 1 EMPSS01 DEPT-EMPLOYEE 1 OBTAIN 1 mm/dd/yy mm/dd
EMPSCHM 1 EMPSS01 DEPT-EMPLOYEE 1 OBTAIN 1 mm/dd/yy mm/dd
EMPSCHM 1 EMPSS01 DEPT-EMPLOYEE 1 OBTAIN 1 mm/dd/yy mm/dd
EMPSCHM 1 EMPSS01 DEPT-EMPLOYEE 1 OBTAIN 1 mm/dd/yy mm/dd
EMPSCHM 1 EMPSS01 DEPT-EMPLOYEE 1 OBTAIN 1 mm/dd/yy mm/dd
EMPSCHM 1 EMPSS01 DEPT-EMPLOYEE 1 OBTAIN 1 mm/dd/yy mm/dd
EMPSCHM 1 EMPSS01 DEPT-EMPLOYEE 1 OBTAIN 1 mm/dd/yy mm/dd
EMPSCHM 1 EMPSS01 DEPT-EMPLOYEE 1 OBTAIN 1 mm/dd/yy mm/dd
EMPSCHM 1 EMPSS01 DEPT-EMPLOYEE 1 OBTAIN 1 mm/dd/yy mm/dd
EMPSCHM 1 EMPSS01 DEPT-EMPLOYEE 1 OBTAIN 1 mm/dd/yy mm/dd
EMPSCHM 1 EMPSS01 DEPT-EMPLOYEE 1 OBTAIN 1 mm/dd/yy mm/dd
```

Field Descriptions

A description of the fields in the sample report follows:

- **SCHEMA/VER**
  Identifies the name and version number of a schema that contains a set referenced by a program.

- **SUBSCHEMA**
  Identifies the name of the subschema in which the set exists.
SET
Identifies the name of a set associated with a program.

PROGRAM/VER
Identifies the name and version number of the program using this set.

USAGE
Identifies the DML command issued against this set by the named program.

TIMES
Indicates the number of times this DML command is issued against the set within the named program.

DATE COMPILED/CREATED
Identifies the date the program was last compiled and the date the program occurrence was defined to the dictionary.

IDMS Record Activity Report (DREPORT 028)

Contents

The IDMS Record Activity report provides information on how programs use records. These record /program relationships are defined through the RECORD clause of the DDDL PROGRAM statement or through the DML processors if the activity log is on and the dictionary is in UPDATE mode at run time.

Sample DREPORT 028:

<table>
<thead>
<tr>
<th>SCHEMA</th>
<th>VER</th>
<th>SUBSCHEMA</th>
<th>RECORD</th>
<th>PROGRAM</th>
<th>VER</th>
<th>USAGE</th>
<th>TIMES</th>
<th>COMPILED</th>
<th>CREATED</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPSCHM</td>
<td>1</td>
<td>AD99SLR</td>
<td>COVERAGE</td>
<td>LRDA01D</td>
<td>1</td>
<td>BIND</td>
<td>1</td>
<td>mm/dd/yy</td>
<td>mm/dd/yy</td>
</tr>
<tr>
<td>EMPSCHM</td>
<td>1</td>
<td>AD99SLR</td>
<td>DEPARTMENT</td>
<td>LRDA01D</td>
<td>1</td>
<td>BIND</td>
<td>1</td>
<td>mm/dd/yy</td>
<td>mm/dd/yy</td>
</tr>
<tr>
<td>EMPSCHM</td>
<td>1</td>
<td>AD99SLR</td>
<td>EMPLOYEE</td>
<td>LRDA01D</td>
<td>1</td>
<td>BIND</td>
<td>1</td>
<td>mm/dd/yy</td>
<td>mm/dd/yy</td>
</tr>
<tr>
<td>EMPSCHM</td>
<td>1</td>
<td>AD99SLR</td>
<td>OFFICE</td>
<td>LRDA01D</td>
<td>1</td>
<td>BIND</td>
<td>1</td>
<td>mm/dd/yy</td>
<td>mm/dd/yy</td>
</tr>
<tr>
<td>EMPSCHM</td>
<td>1</td>
<td>EMPSSLR</td>
<td>DEPARTMENT</td>
<td>LRTD01D</td>
<td>1</td>
<td>BIND</td>
<td>1</td>
<td>mm/dd/yy</td>
<td>mm/dd/yy</td>
</tr>
<tr>
<td>EMPSCHM</td>
<td>1</td>
<td>EMPSSLR</td>
<td>EMPLOYEE</td>
<td>LRTD01D</td>
<td>1</td>
<td>BIND</td>
<td>1</td>
<td>mm/dd/yy</td>
<td>mm/dd/yy</td>
</tr>
<tr>
<td>EMPSCHM</td>
<td>1</td>
<td>EMPSSLR</td>
<td>EMPLOYEE</td>
<td>RENO01D</td>
<td>1</td>
<td>BIND</td>
<td>1</td>
<td>mm/dd/yy</td>
<td>mm/dd/yy</td>
</tr>
<tr>
<td>EMPSCHM</td>
<td>1</td>
<td>EMPSSLR</td>
<td>EMPLOYEE</td>
<td>RENO02D</td>
<td>1</td>
<td>BIND</td>
<td>1</td>
<td>mm/dd/yy</td>
<td>mm/dd/yy</td>
</tr>
</tbody>
</table>
Field Descriptions

A description of the fields in the sample report follows:

- **SCHEMA/VER**
  Identifies the name and version number of a schema that contains a record referenced by a program.

- **SUBSCHEMA**
  Identifies the subschema in which the record is included.

- **RECORD**
  Identifies a record associated with a program.

- **PROGRAM/VER**
  Identifies the name and version number of the program using the record.

- **USAGE**
  Identifies the DML command issued against the record by the named program.

- **TIMES**
  Indicates the number of times the DML command is issued against the record within the named program.

- **DATE COMPILED/Created**
  Identifies the date the program was last compiled and the date the program occurrence was defined to the dictionary.

IDMS Area Activity Report (DREPORT 029)

Contents

The IDMS Area Activity report provides information on how programs use areas. These area/program relationships are defined through the AREA clause of the DDDL PROGRAM statement or through the DML processors if the activity log is on and the dictionary is in UPDATE mode at run time.

Sample DREPORT 029:

```
REPORT NO. 29              DATA DICTIONARY REPORTER   REL nn.
DREPORT 029              IDMS AREA ACTIVITY REPORT

********************************************************************************
********************************************
---- DATE ----
SCHEMA   VER   SUBSCHEMA  AREA          PROGRAM   VER   ACTIVITY
TIMES   COMPILED  CREATED
********************************************************************************
********************************************
```

EMPSCHM  1   AD99SLR   EMP-DEMO-
REGION          LRDA01D  1   RETRIEVAL  3 mm/dd/yy mm/dd/yy
EMPSCHM  1   AD99SLR   INS-DEMO-
REGION          LRDA01D  1   RETRIEVAL  3 mm/dd/yy mm/dd/yy
### Field Descriptions

A description of the fields in the sample report follows:

- **SCHEMA/VER**
  Identifies the name and version number of the schema that contains the area referenced by the program.

- **SUBSCHEMA**
  Identifies the subschema in which the area exists.

- **AREA**
  Identifies the area associated with the named program.

- **PROGRAM/VER**
  Identifies the name and version number of the program using this area.

- **ACTIVITY**
  Identifies the DML command issued against this area by the named program.
Element/Program Cross-Reference Report (DREPORT 030)

Contents

The Element/Program Cross-Reference report lists all element/program relationships defined to the dictionary; all elements associated with a record that is referenced by a program are listed. Element/record and record/area relationships are defined automatically for elements, records, and areas stored in a CA IDMS/DB database. For non-database files, element/record and record/file relationships are defined through the DDDL syntax for elements.

Sample DREPORT 030:

<table>
<thead>
<tr>
<th>PRIMARY ELEMENT NAME</th>
<th>PROGRAM NAME</th>
<th>USAGE</th>
<th>PROGRAM ELEMENT NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE</td>
<td>VER 1 ADMI01D</td>
<td>VER 1 OBTAIN</td>
<td>ACTIVE-0415</td>
</tr>
<tr>
<td>VER 1 ADMI01D</td>
<td>VER 1 RETURN</td>
<td></td>
<td>ACTIVE-0415</td>
</tr>
<tr>
<td>VER 100 ADMS01D</td>
<td>VER 1 OBTAIN</td>
<td></td>
<td>ACTIVE-0415</td>
</tr>
<tr>
<td>VER 100 ADMS01D</td>
<td>VER 1 RETURN</td>
<td></td>
<td>ACTIVE-0415</td>
</tr>
<tr>
<td>VER 100 ADMS02D</td>
<td>VER 1 OBTAIN</td>
<td></td>
<td>ACTIVE-0415</td>
</tr>
<tr>
<td>VER 100 ADMS02D</td>
<td>VER 1 PROT UPD</td>
<td></td>
<td>ACTIVE-0415</td>
</tr>
<tr>
<td>VER 100 ADMS02D</td>
<td>VER 1 RETURN</td>
<td></td>
<td>ACTIVE-0415</td>
</tr>
</tbody>
</table>

Field Descriptions

A description of the fields in the sample report follows:

- **PRIMARY ELEMENT NAME**
  Identifies the primary name and version number of an element associated with a record used by a program.

- **PROGRAM NAME**
  Identifies the name and version number of the program that accesses the record that contains the element.

- **USAGE**
  Identifies the usage mode of the area associated with the element. If the element/program relationship is established through a CA IDMS/DB area, the READY mode of the CA IDMS/DB area.
is displayed. Possible values for the area READY mode are UPDATE, PROTECTED RETRIEVAL, EXCLUSIVE RETRIEVAL, and EXCLUSIVE UPDATE. If the element/program relationship is defined through a non-database file, the OPEN mode of the non-database file is displayed. Possible values for the file OPEN mode are INPUT, OUTPUT, and I-O. Programs that potentially change or reference an element can be identified by the OPEN mode of the file or by the READY mode of the area.

- **PROGRAM ELEMENT NAME**
  Identifies the element name used by the program to reference the element. The name can be the primary element name or an element synonym name.

### Special-Purpose Report Modules

**Contents**
- Comment/Nest Resolution Report Module (DREPORT 000) (see page 740)
- Level Number Report (DREPORT 050) (see page 741)
- Module Text to Card Utility (DREPORT 051) (see page 741)
- Module Text to File Utility (DREPORT 052) (see page 742)

There are four special-purpose report modules (DREPORTs 000, 050, 051, and 052). These report modules perform specific functions such as providing housekeeping parameters for the other reports, listing dictionary level numbers, punching module source text to cards, or writing source text to disk. DREPORT 050 is the only special-purpose report that produces printed output.

**Summary Table**

The following table lists the four special-purpose report modules in order of presentation in this section.

<table>
<thead>
<tr>
<th>DREPORT Module</th>
<th>DREPORT Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>Comment/Nest Resolution</td>
</tr>
<tr>
<td>050</td>
<td>Level Number Report</td>
</tr>
<tr>
<td>051</td>
<td>Module Text to Card Utility(1)</td>
</tr>
<tr>
<td>052</td>
<td>Module Text to Output File Utility(1)</td>
</tr>
</tbody>
</table>

**Note:** DREPORTs 051 and 052 must be run alone.

### Comment/Nest Resolution Report Module (DREPORT 000)

**Contents**

The comment/nest resolution report module performs internal housekeeping functions whenever a dictionary, DC/UCF system, CA ADS, or catalog report module is requested. CA Culprit for CA IDMS parameters for DREPORT 000 are copied the first time a DREPORT=, CREPORT=, or AREPORT= request parameter is encountered in the input stream. There is no printed output for this report module.
Level Number Report (DREPORT 050)

Contents

The level number report lists the current values of all level numbers established in the dictionary. Level numbers are assigned by the dictionary when the element is included in a record. Up to 48 levels of data elements can be established by using the LEVEL NUMBERS ARE clause of the DDDL SET OPTIONS statement.

The report below shows sample output for DREPORT 050. This report lists the 48 level numbers and their current values.

Sample DREPORT 050:

REPORT NO. 50 DATA DICTIONARY REPORTER REL nn.
DREPORT 050 mm/dd/yy PAGE 1 LEVEL NUMBER REPORT

************************************************************************************
********************************************
---- DATA ----
LEVEL NUMBERS ARE clause of the DDDL SET OPTIONS statement.
************************************************************************************

Field Descriptions

A description of the fields in the sample report follows:

- **LEVEL NUMBERS**
  Identifies the current values of level numbers within the dictionary. The first 24 levels are listed on line 1; the second 24 levels are listed on line 2.

- **DATE UPDATED/CREATED**
  Identifies the date the dictionary was last updated and the date the dictionary was defined.

Module Text to Card Utility (DREPORT 051)

The Module Text to Card utility lets you punch the module source code of a specified module to cards. The DREPORT and KEY parameters used to punch module DREPORT 054 to card are:

DREPORT=051
KEY MOD-NAME-067 ‘DREPORT 054’

How to Run DREPORT 051

To run DREPORT 051, include the following specification in the JCL:

- **For z/OS systems:**
Module Text to File Utility (DREPORT 052)

The module text to output file utility (DREPORT 052) lets you output module source code to a disk file. The DREPORT and KEY parameters used to output module DRPT054 to file are shown below:

DREPORT=052
   KEY MOD-NAME-067 ‘DREPORT 054’

How to Run DREPORT 052

To run DREPORT 052, add the following specification to the JCL:

- For z/OS files:

  //SYS020   DD   DSN=user.textfile,DISP=(NEW,CATLG),
  //                  DCB=(RECFM=FB,LRECL=80,BKSIZE=320),
  //                  UNIT=disk,VOL=SER=nnnnnn

- user.textfile
data set name of the output file

- disk
  symbolic device name of disk
- `nnnnnn` volume serial number of disk

- **For z/VSE tape files:**
  ```
  // TLBL SYS020,'user.text'
  ASSIGN SYS020,TAPE,VOL=nnnnnn
  ```

  - `user.text` file-id of tape file
  - `nnnnnn` tape volume serial number

- **For z/VSE disk files:**
  ```
  // DLBL SYS020,'user.text'
  // EXTENT SYS020,nnnnnn
  ASSIGN SYS020,DISK,VOL=nnnnnn,SHR
  ```

  - `user.text` file-id of disk file
  - `nnnnnn` volume serial number of the disk file

- **For z/VM and z/VM files:**
  ```
  FILEDEF SYS020 DISK nonprint file a (RECFM FB LRECL 80 BLKSIZE 320
  ```

  - `nonprint file a` filename, filetype, and filemode of the nonprint/nonpunch output file.
Task Analyzer Reports

CA IDMS Task Analyzer produces seven major types of log reports, including:

- **Billing Reports** -- Three Billing Reports relate CA IDMS/DC statistics to users. The CA IDMS Task Analyzer Billing Reports contain statistics from the CA IDMS Log (or optionally under z/OS, the SMF File) to assist your analysis of system resource use. Depending on the parameters you select, CA IDMS/DC task activity can be tied to a specific operator, terminal, task code, or group for CA IDMS/DC, CICS, z/VM, or batch transactions.
  - Billing Details Report
  - Billing Summary Report
  - Billing System Summary Report

- **Program Reports** -- Three Program Reports contain statistics from the CA IDMS Log (or optionally under z/OS, the SMF File) providing both detailed and summarized information on system and application programs. These reports provide statistics on terminal reads, writes, and errors; storage acquired, allocated, and kept; and scratch and queue usage.
  - Program Details Report
  - Program Summary Report
  - Program System Summary Report

- **CA ADS Reports** -- Three CA ADS Reports contain statistics from the CA IDMS Log (or optionally under z/OS, the SMF File) both detailed and summarized information on CA ADS dialogs. These reports provide statistics on dialog commands, processes, pageable map use, link levels, and record buffer blocks.
  - CA ADS Detail Report
  - CA ADS Summary Report
  - CA ADS System Summary Report

- **Abend Report** -- The Abend Report contains statistics from the CA IDMS Log (or optionally under z/OS, the SMF File) displaying tasks that abend under CA IDMS. Information on tasks that abend includes: the CA IDMS abend code, message number, and severity code.
  - Abend Report

- **Program Loads Report** -- The Program Loads Report contains statistics from the CA IDMS Log (or optionally under z/OS, the SMF File), and lists by task the primary program and all secondary programs called. Both the primary program and secondary programs are identified by name, version, and type; secondary programs are further identified by program, map, table, and subschema. The number of times each secondary program is called during each execution of the task is also presented.
Program Loads Report

**Integrated Index Report** -- Three Integrated Index Reports contain statistics from the CA IDMS Log (or optionally under z/OS, the SMF File) providing detailed and summarized information showing how the execution of CA ADS dialogs affects your current integrated index structure. Unlike the other CA IDMS Task Analyzer reports, the Integrated Index Reports provide information on database activity. The reports show how programs and CA ADS dialogs affect and use integrated indexes.

- Integrated Index Details Report
- Integrated Index Summary Report
- Integrated Index System Summary Report

**Ranking Report** -- The Ranking Report arranges statistics from the CA IDMS Log (or optionally under z/OS, the SMF File) to compare how your CA IDMS/DC tasks are using your system resources.

- Ranking Report

**Input Parameter Report** -- Generated dynamically, the Input Parameter Report lists all the parameters you supplied as well as any processing messages.

---

**Task Analyzer System Output**

For more information, see the following topics:

- CA IDMS Task Analyzer Reports (see page 746)
- About CA IDMS Task Analyzer Billing Reports (see page 746)
- CA IDMS Task Analyzer Billing Details Report (see page 747)
- CA IDMS Task Analyzer Billing Summary Report (see page 750)
- CA IDMS Task Analyzer Billing System Summary Report (see page 753)
- About CA IDMS Task Analyzer Program Reports (see page 755)
- CA IDMS Task Analyzer Program Details Report (see page 756)
- CA IDMS Task Analyzer Program Summary Report (see page 760)
- CA IDMS Task Analyzer Program System Summary Report (see page 763)
- About CA IDMS Task Analyzer CA ADS Reports (see page 765)
- CA IDMS Task Analyzer CA ADS Details Report (see page 766)
- CA IDMS Task Analyzer CA ADS Summary Report (see page 769)
- CA IDMS Task Analyzer CA ADS System Summary Report (see page 772)
- About CA IDMS Task Analyzer Abend Report (see page 775)
- CA IDMS Task Analyzer Abend Report (see page 775)
- About CA IDMS Task Analyzer Program Loads Report (see page 777)
- CA IDMS Task Analyzer Program Loads Report (see page 778)
- About CA IDMS Task Analyzer Integrated Index Reports (see page 781)
- CA IDMS Task Analyzer Integrated Index Details Report (see page 782)
CA IDMS - 19.0

- CA IDMS Task Analyzer Integrated Index Summary Report (see page 786)
- Integrated Index System Summary Report (see page 789)
- CA IDMS Task Analyzer Ranking Report (see page 791)
- CA IDMS Task Analyzer Input Parameter Report (see page 793)

CA IDMS Task Analyzer Reports

CA IDMS Task Analyzer produces seven types of log reports and an input parameter report. Use this section as a general reference to CA IDMS Task Analyzer reports and as a preview of them for selecting parameters.

- Billing
- Program
- CA ADS
- Abend
- Program Loads
- Integrated Index
- Ranking

CA IDMS Task Analyzer also produces an Input Parameter Report that lists all parameters input and processed and also presents a list of all messages that were generated during execution.

About CA IDMS Task Analyzer Billing Reports

Contents
- Tying Task Activity to an ID and a Time (see page 746)
- Three Reports One Set of CA IDMS Log or SMF File Statistics (see page 747)
- Hierarchical Nature of Reports (see page 747)
- Overview of Billing Reports (see page 747)

The Billing Reports use information from the CA IDMS Log (or, optionally under z/OS, the SMF file) to produce both detailed and summarized report statistics. CA IDMS Task Analyzer Billing Reports can serve as supplements to building an effective billing system in your environment. You will get this report by specifying REPORT = BILL on the report parameter statement.

Tying Task Activity to an ID and a Time

Depending on the parameters you choose, task activity can be tied to a specific user, a group of users, a terminal, or a task code. For CA IDMS/DC tasks, the data reported under the headings OPER-ID, TERM-ID, or TASK CODE comes from the CA IDMS Log (or, optionally under z/OS, the SMF file). For CICS tasks, this information is taken from the External Request Element (ERE) Extension; the information is available only if GSISVCX was installed.
The CA IDMS Task Analyzer Billing Reports present this information within the framework of the time interval you select. Task totals are shown; they are also reflected as a ratio of the system totals (that is, the percentage of all time units, system resources, and CA IDMS resources consumed during the specified time interval).

Three Reports One Set of CA IDMS Log or SMF File Statistics

Physically, there are three Billing Reports to choose from. However, it is important to understand that each report is produced from the statistics that are found on the CA IDMS Log (or, optionally under z/OS, the SMF file). Statistics in the reports are presented in various formats and at two levels of summarization. The Billing Report is available at the detail, summary, and system summary levels.

Hierarchical Nature of Reports

Billing Reports are produced on a hierarchical level: If you ask for the lowest level report (LEVL = DET), you will also receive the higher-level reports. These would include the Billing Summary Report, which summarizes the data of the Billing Details Report (LEVL = SUM), and also the Billing System Summary Report (LEVL = SYS).

Overview of Billing Reports

- Billing Details Report -- presents detailed information for each task, reported in start time sequence. Depending on the parameter combination selected, this report allows you to identify task activity by operator ID, terminal ID, task code, group ID, or for all tasks.
- Billing Summary -- records the sum of all tasks invoked by an operator ID, terminal ID, task code, or group ID within the time interval you selected.
- Billing System Summary -- presents a sum of all Billing Summaries within the time interval you selected.

CA IDMS Task Analyzer Billing Details Report

Contents
- Report Fields (see page 748)

The CA IDMS Task Analyzer Billing Details Report presents a detailed view of the activity of each task activity reported in time sequence, based on the parameters selected. Depending on the parameter combination you select, this report allows you to identify task activity by operator ID, terminal ID, task code, group ID, or for all tasks.

Detailed information on CA ADS dialogs is presented within separate reports. For more information, see the Overview of CA ADS Reports.

You will get the Billing Details Report only if you specify LEVL = DET. In addition, with this specification, you will receive the Billing Summary Report and the Billing System Summary Report.
Report Fields

Here is a description of the various fields that make up the CA IDMS Task Analyzer Billing Details Report. The following Billing Details Report screen shows two possible types of Billing Details Reports CA IDMS Task Analyzer will generate.

REPORT TITLE -- The title line of this report varies depending on what you select on the RUNAME and NAME parameter statements.

REQUESTED TIME INTERVAL -- This line lists the start and stop date and time of the time interval you specified on the PROCESS statement. The data displayed in this line depends on what you select using the START and STOP parameters.

ACTUAL INTERVAL -- This line lists the first start date and time and the last start date and time on the CA IDMS Log or SMF File reported on. The data displayed in this line depends upon the actual time range of task activity within the CA IDMS Log or SMF File.

VARIABLE COLUMNS -- These two columns vary in content. Any combination of OPER-ID, TERM-ID, or TASK CODE can appear in the first two columns. (RUNAME types that do not appear in the REPORT TITLE will appear in these two columns.)

TASK INFORMATION

- NUMBER -- Number of the task within the date and time interval selected.
- VER -- Version of the task. Multiple versions of a task are reported separately.
- TY -- Type of task performed, indicating the language of the program the task invokes.
  - A: Assembler
  - C: COBOL
  - N: CA ADS
  - P: PL/1
  - F: Fortran
- OR -- Origin of the task. The operating system or environment where execution of the task originated.
  - D: CA IDMS/DC
  - C: CICS
  - V: z/VM
  - B: batch
- START DATE-TIME -- The start date and time of the task being reported.
- **C C** -- Condition code for CA IDMS. If the task abends while running under CA IDMS, an "X" appears on the report under "C C". If the task does not abend while running under CA IDMS, the "C C" column is blank.

**TOTAL TIME**

- **WALL CLOCK** -- The total elapsed real time, in seconds, reported to the nearest 1/10,000 second.
- **WAIT** -- Wait time, in seconds, reported to the nearest 1/10,000 second; this is the total idle time during the processing of the reported task, when no CPU time is used by either CA IDMS or the programs that make up the task.
- **SYSTEM** -- System time, in seconds, reported to the nearest 1/10,000 second; this is CPU time used by CA IDMS to process the reported task.
- **USER** -- User time, in seconds, reported to the nearest 1/10,000 second; this is CPU time used by the user's programs that make up the reported task.
- **TOTAL CPU** -- Total CPU time used by the task, in seconds, reported to the nearest 1/10,000 second.

**TOTAL I/O** -- Total number of database input/output operations performed by the programs that make up the reported task.

**TOTAL DB CALLS** -- Total number of DML verbs executed: this is the total number of calls issued to the database by programs that make up the reported task.

**TOTAL STOR ALLOC** -- Total amount of storage (in bytes) allocated for the reported task.

**TOTAL TERM I/O** -- Total number of terminal input/output operations performed by the programs that make up the reported task.

**CV NUMBER** -- The number of the CV that the statistics on this report apply to.

**PLAN ID** -- The statistics plan ID that the statistics on this report apply to.

<table>
<thead>
<tr>
<th>ID</th>
<th>TASK ANALYZER</th>
<th>RELEASE DATE</th>
<th>CA-IDMS TIME</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn /yy</td>
<td>Rnn. BILLING REPORT</td>
<td>mm/dd</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**S REQUESTED:** mm/dd/yyyy hh/mm - mm/dd/yyyy hh/mm

**ACTUAL:** mm/dd/yyyy hh/mm - mm/dd/yyyy hh/mm

<table>
<thead>
<tr>
<th>TASK TERM</th>
<th>---------------</th>
<th>TASK</th>
<th>---------------</th>
<th>---------------</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CODE</th>
<th>ID</th>
<th>VER</th>
<th>START DATE-TIME C</th>
<th>WALL</th>
<th>WAIT</th>
<th>SYSTEM</th>
<th>USER</th>
<th>TOTAL I/O</th>
<th>CALLS</th>
<th>STOR</th>
<th>TERM</th>
</tr>
</thead>
<tbody>
<tr>
<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>CPU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TERMINAL REQUEST** $$ERUS$$ 928

| 13 | 7.3436 | 5.1182 | 2.2313 | 0.0000 | 2.2313 | 280 | 830 | 2092 |

**TOTAL**: 749/898
The CA IDMS Task Analyzer Billing Summary Report summarizes all tasks executed for an operator ID, terminal ID, task code, or group ID within the time interval you select. It is a summary of information from the CA IDMS Task Analyzer Billing Details Report.

You will get this report if you specify LEVL = SUM (or if you specify LEVL = DET). In addition to the Billing Summary Report, you will also receive the Billing System Summary Report.

When you look at the Billing Summary Report, focus on the % OF SYSTEM OCCURRENCES in the last column, as shown in the example in Billing Summary Report screen. These statistics reveal trends on the use of your CA IDMS/DC environment. HIGH VALUE, LOW VALUE, MEAN VALUE, and ACCUMULATED VALUE are reported for each statistical category.

Report Fields

Here is an explanation of the fields that make up the CA IDMS Task Analyzer Billing Summary Report.

REPORT TITLE -- The title line of this report varies depending on what you selected on the RUNAME and NAME parameter statements.

REQUESTED TIME INTERVAL -- This line lists the start and stop date/time of the time interval you specified on the PROCESS statement. The data displayed in this line depends on what you select using the START and STOP parameters.

ACTUAL INTERVAL -- This line lists the first start date and time and the last start date and time on the CA IDMS Log or SMF File reported on. The data displayed in this line depends upon the actual time range of task activity within the CA IDMS Log or SMF File.

SUMMARY FOR TASK CODE -- Task code identification information; this includes task name, origin of execution, and version number. Multiple versions of a task are reported separately; tasks with multiple origins of execution are also reported separately.
TOTAL RUN UNITS -- The total number of occurrences of the reported task (within the time interval selected).

TOTAL ABENDS -- The total number of abends that occurred as a result of processing the programs that make up the reported task (within the date and time interval selected).

SYSTEM RESOURCES -- as reported within the Billing Details Report.

- WALL CLOCK TIME -- The total elapsed real time, in seconds, reported to the nearest 1/10,000 second.

- CPU TIME -- Total CPU time used by the task, in seconds, reported to the nearest 1/10,000 second; this is the sum of SYSTEM time and USER time.

- SYSTEM TIME -- System time, in seconds, reported to the nearest 1/10,000 second; this is CPU time used by CA IDMS to process the reported task.

- USER TIME -- User time, reported to the nearest 1/10,000 second; this is CPU time used by the user's programs to perform the reported task.

- WAIT TIME -- Wait time, in seconds, reported to the nearest 1/10,000 second; this is the total idle time during the processing of the reported task, when no CPU time is used by either CA IDMS or the programs that make up the task.

- I/O -- Total number of database input/output operations performed by the programs that make up the reported task.

- DATABASE CALLS -- Total number of DML verbs executed: this is the total number of calls issued to the database by programs that make up the reported task.

- STORAGE ALLOCATED -- Total amount of storage allocated (in bytes) for the reported task.

- TERMINAL I/O -- Total number of terminal input/output operations performed by the programs that make up the reported task.

HIGH VALUE -- The highest value for each of the SYSTEM RESOURCES (depending upon the line) for the reported task.

TASK CODE -- The task code of the task with the highest value of the SYSTEM RESOURCES (depending upon the line).

TASK NUM -- Task number; this is the version number of the task with the highest value of the SYSTEM RESOURCES.

LOW VALUE -- The lowest value for each of the SYSTEM RESOURCES for the reported task.

TASK CODE -- The task code of the task with the lowest value of the SYSTEM RESOURCES (depending upon the line).

TASK NUM -- Task number; this is the version number of the task with the lowest value of the SYSTEM RESOURCES.

MEAN VALUE -- Average value per task occurrence within the reported SYSTEM RESOURCES.
**ACCUM VALUE** -- Total value for all task occurrences for the SYSTEM RESOURCES (depending upon the line) within the reported time interval.

**% OF SYSTEM OCCURRENCES** -- This ratio (expressed as a percentage) is the accumulated value for this task against the accumulated value for all system tasks within the reported time interval. This ratio highlights the tasks that are consuming the largest amount of system resources.

**SUMMARY FOR ALL TASKS** -- When the task has two or more versions or origins of execution, or both, a SUMMARY FOR ALL TASKS is produced. The summary lists high, low, mean, and accumulated values, as well as percent of system occurrences, for all tasks as identified by version and origin of execution.

<table>
<thead>
<tr>
<th>ID</th>
<th>RELEASE</th>
<th>DATE</th>
<th>CA-IDMS TIME</th>
<th>BILLING REPORT</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>/yy</td>
<td>hh:mm:ss</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SUMMARY FOR ALL TASKS**

REQUESTED: mm/dd/yy hh/mm - mm

ACTUAL: mm/dd/yy hh/mm - mm

<table>
<thead>
<tr>
<th>TASK NUMBER</th>
<th>TASK CODE</th>
<th>TASK NUMBER</th>
<th>TASK CODE</th>
<th>TASK VALUE</th>
<th>TASK VALUE</th>
<th>% OF SYSTEM OCCURRENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIGH</td>
<td>TASK</td>
<td>LOW</td>
<td>TASK</td>
<td>MEAN VALUE</td>
<td>ACCUM VALUE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NUMBER</td>
<td>VALUE</td>
<td>NUMBER</td>
<td>VALUE</td>
<td>VALUE</td>
<td></td>
</tr>
</tbody>
</table>

**SUMMARY FOR TASK CODE SGSJARPT ORIGIN BTCH VERSION 0**

TOTAL RUN UNITS 11 0.57

TOTAL ABENDS 0 0.00

WALL CLOCK TIME 15.673564 172.4092 2.65 2.65 1.0389 1.0389

CPU TIME 6.1585

SYSTEM TIME 40.8445 2.71 2.71 2.71

DATABASE CALLS 947

STORAGE ALLOCATED 20928

TERMINAL I/O 0

SUMMARY FOR ALL TASKS

TOTAL RUN UNITS 11 0.57

TOTAL ABENDS 0 0.00

WALL CLOCK TIME 21.9847 21.9847 0.00 0.00 7.3436 7.3436

CPU TIME 6.1585

SYSTEM TIME 6.1585
CA IDMS Task Analyzer Billing System Summary Report

Contents

- Report Fields (see page 753)

The CA IDMS Task Analyzer Billing System Summary Report presents a sum-total of all Billing Summaries within the time interval you select. All statistical categories are reported within run-unit origin: CA IDMS/DC, CICS, z/vm, and BATCH; or ALL. The report shows actual accumulated values for RUN UNITS, ABENDS, and SYSTEM RESOURCES and presents the percentage that each value is of total system resources.

You will get this report by specifying LEVL = SYS. (This report will also be created if you specify LEVL = DET or LEVL = SUM.)

When you review this report, focus on % of SYSTEM OCCURRENCES in the last column, as shown in System Summary Report screen. These statistics reveal trends on the use of your CA IDMS/DC environment. HIGH VALUE, LOW VALUE, MEAN VALUE, and ACCUMULATED VALUE are reported for each statistical category.

Report Fields

Here is a description of the various fields that make up the System Summary Report.

REPORT TITLE -- The title line of this report varies depending on what you selected on the RUNAME and NAME parameter statements.

REQUESTED TIME INTERVAL -- This line lists the start and stop date/time of the time interval you specified on the PROCESS statement. The data displayed in this line depends on what you select using the START and STOP parameters.

ACTUAL INTERVAL -- This line lists the first start date and time and the last start date and time on the CA IDMS Log or SMF File reported on. The data displayed in this line depends upon the actual time range of task activity within the CA IDMS Log or SMF File.

SUMMARY FOR ALL TASKS -- This line indicates that a summarization of statistics follows.

TOTAL RUN UNITS -- The total number of run-units performed by programs that make up the reported task (within the time interval selected).
TOTAL ABENDS -- The total number of abends that occurred as a result of processing the programs that make up the reported task (within the date and time interval selected).

SYSTEM RESOURCES: The CA IDMS/DC statistics are reported. For more information on the CA IDMS/DC statistics, see the CA IDMS Task Analyzer Billing Details Report.

- WALL CLOCK TIME
- CPU TIME
- SYSTEM TIME
- USER TIME
- WAIT TIME
- I/O
- DATABASE CALLS
- STORAGE ALLOCATED
- TERMINAL I/O

HIGH VALUE -- The highest value for each of the SYSTEM RESOURCES (depending upon the line) for the reported task.

TASK CODE -- The ID of the task with the highest value of the SYSTEM RESOURCES (depending upon the line).

TASK NUM -- Task number; this is the number of the task with the highest value of the SYSTEM RESOURCES.

LOW VALUE -- The lowest value for each of the SYSTEM RESOURCES for the reported task.

TASK CODE -- The ID of the task with the lowest value of the SYSTEM RESOURCES (depending upon the line).

TASK NUM -- Task number; this is the number of the task with the highest value of the SYSTEM RESOURCES.

MEAN VALUE -- Average value per program occurrence within the reported SYSTEM RESOURCES.

ACCUM VALUE -- Total value for all program occurrences for the SYSTEM RESOURCES (depending upon the line) within the reported time interval.

% OF SYSTEM OCCURRENCES -- This ratio (expressed as a percentage) is the accumulated value for this program against the accumulated value for all selected programs active within the reported time interval. This ratio highlights the programs that are consuming the largest amount of system resources.
CA IDMS - 19.0

比利宾总结

要求:

实际:

任务

平均

累计

%系统

发生次数

总结

所有任务

总运行单位

11 0.5

总出错

0 0.0

墙钟时间

21.9847 CMPCARLA 3139 7.3436 KENNIRPT

928 15.673564 172.4092 2.6

CPU时间

6.1585 CMPCARLA 3139 1.0389 KENNIRPT

1716 3.713136 40.8445 2.7

系统时间

6.1585 CMPCARLA 3139 1.0389 KENNIRPT

1716 3.713136 40.8445 4.3

用户时间

0.0000 KENNIRPT 928 .

等待时间

18.8502 SGSJARPT 2072 5.1182 KENNIRPT

928 12.046445 132.5109 2.6

I/O

614 CMPCARLA 3139 209 KENNIRPT 1

716 428.64 4715 11.8

数据库调用

2522 CMPCARLA 3139 325 KENNIRPT

1716 1425.09 15676 5.3

存储分配

20928 KENNIRPT 928 20928 CMPCARLA

3142 20928.00 230208 0.2

终端I/O

0.00 0.000000 0.0000 0.0


关于CA IDMS任务分析器程序报表

内容

- 三个报告一组CA IDMS日志或SMF文件统计（见第756页）
- 级联性质的报告（见第756页）
- 程序报告概述（见第756页）

CA IDMS任务分析器程序报表与CA IDMS任务分析器计费报表相似。它们使用CA IDMS日志（或在z/OS下，从SMF文件）的信息来提供详细和总结的统计报告，显示应用程序如何有效地使用系统资源。

它们的区别在于它们是按照组成任务的程序来组织的。信息包括程序的名称、版本、类型和执行的起源。首先，每个程序的总和被报告；然后，它们以系统总和的百分比表示（即，所有系统资源在给定的时间区间内被选中的程序所消耗的百分比）。您会得到这个报告通过指定REPORT = PROG。
Three Reports One Set of CA IDMS Log or SMF File Statistics

Physically, there are three Program Reports to choose from. It is important to understand, however, that each report is produced from the same statistics taken from the CA IDMS Log (or, optionally under z/OS, from the SMF file). Statistics are presented in different formats and at various levels of summarization. The Program Report is available at the detail, summary, and system summary level.

Hierarchical Nature of Reports

Program reports are produced for three hierarchical levels: If you ask for the lowest level report (LEVL = DET), you will also receive the higher-level reports. This would include the Program Summary Report, which summarizes the data of the Program Details Reports (LEVL = SUM), and also the Program System Summary Report (LEVL = SYS).

Overview of Program Reports

Program Reports are of the following types:

- Program Details Report -- shows the information for programs that make up each task reported in time sequence, within the selected time interval. You may specify which program or class of programs CA IDMS Task Analyzer is to report on. You also control the time period.

- Program Summary Report -- records the sum for all tasks that use the application program within the time interval you select. In addition, if multiple versions of a reported task exist, a summary of all tasks by version is presented.

- Program System Summary -- presents a sum of all Program Summaries within the time interval you selected.

CA IDMS Task Analyzer Program Details Report

Contents

- Report Fields (see page 756)

The CA IDMS Task Analyzer Program Details Report presents a detailed view of application program CA IDMS/DC activity. (CA ADS dialog activity is reported separately. For more information, see the CA IDMS Task Analyzer CA ADS Report.) The statistics in the report are CA IDMS STATISTICS taken from the CA IDMS Log (or optionally under z/OS, from the SMF file). You will get this report when you request LEVL = DET.

Request this report after you have used the Billing Report to identify tasks that require large amounts of storage or that perform a large number of I/O operations.

Report Fields

The following is a description of the various fields that make up the CA IDMS Task Analyzer Program Details Report (see the Program Detail Report screen).

REPORT TITLE -- The title line of this report varies depending on what you select on the RUNAME and NAME parameter statements.
REQUESTED TIME INTERVAL -- This line lists the start and stop date/time of the time interval you specified on the PROCESS statement. The data displayed in this line depends on what you select using the START and STOP parameters.

ACTUAL INTERVAL -- This line lists the first start date and time and the last start date and time on the CA IDMS Log or SMF File reported on. The data displayed in this line depends upon the actual time range of task activity within the CA IDMS Log or SMF File.

VARIABLE COLUMNS -- These two columns vary in content. Any combination of OPER-ID, TERM-ID, or TASK CODE can appear in the first two columns. (RUNAME types that do not appear in the REPORT TITLE will appear in these two columns.)

TASK INFORMATION

- NUMBER -- Number of the task within the date and time interval selected.
- TY -- Type of task performed, indicating the language of the program the task invokes.
  - A: Assembler
  - P: PL/1
  - C: COBOL
  - F: FORTRAN
  - N: CA ADS
- OR -- Origin of the task. The operating system or environment where execution of the task originated.
  - D: CA IDMS/DC
  - C: CICS
  - V: z/VM
  - B: batch
- START DATE-TIME -- The start date and time of the task being reported.
- C C -- Condition code for CA IDMS. If the task abends while running under CA IDMS, an "X" appears on the report under "C C". If the task does not abend while running under CA IDMS, the "C C" column is blank.

PROGRAM INFORMATION

- NAME -- Name of the reported program.
- VER -- Version of the program.
- TY -- Type of program performed, according to the language that the program consists of (see the list under TASK INFORMATION).
TERMINAL INFORMATION

- **RDS** -- The total number of reads performed from the reported terminal.
- **WRT** -- The total number of writes performed from the reported terminal.
- **ERR** -- The total number of errors occurring during reads and writes from the reported terminal.

**STORAGE INFORMATION** -- These columns give you an idea of how much system storage is required for the reported tasks.

- **ACQ** -- The number of requests made to acquire system storage.
- **ALLOC** -- The total amount of storage requested (in bytes).
- **KEPT** -- The total amount of storage (in bytes) retained by the program and not released for reuse.

**SCRATCH INFORMATION**

- **GET** -- The number of times records were retrieved from the scratch area.
- **PUT** -- The number of times records were placed into the scratch area.
- **DELETE** -- The number of times records were deleted from the scratch area.

**QUEUE INFORMATION**

- **GET** -- The number of times records were retrieved from the queue area.
- **PUT** -- The number of times records were placed into the queue area.
- **DELETE** -- The number of times records were deleted from the queue area.

* -- Programs loaded and executed by the previous task (see the Program Details Report).

**CV NUMBER** -- The number of the CV that the statistics on this report apply to.

**PLAN ID** -- The statistics plan ID that the statistics on this report apply to.

<table>
<thead>
<tr>
<th>ID</th>
<th>TASK ANALYZER</th>
<th>RELEASE DATE</th>
<th>TIME</th>
<th>CA-IDMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn</td>
<td>mm:ss</td>
<td>PROGRAM REPORT</td>
<td>mm/dd/yy</td>
<td>hh:mm:ss</td>
</tr>
</tbody>
</table>

**CV NUMBER:** 19  
**PLAN ID:** PLAN0001  
**DETAILS FOR TASK CODE**

<table>
<thead>
<tr>
<th>/yy hh/mm - mm/dd/yy hh/mm</th>
<th>REQUESTED: mm/dd</th>
</tr>
</thead>
<tbody>
<tr>
<td>/yy hh/mm - mm/dd/yy hh/mm</td>
<td>ACTUAL: mm/dd</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TASK</th>
<th>TERMINAL</th>
<th>CODE</th>
<th>ID</th>
<th>NUMBER</th>
<th>START DATE</th>
<th>NAME</th>
<th>VER</th>
<th>T</th>
<th>RDS</th>
<th>WRT</th>
<th>ERR</th>
<th>ACQ</th>
<th>ALLOC</th>
<th>KEPT</th>
<th>GET</th>
<th>PUT</th>
<th>DEL</th>
<th>GET</th>
<th>PUT</th>
<th>DEL</th>
</tr>
</thead>
</table>

16-Jan-2018 758/898
<table>
<thead>
<tr>
<th>Y</th>
<th>R</th>
<th>C</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>RHDCSGDC</td>
<td>VTAMLT05</td>
<td>A</td>
</tr>
<tr>
<td>39</td>
<td>RHDCSGDC</td>
<td>VTAMLT05</td>
<td>A</td>
</tr>
<tr>
<td>53</td>
<td>RHDCSGDC</td>
<td>VTAMLT05</td>
<td>A</td>
</tr>
<tr>
<td>01</td>
<td>RHDCSGDC</td>
<td>VTAMLT05</td>
<td>A</td>
</tr>
<tr>
<td>03</td>
<td>RHDCSGDC</td>
<td>VTAMLT05</td>
<td>A</td>
</tr>
<tr>
<td>15</td>
<td>TRCREPD1</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>*</td>
<td>TRCREPD1</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>7</td>
<td>TRCREPD1</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>24</td>
<td>ADST</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>7</td>
<td>26752</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>13</td>
<td>TRCREPD1</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>1</td>
<td>17920</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>3</td>
<td>18496</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>57</td>
<td>ADST</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>11</td>
<td>18496</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>10</td>
<td>26752</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>51</td>
<td>TRCREPD1</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>24</td>
<td>28800</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>59</td>
<td>TRCREPD1</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>32</td>
<td>28800</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>4</td>
<td>19840</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>16</td>
<td>19840</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>51</td>
<td>TRCREPD1</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>24</td>
<td>19840</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>4</td>
<td>19072</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>09</td>
<td>TRCREPD1</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>12</td>
<td>19072</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>1</td>
<td>19304</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>17</td>
<td>TRCREPD1</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>1</td>
<td>11072</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>3</td>
<td>18432</td>
<td>VTAMLT03</td>
<td>N</td>
</tr>
<tr>
<td>49</td>
<td>ADSOGEN1</td>
<td>VTAMLT04</td>
<td>A</td>
</tr>
<tr>
<td>49</td>
<td>ADSOGEN1</td>
<td>VTAMLT04</td>
<td>A</td>
</tr>
</tbody>
</table>
CA IDMS Task Analyzer Program Summary Report

Contents

- Report Fields (see page 760)

The CA IDMS Task Analyzer Program Summary Report summarizes all executions of the first application program within the reported task within a time interval. The CA IDMS Task Analyzer Program Summary Report summarizes information taken from the CA IDMS Task Analyzer Program Details Report. You will get this report by specifying LEVL = SUM (or LEVL = DET).

When you look at the CA IDMS Task Analyzer Program Summary Totals Report, focus on the high, low, mean, and accumulated values, and percent (%) of system occurrences. These statistics reveal trends on the efficiency and use of your application programs.

Report Fields

Here is an explanation of the fields that make up the Program Summary Report (see the following Program Summary Report screen).

REPORT TITLE -- The title line of this report varies depending on what you selected on the RUNAME and NAME parameter statements.

REQUESTED TIME INTERVAL -- This line lists the start and stop date/time of the time interval you specified on the PROCESS statement. The data displayed in this line depends on what you select using the START and STOP parameters.

ACTUAL INTERVAL -- This line lists the first start date and time and the last start date and time on the CA IDMS Log or SMF File reported on. The data displayed in this line depends upon the actual time range of task activity within the CA IDMS Log or SMF File.

SUMMARY FOR TASK CODE -- Task code identification information; this includes task name, origin of execution, and version number. Multiple versions of a task are reported separately; tasks with multiple origins of execution are also reported separately.

TOTAL RUN UNITS -- The total number of run-units performed by programs that make up the reported task (within the time interval selected).

TOTAL ABENDS -- The total number of abends that occurred as a result of processing the programs that make up the reported task (within the date and time interval selected).

SYSTEM RESOURCES: The CA IDMS/DC statistics are reported. For a detailed explanation of the CA IDMS/DC statistics, see the CA IDMS Task Analyzer Program Details Report.

- TERMINAL READS

- TERMINAL WRITES

- TERMINAL ERRORS
- STORAGE ACQUIRED
- STORAGE ALLOCATED
- STORAGE KEPT
- SCRATCH-GETS
- SCRATCH-PUTS
- SCRATCH DELETES
- QUEUE-GETS
- QUEUE-PUTS
- QUEUE-DELETES

**HIGH VALUE** -- The highest value for each of the SYSTEM RESOURCES (depending upon the line) for the reported task.

**TASK CODE** -- The ID of the task with the highest value of the SYSTEM RESOURCES (depending upon the line).

**TASK NUMBER** -- Task number; this is the number of the task with the highest value of the SYSTEM RESOURCES.

**LOW VALUE** -- The lowest value for each of the SYSTEM RESOURCES for the reported task.

**TASK CODE** -- The ID of the task with the lowest value of the SYSTEM RESOURCES.

**TASK NUMBER** -- Task number; this is the number of the task with the highest value of the SYSTEM RESOURCES.

**MEAN VALUE** -- Average value per program occurrence within the reported SYSTEM RESOURCES.

**ACCUM VALUE** -- Total value for all program occurrences for the SYSTEM RESOURCES within the reported time interval.

**% OF SYSTEM OCCURRENCES** -- This ratio (expressed as a percentage) is the accumulated value for this program against the accumulated value for all selected programs active within the reported time interval. This ratio highlights the programs that are consuming the largest amount of system resources.

<table>
<thead>
<tr>
<th>ID / TASK ANALYZER / yy</th>
<th>RELEASE / Rnn.nn</th>
<th>DATE / hh:mm:ss</th>
<th>CA-IDMS / PROGRAM REPORT</th>
<th>TIME / mm/dd</th>
<th>SUMMARY FOR ALL TASKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQUESTED: mm/dd/yy</td>
<td>requested</td>
<td>mm/dd/yy</td>
<td>hh/mm - mm/dd/yy</td>
<td>hh/mm</td>
<td>mm/dd/yy</td>
</tr>
<tr>
<td>ACTUAL: mm/dd/yy</td>
<td>actual</td>
<td>mm/dd/yy</td>
<td>hh/mm - mm/dd/yy</td>
<td>hh/mm</td>
<td>mm/dd/yy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SYSTEM</th>
<th>HIGH TASK CODE NUMBER VALUE</th>
<th>LOW TASK CODE NUMBER VALUE</th>
<th>TASK MEAN VALUE</th>
<th>ACCUM VALUE</th>
<th>% OF OCCUR</th>
</tr>
</thead>
</table>

16-Jan-2018 761/898
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>OEDR0001</td>
<td></td>
<td></td>
<td>1</td>
<td>40</td>
<td>2.0</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>56</td>
<td>28480</td>
<td>52480</td>
<td>21952</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>OEDR0001</td>
</tr>
<tr>
<td>OEDR0002</td>
<td></td>
<td></td>
<td>1</td>
<td>61</td>
<td>3.1</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>18</td>
<td>28672</td>
<td>52480</td>
<td>21888</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>OEDR0002</td>
</tr>
<tr>
<td>OLM</td>
<td></td>
<td></td>
<td>1</td>
<td>858</td>
<td>44.7</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>858</td>
<td>20672</td>
<td>28672</td>
<td>21888</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>OLM</td>
</tr>
</tbody>
</table>

16-Jan-2018
The CA IDMS Task Analyzer Program System Summary Report presents a sum-total of all Program Summaries within the time interval you select. All statistical categories are reported within run-unit origin: CA IDMS/DC, CICS, z/VM, and BATCH; or ALL. The report shows actual accumulated values for RUN UNITS, ABENDS, and SYSTEM RESOURCES, and presents the percentage that each value is of total system resources.

You will get this report by specifying LEVL = SYS. (This report will also be created if you specify LEVL = DET or LEVL = SUM.)

When you review this report, focus on % of SYSTEM OCCURRENCES in the last column, as shown in Program System Summary Report screen. These statistics reflect trends on the use of your CA IDMS/DC environment. HIGH VALUE, LOW VALUE, MEAN VALUE, and ACCUMULATED VALUE are reported for each statistical category.

Report Fields

Here is a description of the various fields that make up the Program System Summary Report.

REPORT TITLE -- The title line of this report varies depending on what you selected on the RUNAME and NAME parameter statements.

REQUESTED TIME INTERVAL -- This line lists the start and stop date/time of the time interval you specified on the PROCESS statement. The data displayed in this line depends on what you select using the START and STOP parameters.

ACTUAL INTERVAL -- This line lists the first start date and time and the last start date and time on the CA IDMS Log or SMF File reported on. The data displayed in this line depends upon the actual time range of task activity within the CA IDMS Log or SMF File.

SUMMARY FOR ALL REQUESTED ENTITIES -- This line indicates that a summarization of statistics follows.

TOTAL RUN UNITS -- The total number of run-units performed by programs that make up the reported task (within the time interval selected).
TOTAL ABENDS -- The total number of abends that occurred as a result of processing the programs that make up the reported task (within the date and time interval selected).

SYSTEM RESOURCES: The CA IDMS/DC statistics are reported. For more information on the CA IDMS/DC statistics, see the CA IDMS Task Analyzer Program Details Report.

- TERMINAL READS
- TERMINAL WRITES
- TERMINAL ERRORS
- STORAGE ACQUIRED
- STORAGE ALLOCATED
- STORAGE KEPT
- SCRATCH-GETS
- SCRATCH-PUTS
- SCRATCH-DELETES
- QUEUE-GETS
- QUEUE-PUTS
- QUEUE-DELETES

HIGH VALUE -- The highest value for each of the SYSTEM RESOURCES (depending upon the line) for the reported task.

TASK CODE -- The ID of the task with the highest value of the SYSTEM RESOURCES (depending upon the line).

TASK NUMBER -- Task number; this is the number of the task with the highest value of the SYSTEM RESOURCES.

LOW VALUE -- The lowest value for each of the SYSTEM RESOURCES for the reported task.

TASK CODE -- The ID of the task with the lowest value of the SYSTEM RESOURCES (depending upon the line).

TASK NUMBER -- Task number; this is the number of the task with the highest value of the SYSTEM RESOURCES.

MEAN VALUE -- Average value per program occurrence within the reported SYSTEM RESOURCES.

ACCUM VALUE -- Total value for all program occurrences for the SYSTEM RESOURCES (depending upon the line) within the reported time interval.
This ratio (expressed as a percentage) is the accumulated value for this program against the accumulated value for all selected programs active within the reported time interval. This ratio highlights the programs that are consuming the largest amount of system resources.

<table>
<thead>
<tr>
<th>ID</th>
<th>RELEASE</th>
<th>DATE</th>
<th>TIME</th>
<th>CA-IDMS</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn</td>
<td>/TASK</td>
<td>RELEASE</td>
<td>DATE</td>
<td>TIME</td>
<td>CA-IDMS</td>
</tr>
<tr>
<td>yy</td>
<td>/TASK</td>
<td>Rnn.</td>
<td>SYSTEM SUMMARY REPORT</td>
<td>mm/dd</td>
<td></td>
</tr>
</tbody>
</table>

Program System Summary Report:

### Summary for all tasks

| Total Run Units | 1919 |
| Total Abends   | 3    |
| Terminal Reads | 62   |
| Terminal Writes | 63    |
| Terminal Errors | 0     |
| Storage Acquired | 1022 |
| Storage Allocated | 266816 |
| Scratch - Gets | 71   |
| Scratch - Puts | 79   |
| Scratch - Deletes | 79 |
| Queue - Gets | 2148 |
| Queue - Puts | 476  |
| Queue - Deletes | 4 |

About CA IDMS Task Analyzer CA ADS Reports

**Contents**

- Three Reports One Set of CA IDMS Log or SMF File Statistics (see page 766)
- Hierarchical Nature of Reports (see page 766)
- Overview of CA ADS Reports (see page 766)
The CA IDMS Task Analyzer CA ADS Reports are available for reporting on dialogs. They use information from the CA IDMS Log (or, optionally under z/OS, from the SMF file) to offer detailed and summarized statistical reports that show how efficiently CA ADS dialogs are using system resources.

The CA ADS Reports are organized by the CA ADS dialogs that make up the tasks. Information about the CA ADS dialog is grouped by selected time intervals. Identifying information includes the dialog’s name, version, type, and origin of execution. First, each dialog’s totals are reported, then they are reflected as a ratio of the system totals (that is, the percentage of all system resources used during the time interval by dialogs that were selected by CA IDMS Task Analyzer).

You will get this report by specifying REPORT = ADSO.

Three Reports One Set of CA IDMS Log or SMF File Statistics

Physically, there are three CA ADS Reports to choose from. It is important, however, to understand that each report is produced from the same statistics taken from the CA IDMS Log (or, optionally under z/OS, from the SMF file). Statistics are presented in different formats and at various levels of summarization. The CA ADS Report is available at the detail, summary, and system summary level.

Hierarchical Nature of Reports

Program reports are produced for three hierarchical levels: if you ask for the lowest level report (LEVL = DET), you will also receive the higher-level reports. This would include the CA ADS Summary Report, which summarizes the data of the CA ADS Details Reports (LEVL = SUM), and also the CA ADS System Summary Report (LEVL = SYS).

Overview of CA ADS Reports

- **CA ADS Details Report** -- shows the information for CA ADS dialogs that make up each task reported in time sequence, within the selected time interval. You may specify which dialog or class of dialogs CA IDMS Task Analyzer is to report on. You also control the time period and duration of the time interval.

- **CA ADS Summary Report** -- records the sum for all tasks that use the CA ADS dialog within the time interval you select. In addition, if multiple versions of a reported task exist, a summary of all tasks by version is presented.

- **CA ADS System Summary** -- presents a sum of all CA ADS Summaries within the time interval you selected.

CA IDMS Task Analyzer CA ADS Details Report

**Contents**

- Report Fields (see page 767)

The CA IDMS Task Analyzer CA ADS Details Report is produced for CA ADS dialogs. The CA ADS Details Report presents a detailed view of the activity of each task that is an CA ADS dialog. The CA ADS dialogs are identified by CA ADS dialog name and version, and are reported in time sequence.
You will get the CA ADS Details Report only if you specify LEVL = DET. In addition, with this specification, you will receive the CA ADS Summary Report and the CA ADS Summary Report.

Report Fields

Here is a description of the various fields that make up the CA IDMS Task Analyzer CA ADS Details Report. A sample report is shown in CA ADS Details Report screen.

**REPORT TITLE** -- The title line of this report varies depending on the RUNAME specified and on the dialog name and version number you select on the NAME parameter statement.

**REQUESTED TIME INTERVAL** -- This line lists the start and stop date/time of the time interval you specified on the PROCESS statement. The data displayed in this line depends on what you select using the START and STOP parameters.

**ACTUAL INTERVAL** -- This line lists the first start date and time and the last start date and time on the CA IDMS Log or SMF File reported on. The data displayed in this line depends upon the actual time range of task activity within the CA IDMS Log or SMF File.

**VARIABLE COLUMNS** -- These two columns vary in content. Any combination of OPER-ID, TERM-ID, or TASK CODE can appear in the first two columns. (RUNAME types that do not appear in the REPORT TITLE will appear in these two columns.)

**TASK INFORMATION** -- Identifying information on the reported task.

- **NUMBER** -- Number of the task within the date and time interval selected.
- **VER** -- Version of the task. Multiple versions of a task are reported separately.
- **OR** -- Origin of the task. The operating system or environment where execution of the task originated.
  - D: CA IDMS/DC
  - C: CICS
  - V: z/VM
  - B: batch
- **START DATE-TIME** -- The start date and time of the task being reported.
- **C C**--Condition code for CA IDMS. If the task abends while running under CA IDMS, an "X" appears on the report under "C C". If the task does not abend while running under CA IDMS, the "C C" column is blank.

**DIALOG INFORMATION**

- **NAME** -- The name of each dialog contained within the reported task.
- **VER** -- The version of the dialog.
INTER DIALOG -- This is a summary of all commands issued by the CA ADS dialog.

- DSP -- The number of display and display continue commands issued by the dialog.
- INV -- The number of invoke commands issued by the dialog.
- LNK -- The number of link to dialog and link to program commands issued by the dialog.
- RET -- The number of return and return continue commands issued by the dialog.
- TRN -- The number of transfer commands issued by the dialog.
- LEV -- The number of leave CA ADS and leave applications commands issued by the dialog.

PRCS -- Premap and response processes.

- PRE MAP -- The number of premap processes.
- RSP -- The number of response processes.

DETAIL -- These statistics show the amount of processing performed using the Pageable Maps feature.

- PUT NEW -- The number of writes to the Detail scratch area.
- PUT CUR -- The number of rewrites to the Detail scratch area.
- GET -- The number of reads from the Detail scratch area.

LINK -- Link levels.

- HGH -- The highest level in the associated CA ADS transaction at which the reported task was executed.
- LOW -- The lowest level in the associated CA ADS transaction at which the reported task was executed.

RBB -- Record buffer block.

- MAX -- The maximum size (in bytes) of record buffer blocks allocated.
- MIN -- The minimum, or least, size (in bytes) of record buffer blocks allocated.

CV NUMBER -- The number of the CV that the statistics on this report apply to.

PLAN ID -- The statistics plan ID that the statistics on this report apply to.
CA IDMS Task Analyzer CA ADS Summary Report

Contents

- Report Fields (see page 769)

The CA IDMS Task Analyzer CA ADS Summary Report is produced for CA ADS dialogs. The CA ADS Summary Report summarizes activity within the specified time period for all DC run-units (tasks) that are also CA ADS dialogs. The summaries are presented for all dialogs by version numbers.

You will get the CA ADS Summary Report if you specify LEVL = SUM (or if you specify LEVL = DET).

When you look at the CA ADS Summary Report, focus on the % OF SYSTEM OCCURRENCES in the last column, as shown in CA ADS Summary Report screen. These statistics reveal trends on the use of your CA IDMS/DC environment. HIGH VALUE, LOW VALUE, MEAN VALUE, and ACCUMULATED VALUE are reported for each statistical category.

Report Fields

Here is a description of the various fields that make up the CA IDMS Task Analyzer CA ADS Summary Report (see the CA ADS Summary Report screen).

REPOR TITLE -- The title line of this report varies depending on what you selected on the RUNAME and NAME parameter statements.
REQUESTED TIME INTERVAL -- This line lists the start and stop date/time of the time interval you specified on the PROCESS statement. The data displayed in this line depends on what you select using the START and STOP parameters.

ACTUAL INTERVAL -- This line lists the first start date and time and the last start date and time on the CA IDMS Log or SMF File reported on. The data displayed in this line depends upon the actual time range of task activity within the CA IDMS Log or SMF File.

SUMMARY FOR VERSION -- The version number of the dialog. Summaries of multiple versions of a dialog are reported separately.

TOTAL RUN UNITS -- The total number of DC run-units (tasks) terminated within the reported time interval.

TOTAL ABENDS -- The total number of abends that occurred as a result of processing the reported tasks within the date and time interval selected.

CA ADS Dialog Statistics -- The CA ADS dialog statistics are reported (taken from the CA ADS Details Report).

- DISPLAY -- The number of display and display continue commands issued by the task.
- INVOKE -- The number of invoke commands issued by the task.
- LINK -- The number of link to dialog and link to program commands issued by the task.
- RETURN -- The number of return and return continue commands issued by the task.
- TRANSFER -- The number of transfer commands issued by the task.
- LEAVE -- The number of leave CA ADS and leave applications commands issued by the task.
- PROCESS - PREMAP -- The number of premap processes.
- PROCESS - RESPONSE -- The number of response processes.
- DETAIL - PUT NEW -- The number of writes occurring to the Detail scratch area.
- DETAIL - PUT CUR -- The number of records rewritten to the Detail scratch area.
- DETAIL - GET -- The number of details retrieved from the Detail scratch area.
- LINK LEVL - MAX -- The highest level in the associated CA ADS transaction at which the reported task was executed.
- LINK LEVL - MIN -- The lowest level in the associated CA ADS transaction at which the reported task was executed.
- RBB STORAGE - MAX -- The maximum size (in bytes) of record buffer blocks buffers allocated.
- RBB STORAGE - MIN -- The minimum, or least, size (in bytes) of record buffer blocks allocated.
**HIGH VALUE** -- The highest value for each of the CA ADS Dialog Statistics for the reported task (depending upon the line).

**TASK CODE** -- The task code of the task with the highest value of the CA ADS Dialog Statistics (depending upon the line).

**TASK NUMBER** -- Task number; this is the version number of the task with the highest value of the CA ADS Dialog Statistics (depending upon the line).

**LOW VALUE** -- The lowest value for each of the CA ADS Dialog Statistics for the reported task (depending upon the line).

**TASK CODE** -- The task code of the task with the lowest value of the CA ADS Dialog Statistics (depending upon the line).

**TASK NUMBER** -- Task number; this is the version number of the task with the lowest value of the CA ADS Dialog Statistics (depending upon the line).

**MEAN VALUE** -- Average value per DC run-unit occurrence within the reported CA ADS Dialog Statistics (depending upon the line).

**ACCUM VALUE** -- Total value for all DC run-unit occurrences for the CA ADS Dialog Statistics (depending upon the line) within the reported time interval.

**% OF SYSTEM OCCURRENCES** -- This ratio (expressed as a percentage) is the accumulated value for this DC run-unit (CA ADS dialog) against the accumulated value for all selected DC run-units active within the reported time interval. This ratio highlights the DC run-units that are consuming the largest amount of system resources.

<table>
<thead>
<tr>
<th>ID/TASK ANALYZER</th>
<th>RELEASE DATE</th>
<th>TIME</th>
<th>CA-IDMS PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rnn.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/yy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hh:mm:ss</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SUMMARY FOR ALL TASKS**

<table>
<thead>
<tr>
<th>HIGH TASK</th>
<th>TASK VALUE</th>
<th>LOW TASK</th>
<th>TASK VALUE</th>
<th>MEAN VALUE</th>
<th>ACCUM VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADSA</td>
<td>1506</td>
<td>ADSA</td>
<td>1505</td>
<td>1.33</td>
<td>4</td>
</tr>
</tbody>
</table>

**SUMMARY FOR TASK CODE ADSA ORIGIN IDMS VERSION 1**

<table>
<thead>
<tr>
<th>TOTAL RUN UNITS</th>
<th>TOTAL ABENDS</th>
<th>DISPLAY</th>
<th>INVOKE</th>
<th>LINK</th>
<th>RETURN</th>
<th>TRANSFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0.24</td>
<td>0.81</td>
<td>0.00</td>
<td>0.00</td>
<td>7.09</td>
</tr>
</tbody>
</table>

**SUMMARY FOR TASK CODE ADSA ORIGIN IDMS VERSION 1**

<table>
<thead>
<tr>
<th>ORIGIN IDMS</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADSA</td>
<td>1</td>
</tr>
</tbody>
</table>
CA IDMS Task Analyzer CA ADS System Summary Report

Contents

- Report Fields (see page 772)

The CA IDMS Task Analyzer CA ADS System Summary Report presents a sum-total of all CA ADS Summaries within the time interval you select. All statistical categories are reported within run-unit type: CA IDMS/DC, CICS, and z/VM; or ALL.

You will get this report by specifying LEVL = SYS. (This report will also be created if you specify LEVL = DET or LEVL = SUM.)

When you review this report, focus on % of SYSTEM OCCURRENCES in the last column, as shown in ADS System Summary Report. These statistics reflect trends on the use of your CA IDMS/DC environment. HIGH VALUE, LOW VALUE, MEAN VALUE, and ACCUMULATED VALUE are reported for each statistical category.

Report Fields

Here is a description of the various fields that make up the CA ADS System Summary Report.

REPORT TITLE -- The title line of this report varies depending on what you selected on the RUNAME and NAME parameter statements.

REQUESTED TIME INTERVAL -- This line lists the start and stop date/time of the time interval you specified on the PROCESS statement. The data displayed in this line depends on what you select using the START and STOP parameters.

ACTUAL INTERVAL -- This line lists the first start date and time and the last start date and time on the CA IDMS Log or SMF File reported on. The data displayed in this line depends upon the actual time range of task activity within the CA IDMS Log or SMF File.

SUMMARY FOR ALL REQUESTED ENTITIES -- This line indicates that a summarization of statistics follows.
TOTAL RUN UNITS -- The total number of run-units performed by programs that make up the reported task (within the time interval selected).

TOTAL ABENDS -- The total number of abends that occurred as a result of processing the programs that make up the reported task (within the date and time interval selected).

SYSTEM RESOURCES: The CA IDMS/DC statistics are reported. For a detailed explanation of the CA IDMS/DC statistics, see the CA IDMS Task Analyzer CA ADS Details Report.

- DISPLAY
- INVOKE
- LINK
- RETURN
- TRANSFER
- LEAVE
- PROCESS - PREMAP
- PROCESS - RESPONSE
- DETAIL - PUT NEW
- DETAIL - PUT CUR
- DETAIL - GET
- LINK LEVEL - MAX
- LINK LEVEL - MIN
- RBB STORAGE - MAX
- RBB STORAG - MIN

HIGH VALUE -- The highest value for each of the SYSTEM RESOURCES (depending upon the line) for the reported task.

TASK CODE -- The ID of the task with the highest value of the SYSTEM RESOURCES (depending upon the line).

TASK NUM -- Task number; this is the number of the task with the highest value of the SYSTEM RESOURCES.

LOW VALUE -- The lowest value for each of the SYSTEM RESOURCES for the reported task.

TASK CODE -- The ID of the task with the lowest value of the SYSTEM RESOURCES (depending upon the line).
**TASK NUM** -- Task number; this is the number of the task with the highest value of the SYSTEM RESOURCES.

**MEAN VALUE** -- Average value per program occurrence within the reported SYSTEM RESOURCES.

**ACCUM VALUE** -- Total value for all program occurrences for the SYSTEM RESOURCES (depending upon the line) within the reported time interval.

**% OF SYSTEM OCCURRENCES** -- This ratio (expressed as a percentage) is the accumulated value for this program against the accumulated value for all selected programs active within the reported time interval. This ratio highlights the programs that are consuming the largest amount of system resources.

<table>
<thead>
<tr>
<th>ID/TASK ANALYZER</th>
<th>RELEASE</th>
<th>DATE</th>
<th>TIME</th>
<th>CA IDMS SUMMARY REPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rnn.</td>
<td></td>
<td></td>
<td></td>
<td>mm/dd</td>
</tr>
<tr>
<td>/yy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hh:mm:ss</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ADSO SUMMARY**

REQUESTED: mm/dd/yy hh:mm - mm/dd/yy hh:mm

ACTUAL: mm/dd/yy hh:mm - mm/dd/yy hh:mm

**SYSTEM**

<table>
<thead>
<tr>
<th>HIGH TASK VALUE</th>
<th>TASK CODE</th>
<th>LOW TASK VALUE</th>
<th>TASK CODE</th>
<th>MEAN TASK VALUE</th>
<th>ACCUM TASK VALUE</th>
<th>% OF OCCURRENCE</th>
</tr>
</thead>
</table>

**SUMMARY FOR REQUESTED ENTITIES**

TOTAL RUN UNITS 24
TOTAL ABENDS 00
DISPLAY 81
INVOKE 00
LINK 00
RETURN 00
TRANSFER 69
LEAVE 62
PROCESS - PREMAP 23
PROCESS - RESPONSE 00
DETAIL - PUT NEW 00
DETAIL - PUT CUR 00
DETAIL - GET 00
LINK LEVEL - MAX 68
LINK LEVEL - MIN 74
RBB STORAGE - MAX 13976
RBB STORAGE - MIN 13976

CA ADS System Summary Report:
About CA IDMS Task Analyzer Abend Report

Contents
- Tying Abend Activity to an ID and a Time (see page 775)
- One Report One Set of CA IDMS Log Statistics (see page 775)
- Overview of Abend Reports (see page 775)

The CA IDMS Task Analyzer Abend Report uses information from the CA IDMS Log (or, optionally under z/OS, the SMF file) to produce detailed report statistics. The Abend Report reports on those tasks that have abended while running under CA IDMS, whether or not the tasks have issued database calls. (If a task abends while running under CA IDMS, an "X" appears under "C C" on the Billing Details, Program Details, CA ADS Details, or Program Loads report.) You will get the Abend Report by specifying REPORT = ABND on the parameter statement.

Tying Abend Activity to an ID and a Time

Depending on the parameters you choose, abend activity can be tied to a specific operator, terminal, task code, or group. For CA IDMS/DC tasks, the data reported under the headings OPER-ID, TERM-ID, or TASK CODE comes from the CA IDMS Log (or, optionally under z/OS, the SMF file). For CICS tasks, this information is taken from the External Request Element (ERE) Extension; the information is available only if GSISVCX was installed.

The CA IDMS Task Analyzer Abend Report presents this information within the framework of the time interval you select. Task, time, and error message information is shown.

One Report One Set of CA IDMS Log Statistics

Physically, there is one Abend Report. The report is produced from the statistics that are found on the CA IDMS Log (or, optionally under z/OS, the SMF file). Statistics in the Abend Report are presented at the detail level.

Overview of Abend Reports

Abend Report presents detailed information for each task that abends while running under CA IDMS, reported in termination time sequence. Depending on the parameter combination selected, this report allows you to identify task activity by terminal ID, operator ID, task code, or group name.

CA IDMS Task Analyzer Abend Report

Contents
- Report Fields (see page 776)

The CA IDMS Task Analyzer Abend Report presents a detailed view of the activity of each task that abended while running under CA IDMS, reported in time sequence and based on the parameters selected. Depending on the parameter combination you select, this report allows you to identify by terminal ID, operator ID, task code, or program name tasks that abended while running under CA IDMS.

You will get the Abend Report only if you specify LEVL = DET.
Report Fields

Here is a description of the various fields that make up the CA IDMS Task Analyzer Abend Details Report. The Abend Report screen shows one possible type of Abend Report CA IDMS Task Analyzer will generate.

REPORT TITLE -- The title line of this report varies depending on what you selected on the RUNAME and NAME parameter statements.

REQUESTED TIME INTERVAL -- This line lists the start and stop date/time of the time interval you specified on the PROCESS statement. The data displayed in this line depends on what you select using the START and STOP parameters.

ACTUAL INTERVAL -- This line lists the first start date and time and the last start date and time on the CA IDMS Log or SMF File reported on. The data displayed in this line depends upon the actual time range of task activity within the CA IDMS Log or SMF File.

VARIABLE COLUMNS -- These two columns vary in content. Any combination of OPER-ID, TERM-ID, or TASK CODE can appear in the first two columns. (RUNAME types that do not appear in the REPORT TITLE will appear in these two columns.)

TASK INFORMATION

- NUMBER -- Number of the task within the date and time interval selected.
- VER -- Version of the task.
- TYPE -- Type of task performed, indicating the language of the program the task invokes.
  - ASSEM: Assembler
  - COBOL: COBOL
  - ADSO: CA ADS
  - PL1: PL/1
  - FTRAN: FORTRAN
  - ORG -- Origin of the task. The operating system or environment where execution of the task originated.
  - IDMS: CA IDMS/DC
  - CICS: CICS
  - z/VM: z/VM
  - BTCH: batch
- START DATE-TIME -- The start date and time of the task being reported.

MESSAGE INFORMATION
CA IDMS - 19.0

**ABEND CODE** -- The return code received from CA IDMS.

**MESSAGE NUMBER** -- The CA IDMS message number.

**SEVR CODE** -- The severity code of the CA IDMS message, ranging from 0-3.

**ABEND SUMMARY** -- The total number of abends that occurred during the execution of the reported task.

**CV NUMBER** -- The number of the CV that the statistics on this report apply to.

**PLAN ID** -- The statistics plan ID that the statistics on this report apply to.

---

CV NUMBER: 19

PLAN ID: PLAN0001

DETAILS FOR TASK CODE ALL TASKS

REQUESTED: mm/dd/yy hh

ACTUAL: mm/dd/yy hh

---

**TASK**

<table>
<thead>
<tr>
<th>ID</th>
<th>CODE</th>
<th>TERM</th>
<th>NUMBER</th>
<th>VER</th>
<th>TYPE</th>
<th>ORG</th>
<th>START</th>
<th>DATE-</th>
<th>SEVR</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLM</td>
<td>VTAMLTD2</td>
<td>D0002</td>
<td>027</td>
<td>2299</td>
<td>ASSEM</td>
<td>IDMS</td>
<td>04/14</td>
<td>14:05</td>
<td></td>
</tr>
<tr>
<td>ADSO@ML$L</td>
<td>VTAMLTD2</td>
<td>D0002</td>
<td>027</td>
<td>2616</td>
<td>ADSO</td>
<td>IDMS</td>
<td>04/14</td>
<td>14:32</td>
<td></td>
</tr>
<tr>
<td>ADSO@MS$</td>
<td>VTAMLTD6</td>
<td>D0002</td>
<td>027</td>
<td>3156</td>
<td>ADSO</td>
<td>IDMS</td>
<td>04/14</td>
<td>15:40</td>
<td></td>
</tr>
</tbody>
</table>

***** 3 ABENDS FOR TASK CODE ALL TASKS

Abend Report:

About CA IDMS Task Analyzer Program Loads Report

**Contents**

- Tying Task Activity to an ID and a Time (see page 778)
- One Report One Set of CA IDMS Log Statistics (see page 778)
- Overview of Program Loads Reports (see page 778)

The CA IDMS Task Analyzer Program Loads Report uses information from the CA IDMS Log (or, optionally under z/OS, the SMF file) to produce detailed report statistics. The Program Loads Report reports on the primary program and secondary programs (including tables, maps, and subschemas) that a task loads (calls). You will get the Program Loads Report by specifying REPORT=LOAD on the parameter statement.
Tying Task Activity to an ID and a Time

Depending on the parameters you choose, abend activity can be tied to a specific operator, terminal, task code, or group. The data reported under the headings OPER-ID, TERM-ID, or TASK CODE comes from the CA IDMS Log (or, optionally under z/OS, from the SMF file.)

For CICS, this information is taken from the External Request Element (ERE) Extension as it is built by the CA IDMS SVC exit routine. The CA IDMS Task Analyzer Program Loads Report presents this information within the framework of the time interval you select. Task, time, and error message information is shown.

One Report One Set of CA IDMS Log Statistics

Physically, there is one Program Loads Report. The report is produced from the statistics that are found on the CA IDMS Log or, optionally under z/OS, the SMF file. Statistics in the Program Loads Report are presented at the detail level.

Overview of Program Loads Reports

- **Program Loads Details Report**
  Presents for each specified task the primary program and all secondary programs (including tables, maps, and subschemas), reported in termination time sequence. Depending on the parameter combination selected, this report allows you to identify task activity by terminal ID, operator ID, task code, group name, or for all tasks.

CA IDMS Task Analyzer Program Loads Report

**Contents**

- Report Fields (see page 778)

The CA IDMS Task Analyzer Program Loads Report presents a detailed view of the primary program and all secondary programs (including tables, maps, and subschemas) that a task loads (calls). The tasks are identified by task code, and are reported in time sequence. You will get the Program Loads Report only if you specify LEVL = DET.

**Report Fields**

Here is a description of the various fields that make up the CA IDMS Task Analyzer Program Loads Report. A sample report is shown in Program Loads Report screen.

**REPORT TITLE** -- The title line of this report varies depending on what you selected on the RUNAME and NAME parameter statements.

**REQUESTED TIME INTERVAL** -- This line lists the start and stop date/time of the time interval you specified on the PROCESS statement. The data displayed in this line depends on what you select using the START and STOP parameters.

**ACTUAL INTERVAL** -- This line lists the first start date and time and the last start date and time on the CA IDMS Log or SMF File reported on. The data displayed in this line depends upon the actual time range of task activity within the CA IDMS Log or SMF File.
VARIABLE COLUMNS -- These two columns vary in content. Any combination of OPER-ID, TERM-ID, or TASK CODE can appear in the first two columns. (RUNAME types that do not appear in the REPORT TITLE will appear in these two columns.)

TASK INFORMATION -- Identifying information on the task.

- **NUMBER** -- Number of the task within the date and time interval selected.
- **VER** -- Version of the task. Multiple versions of a task are reported separately.
- **ORG** -- Origin of the task. The operating system or environment where execution of the task originated.
- IDMS: CA IDMS/DC
- CICS: CICS
- z/VM: z/VM
- BTCH: batch
- **START DATE-TIME** -- The start date and time of the task being reported.
- **C C** -- Condition code for CA IDMS. If the task abended while running under CA IDMS, an "X" appears on the report under "C C". If the task did not abend while running under CA IDMS, the "C C" column is blank.

PRIMARY PROGRAM -- Information about the program or dialog initiating the transaction.

- **NAME** -- The name of the primary program.
- **VER** -- The version of the primary program. Multiple versions are reported separately.
- **TYPE** -- Type of task performed, indicating the language of the program the task invokes.
- ASSEM: Assembler
- COBOL: COBOL
- ADSO: CA ADS
- FTRAN: Fortran
- PL1: PL/1

SECONDARY PROGRAM -- Information about secondary programs (including tables, maps, and subschemas) loaded or linked by the primary program. These statistics reveal what programs are lower level programs within the reported task, and reveal how frequently the lower level programs are loaded or linked.

- **NAME** -- The name of the secondary program loaded or linked by the primary program for the reported task.
- **VER** -- The version of the secondary program. Multiple versions are listed separately.
- **TYPE** -- The type of the secondary program loaded or linked. Multiple types of a secondary program are listed separately.
- **ASSEM**: Assembler
- **COBOL**: COBOL
- **ADSO**: CA ADS
- **PL1**: PL/1
- **SUBS**: Subschema
- **MAP**: Map
- **FTRAN**: Fortran
- **TABLE**: Table
- **COUNT** -- The number of times the secondary program is loaded or linked by the primary program during each execution of the reported task.

**CV NUMBER** -- The number of the CV that the statistics on this report apply to.

**PLAN ID** -- The statistics plan ID that the statistics on this report apply to.

<table>
<thead>
<tr>
<th>CV NUMBER: 19</th>
<th>PLAN ID: PLAN0001</th>
<th>DETAILS FOR TASK CODE</th>
<th>ADS</th>
</tr>
</thead>
<tbody>
<tr>
<td>mm/dd/yyyy</td>
<td>hh:mm</td>
<td>mm/dd/yyyy</td>
<td>hh:mm</td>
</tr>
</tbody>
</table>

**REQUESTED**:

**ACTUAL**:

<table>
<thead>
<tr>
<th>OPER</th>
<th>TERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>-----</td>
<td>------</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PRIMARY PROGRAM</th>
<th>----------SECONDARY PROGRAM----------</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ID</td>
</tr>
<tr>
<td>VTAMLT04</td>
<td>162</td>
</tr>
<tr>
<td>36</td>
<td></td>
</tr>
<tr>
<td>CDO2NWKS</td>
<td>1</td>
</tr>
<tr>
<td>RHDCRUAL</td>
<td>0</td>
</tr>
<tr>
<td>ADSOMENU</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SECONDARY PROGRAM</th>
<th>----------SECONDARY PROGRAM----------</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ID</td>
</tr>
<tr>
<td>VTAMLT04</td>
<td>163</td>
</tr>
<tr>
<td>37</td>
<td></td>
</tr>
<tr>
<td>CDO2NWKS</td>
<td>1</td>
</tr>
<tr>
<td>RHDCRUAL</td>
<td>0</td>
</tr>
<tr>
<td>ADSOMENU</td>
<td>1</td>
</tr>
</tbody>
</table>
About CA IDMS Task Analyzer Integrated Index Reports

Contents

- Tying Task Activity to an ID and a Time (see page 781)
- Three Reports One Set of CA IDMS Log or SMF File Statistics (see page 782)
- Hierarchical Nature of Reports (see page 782)
- Overview of Integrated Index Reports (see page 782)

The CA IDMS Task Analyzer Integrated Index Reports use information from the CA IDMS Log (or, optionally under z/OS, the SMF file) to produce both detailed and summarized report statistics on how programs affect your integrated indexing structure. The Integrated Index Reports differ from the other CA IDMS Task Analyzer reports in that the Integrated Index Reports list database activity. You will get this report by specifying REPORT = INDEX on the parameter statement.

Tying Task Activity to an ID and a Time

Depending on the parameters you choose, task activity can be tied to a specific user, terminal, task code, or group. For CA IDMS/DC tasks, the data reported under the headings OPER-ID, TERM-ID, or TASK CODE comes from the CA IDMS Log (or, optionally under z/OS, the SMF file).
For CICS, this information is taken from the External Request Element (ERE) Extension as it is built by the CA version of the CA IDMS SVC exit routine.

The CA IDMS Task Analyzer Integrated Index Reports present this information within the framework of the time interval you select. The effects of task processing upon your current integrated indexing structure are displayed on the report.

Three Reports One Set of CA IDMS Log or SMF File Statistics

Physically, there are three Integrated Index Reports to choose from. However, it is important to understand that each report is produced from the statistics that are found on the CA IDMS Log (or, optionally under z/OS, the SMF file). Statistics in the reports are presented in various formats and at one level of summarization. The Integrated Index Report is available at the detail, summary, and system summary levels.

Hierarchical Nature of Reports

Integrated Index Reports are produced on a hierarchical level: if you ask for the lowest level report (LEVL = DET), you will also receive the higher-level report, the Integrated Index Summary Report, which summarizes the data of the Integrated Index Details Report (LEVL = SUM), and also the Integrated Index System Summary Report.

Overview of Integrated Index Reports

- **Integrated Index Details Report** -- presents detailed information for each task, reported in termination time sequence. Depending on the parameter combination selected, this report allows you to identify task activity by operator ID, terminal ID, task code, group ID, or for all tasks.

- **Integrated Index Summary** -- records the sum of all tasks invoked by an operator ID, terminal ID, task code ID, or group ID within the time interval you selected.

- **Integrated Index System Summary** -- presents a sum of all Integrated Index Summaries within the time interval you selected.

CA IDMS Task Analyzer Integrated Index Details Report

**Contents**

- Report Fields (see page 783)

The CA IDMS Task Analyzer Integrated Index Details Report presents a detailed view of the activity of each task activity reported in time sequence, based on the parameters selected. Depending on the parameter combination you select, this report allows you to identify task activity by terminal ID, operator ID, task code, group ID, or for all tasks.

You will get the Integrated Index Details Report only if you specify LEVL = DET. In addition, with this specification, you will receive the Integrated Index Summary Report and the Integrated Index System Summary Report.
Report Fields

Here is a description of the various fields that make up the CA IDMS Task Analyzer Integrated Index Details Report. Integrated Index Details Report shows one possible type of Integrated Index Details Reports CA IDMS Task Analyzer will generate.

REPORT TITLE -- The title line of this report varies depending on what you select on the RUNAME and NAME parameter statements.

REQUESTED TIME INTERVAL -- This line lists the start and stop date/time of the time interval you specified on the PROCESS statement. The data displayed in this line depends on what you select using the START and STOP parameters.

ACTUAL INTERVAL -- This line lists the first start date and time and the last start date and time on the CA IDMS Log or SMF File reported on. The data displayed in this line depends upon the actual time range of task activity within the CA IDMS Log or SMF File.

VARIABLE COLUMNS -- These two columns vary in content. Any combination of OPER-ID, TERM-ID, or TASK CODE can appear in the first two columns. (RUNAME types that do not appear in the REPORT TITLE will appear in these two columns.)

TASK INFORMATION

- **NUMBER** -- Number of the task within the date and time interval selected.
- **VER** -- Version of the task. Multiple versions of a task are reported separately.
- **TY** -- Type of task performed, indicating the language of the program the task invokes.
  - A: Assembler
  - C: COBOL
  - N: CA ADS
  - P: PL/1
  - F: Fortran
  - OR -- Origin of the task. The operating system or environment where execution of the task originated.
  - D: CA IDMS/DC
  - C: CICS
  - V: z/VM
  - B: batch
- **START DATE-TIME** -- The start date and time of the task being reported.
- **C C** -- Condition code for CA IDMS. If the task abends while running under CA IDMS, an "X" appears on the report under "C C". If the task does not abend while running under CA IDMS, the "C C" column is blank.

**PROGRAM INFORMATION**

- **NAME** -- The name of the program invoked by the reported task.
- **VER** -- Version of the program. Multiple versions of a program are reported separately.
- **TY** -- Type of task performed, indicating the language of the program invoked by the task. **A**: Assembler **C**: COBOL **N**: CA ADS **P**: PL/1 **F**: Fortran

**SR8 INDEX INFORMATION**

- **SPLIT** -- The number of SR8 record splits that have occurred as a result of processing the reported task.
- **SPAWN** -- The number of SR8 record spawns that have occurred as a result of processing the reported task.
- **STORED** -- The number of SR8 records stored as a result of processing the reported task.
- **ERASED** -- The number of SR8 records erased as a result of processing the reported task.

**SR7 INDEX INFORMATION**

- **STORED** -- The number of SR7 records stored as a result of processing the reported task.
- **ERASED** -- The number of SR7 records stored as a result of processing the reported task.

**B-TREE INFORMATION**

- **SEARCH** -- The number of searches into the integrated index required to locate the reported task.
- **LEVEL** -- The number of levels required to complete the search.

**ORPHANS ADOPTED** -- The number of integrated index orphans adopted as a result of processing the reported task.

*Programs loaded and executed by the previous task.

**CV NUMBER** -- The number of the CV that the statistics on this report apply to.

**PLAN ID** -- The statistics plan ID that the statistics on this report apply to.
| TASK | TERM | TASK PROGRAM- | ID | NUMBER | VER | T | O | START DATE- | TIME | C | NAME | VER | T | SPLIT | SPAWN | STORED | ERASED | STORED | ERASED | SEARCH | LEVEL | ADO |
|------|------|---------------|----|--------|-----|---|---|-------------|------|---|------|-----|---|-------|-------|--------|--------|--------|--------|-------|     |     |
| ASFFDEFD | VTAMLTL01 | 45 | 1 | N | D | 04/17 12:29:55 | RHDCRUAL | 0 | 0 | 0 | 0 | 390 781 | IDMSRSUB | 1 | S | 0 | 0 | 0 | | |
| ASFFDEFD | VTAMLTL01 | 46 | 1 | N | D | 04/17 12:30:11 | RHDCRUAL | 0 | 0 | 0 | 0 | 147 298 | IDMSRSUB | 1 | S | 0 | 0 | 0 | | |
| ASFRDEFD | VTAMLTL01 | 47 | 1 | N | D | 04/17 12:30:26 | RHDCRUAL | 0 | 0 | 0 | 0 | 141 354 | IDBCAT | 1 | A | 0 | 0 | 0 | | |
| ASFASELD | VTAMLTL01 | 49 | 1 | N | D | 04/17 12:34:03 | RHDCRUAL | 0 | 0 | 0 | 0 | 472 1219 | IDMSRSUB | 1 | S | 0 | 0 | 0 | | |
| ASFDSELD | VTAMLTL01 | 51 | 1 | N | D | 04/17 12:34:23 | RHDCRUAL | 0 | 0 | 0 | 0 | 414 852 | RD000231 | 1 | N | 0 | 0 | 0 | | |
| RD000231 | VTAMLTL01 | 52 | 1 | N | D | 04/17 12:37:38 | RHDCRUAL | 0 | 0 | 0 | 0 | 17 36 | IDMSRRTM | 1 | A | 0 | 0 | 0 | | |
| RD000231 | VTAMLTL01 | 53 | 1 | N | D | 04/17 12:37:49 | RHDCRUAL | 0 | 0 | 0 | 0 | 2 2 | RD000231 | 1 | N | 0 | 0 | 0 | | |
| RD000231 | VTAMLTL01 | 54 | 1 | N | D | 04/17 12:37:53 | RHDCRUAL | 0 | 0 | 0 | 0 | 2 2 | RD000231 | 1 | N | 0 | 0 | 0 | | |
| RD000231 | VTAMLTL01 | 55 | 1 | N | D | 04/17 12:38:00 | RHDCRUAL | 0 | 0 | 0 | 0 | 2 2 | RD000231 | 1 | N | 0 | 0 | 0 | | |
| ASFASELD | VTAMLTL01 | 58 | 1 | N | D | 04/17 12:38:27 | RHDCRUAL | 0 | 0 | 0 | 0 | 2 2 | IDMSRSUB | 1 | S | 0 | 0 | 0 | | |
| ASFRDEFD | VTAMLTL01 | 59 | 1 | N | D | 04/17 12:39:04 | RHDCRUAL | 0 | 0 | 0 | 0 | 279 582 | IDBCAT | 1 | A | 0 | 0 | 0 | | |
| ASFFDEFD | VTAMLTL01 | 60 | 1 | N | D | 04/17 12:39:26 | RHDCRUAL | 0 | 0 | 0 | 0 | 631 1261 | IDMSRSUB | 1 | S | 0 | 0 | 0 | | |
Integrated Index Details Report:

Integrated Index Details Report:

Integrated Index Details Report:

Integrated Index Details Report:

Integrated Index Details Report:

Integrated Index Details Report:

Integrated Index Details Report:

Integrated Index Details Report:

CA IDMS Task Analyzer Integrated Index Summary Report

Contents

- Report Fields (see page 786)

The CA IDMS Task Analyzer Integrated Index Summary Report summarizes all tasks executed for a terminal ID, operator ID, task code, or group within the time interval you select. It is a summary of information from the CA IDMS Task Analyzer Integrated Index Details Report. You will get this report if you specify LEVL = SUM (or if you specify LEVL = DET).

When you look at the Integrated Index Summary Report, focus on the % OF SYSTEM OCCURRENCES in the last column, as shown in Index Summary Report screen. HIGH VALUE, LOW VALUE, MEAN VALUE, and ACCUMULATED VALUE are reported for each statistical category.

Report Fields

Here is an explanation of the fields that make up the CA IDMS Task Analyzer Integrated Index Summary Report.

REPORT TITLE -- The title line of this report varies depending on what you selected on the RUNAME and NAME parameter statements.

REQUESTED TIME INTERVAL -- This line lists the start and stop date/time of the time interval you specified on the PROCESS statement. The data displayed in this line depends on what you select using the START and STOP parameters.

ACTUAL INTERVAL -- This line lists the first start date and time and the last start date and time on the CA IDMS Log or SMF File reported on. The data displayed in this line depends upon the actual time range of task activity within the CA IDMS Log or SMF File.

SUMMARY FOR TASK CODE -- Task code identification information; this includes task name, origin of execution, and version. Multiple versions of a task are reported separately; tasks with multiple origins of execution are also reported separately.

TOTAL RUN UNITS -- The total number of run-units performed by programs that make up the reported task (within the time interval selected).

TOTAL ABENDS -- The total number of abends that occurred as a result of processing the programs that make up the reported task (within the date and time interval selected).
SYSTEM RESOURCES -- The CA IDMS/DC statistics are reported. For a detailed explanation of the integrated index statistics, see the CA IDMS Task Analyzer Integrated Index Details Report.

- SR8 SPLITS
- SR8 SPAWNS
- SR8 STORED
- SR8 ERASED
- SR7 STORED
- SR7 ERASED
- B-TREE SEARCHES
- B-TREE LEVEL SEARCHES
- ORPHANS ADOPTED

HIGH VALUE -- The highest value for each of the INDEX INFORMATION lines (depending upon the line) for the reported task.

TASK CODE -- The task code of the task with the highest value of the INDEX INFORMATION lines (depending upon the line).

TASK NUMBER -- Task number; this is the version number of the task with the highest value of the INDEX INFORMATION.

LOW VALUE -- The lowest value for each of the INDEX INFORMATION lines for the reported task.

TASK CODE -- The task code of the task with the lowest value of the INDEX INFORMATION lines (depending upon the line).

TASK NUMBER -- Task number; this is the version number of the task with the lowest value of the INDEX INFORMATION lines.

MEAN VALUE -- Average value per task occurrence within the reported INDEX INFORMATION lines.

ACCUM VALUE -- Total value for all task occurrences for the INDEX INFORMATION lines (depending upon the line) within the reported time interval.

% OF SYSTEM OCCURRENCES -- This ratio (expressed as a percentage) is the accumulated value for this task against the accumulated value for all selected tasks active within the reported time interval. This ratio highlights the tasks that are consuming the largest amount of system resources.
### Summary for Task Code ASFRDEFD Origin IDMS Version 1

<table>
<thead>
<tr>
<th>Category</th>
<th>Value</th>
<th>Code</th>
<th>Number</th>
<th>Value</th>
<th>Code</th>
<th>Number</th>
<th>Value</th>
<th>Occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TOTAL RUN UNITS</strong></td>
<td>6</td>
<td>0.48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TOTAL ABENDS</strong></td>
<td>0</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SR8 SPLITS</td>
<td>4</td>
<td>4.55</td>
<td>47</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>9</td>
<td>4.29</td>
</tr>
<tr>
<td>SR8 SPAWNS</td>
<td>0</td>
<td>0.00</td>
<td>35</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>9</td>
<td>4.29</td>
</tr>
<tr>
<td>SR8 STORED</td>
<td>4</td>
<td>4.29</td>
<td>47</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>9</td>
<td>4.29</td>
</tr>
<tr>
<td>SR8 ERASED</td>
<td>7</td>
<td>80.00</td>
<td>65</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>4</td>
<td>80.00</td>
</tr>
<tr>
<td>SR7 STORED</td>
<td>0</td>
<td>0.00</td>
<td>35</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>SR7 ERASED</td>
<td>0</td>
<td>0.00</td>
<td>35</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>B- TREE SEARCHES</td>
<td>687</td>
<td>12.34</td>
<td>ASFRDEFD</td>
<td>65</td>
<td>15</td>
<td>ASFRDEFD</td>
<td>404</td>
<td>322.83</td>
</tr>
<tr>
<td>B- TREE LEVEL SEARCHES</td>
<td>45</td>
<td>13.94</td>
<td>ASFRDEFD</td>
<td>65</td>
<td>30</td>
<td>ASFRDEFD</td>
<td>404</td>
<td>774.33</td>
</tr>
</tbody>
</table>

### Summary for Task Code ASFSIGND Origin IDMS Version 1

<table>
<thead>
<tr>
<th>Category</th>
<th>Value</th>
<th>Code</th>
<th>Number</th>
<th>Value</th>
<th>Occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL RUN UNITS</td>
<td>3</td>
<td>0.24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL ABENDS</td>
<td>0</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SR8 SPLITS</td>
<td>0</td>
<td>0.00</td>
<td>374</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>SR8 SPAWNS</td>
<td>0</td>
<td>0.00</td>
<td>374</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>SR8 STORED</td>
<td>0</td>
<td>0.00</td>
<td>374</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>SR8 ERASED</td>
<td>0</td>
<td>0.00</td>
<td>374</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>SR7 STORED</td>
<td>0</td>
<td>0.00</td>
<td>374</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>SR7 ERASED</td>
<td>0</td>
<td>0.00</td>
<td>374</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>B- TREE SEARCHES</td>
<td>15</td>
<td>0.29</td>
<td>ASFSIGND</td>
<td>374</td>
<td>15</td>
</tr>
<tr>
<td>B- TREE LEVEL SEARCHES</td>
<td>30</td>
<td>0.27</td>
<td>ASFSIGND</td>
<td>374</td>
<td>30</td>
</tr>
</tbody>
</table>

### Summary for Task Code ASFXDERD Origin IDMS Version 1

<table>
<thead>
<tr>
<th>Category</th>
<th>Value</th>
<th>Code</th>
<th>Number</th>
<th>Value</th>
<th>Occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL RUN UNITS</td>
<td>3</td>
<td>0.24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL ABENDS</td>
<td>0</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SR8 SPLITS</td>
<td>5</td>
<td>4.04</td>
<td>ASFXDERD</td>
<td>36</td>
<td>0</td>
</tr>
</tbody>
</table>
Integrated Index System Summary Report

Contents

- Report Fields (see page 789)

The CA IDMS Task Analyzer Integrated Index System Summary Report presents a sum-total of all
Integrated Index Summaries within the time interval you select. All statistical categories are reported
within run-unit origin: CA IDMS/DC, CICS, and z/VM; or ALL. You will get this report by specifying LEVL = SYS. (This report will also be created if you specify LEVL = DET or LEVL = SUM.)

When you review this report, focus on % of SYSTEM OCCURRENCES in the last column, as shown in
System Summary Report screen. These statistics reflect trends on the use of your CA IDMS/DC
environment. HIGH VALUE, LOW VALUE, MEAN VALUE, and ACCUMULATED VALUE are reported for
each statistical category.

Report Fields

Here is a description of the various fields that make up the System Summary Report.

REPORT TITLE -- The title line of this report varies depending on what you selected on the RUNAME
and NAME parameter statements.

REQUESTED TIME INTERVAL -- This line lists the start and stop date/time of the time interval you
specified on the PROCESS statement. The data displayed in this line depends on what you select using
the START and STOP parameters.

ACTUAL INTERVAL -- This line lists the first start date and time and the last start date and time on the
CA IDMS Log or SMF File reported on. The data displayed in this line depends upon the actual time
range of task activity within the CA IDMS Log or SMF File.

SUMMARY FOR ALL REQUESTED ENTITIES -- This line indicate that a summarization of statistics
follows.

TOTAL RUN UNITS -- The total number of run-units performed by programs that make up the
reported task (within the time interval selected).
TOTAL ABENDS -- The total number of abends that occurred as a result of processing the programs that make up the reported task (within the date and time interval selected).

SYSTEM RESOURCES: The CA IDMS/DC statistics are reported. For a detailed explanation of the CA IDMS/DC statistics, see the CA IDMS Task Analyzer Integrated Index Details Report.

- SR8 SPLITS
- SR8 SPAWNS
- SR8 STORED
- SR8 ERASED
- SR7 STORED
- SR7 ERASED
- B-TREE SEARCHES
- B-TREE LEVEL SEARCHES
- ORPHANS ADOPTED

HIGH VALUE -- The highest value for each of the SYSTEM RESOURCES (depending upon the line) for the reported task.

TASK CODE -- The ID of the task with the highest value of the SYSTEM RESOURCES (depending upon the line).

TASK NUM -- Task number; this is the number of the task with the highest value of the SYSTEM RESOURCES.

LOW VALUE -- The lowest value for each of the SYSTEM RESOURCES for the reported task.

TASK CODE -- The ID of the task with the lowest value of the SYSTEM RESOURCES (depending upon the line).

TASK NUM -- Task number; this is the number of the task with the highest value of the SYSTEM RESOURCES.

MEAN VALUE -- Average value per program occurrence within the reported SYSTEM RESOURCES.

ACCUM VALUE -- Total value for all program occurrences for the SYSTEM RESOURCES (depending upon the line) within the reported time interval.

% OF SYSTEM OCCURRENCES -- This ratio (expressed as a percentage) is the accumulated value for this program against the accumulated value for all selected programs active within the reported time interval. This ratio highlights the programs that are consuming the largest amount of system resources.
Integrated Index System Summary Report:

CA IDMS Task Analyzer Ranking Report

Contents

- Report Fields (see page 792)

The CA IDMS Task Analyzer Ranking Report uses system performance and resource consumption information derived from the original Details Reports. Statistics are ranked under ASCENDING or DESCENDING, depending on the ORDER parameter you select.

Unlike the Details Reports, however, which contain information for all attributes of a task presented in time sequence, the Ranking Report presents the specific task attribute you select, in the sequence you specify. You may also specify whether the ABSOLUTE value of the attribute is to be ranked or whether to rank the task by occurrence EQUAL, LESS THAN, LESS THAN OR EQUAL TO, GREATER THAN, or GREATER THAN OR EQUAL TO the occurrence of the attribute.

You will get this report if you specify REPORT = RANK on the parameter statement.
Report Fields

These fields make up the CA IDMS Task Analyzer Ranking Report. The following report fields are valid for all versions of the Ranking Report:

**REPORT TITLE** -- The title line of this report varies depending on what you select on the HOW and WHAT parameter statements.

**REQUESTED TIME INTERVAL** -- This line lists the start and stop date/time of the time interval you specified on the PROCESS statement. The data displayed in this line depends on what you select using the START and STOP parameters.

**ACTUAL INTERVAL** -- This line lists the first start date and time and the last start date and time on the CA IDMS Log or SMF File reported on. The data displayed in this line depends upon the actual time range of task activity within the CA IDMS Log or SMF File.

**RANK** -- Ranking of task as determined by CA IDMS Task Analyzer.

**TASK CODE** -- The identifying number of the task reported.

**TASK NUMBER** -- Number of the task within the date and time interval selected.

**TASK VER** -- Version of the task. Multiple versions of a task are reported separately.

**TYPE** -- Type of task performed, indicating the language of the program the task invokes.

- **ASSEM**: Assembler
- **COBOL**: COBOL
- **ADSO**: CA ADS
- **PL1**: PL/1
- **FTRAN**: Fortran

**ORIGIN** -- Origin of the task. The operating system or environment where execution of the task originated.

- **IDMS**: CA IDMS/DC
- **CICS**: CICS
- **VM**: z/VM
- **BTCH**: batch

**START DATE-TIME** -- The start date and time of the task being reported.

**VALUE** -- Value of WHAT for the task attribute, (seconds, percentage, etc.).

**CV NUMBER** -- The number of the CV that the statistics on this report apply to.
### PLAN ID -- The statistics plan ID that the statistics on this report apply to.

<table>
<thead>
<tr>
<th>ID</th>
<th>TASK ANALYZER</th>
<th>RELEASE</th>
<th>DATE</th>
<th>TIME</th>
<th>CA-IDMS</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>nn</td>
<td></td>
<td></td>
<td>Rnn.</td>
<td></td>
</tr>
<tr>
<td>/yy</td>
<td></td>
<td>/yy</td>
<td></td>
<td></td>
<td>hh:mm:s</td>
<td></td>
</tr>
</tbody>
</table>

CV NUMBER: 19 PLAN ID: PLAN0001

<table>
<thead>
<tr>
<th>REQUESTED:</th>
<th>mm/dd/yy</th>
<th>hh:mm</th>
<th>mm/dd</th>
<th>ACTUAL:</th>
<th>mm/dd/yy</th>
<th>hh:mm</th>
<th>mm/dd</th>
</tr>
</thead>
</table>

### SELECTION TYPE - WAIT TIME

<table>
<thead>
<tr>
<th>SELECTION</th>
<th>ABSOLUTE VALUE</th>
</tr>
</thead>
</table>

### TASK ANALYZER Input Parameter Report

<table>
<thead>
<tr>
<th>TASK</th>
<th>START CODE</th>
<th>NUMBER</th>
<th>VER</th>
<th>TYPE</th>
<th>ORIGIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SCCULP</td>
<td>70</td>
<td>0</td>
<td>ADSO</td>
<td>BTCH</td>
</tr>
<tr>
<td>2</td>
<td>MAKLOG</td>
<td>150</td>
<td>0</td>
<td>ADSO</td>
<td>BTCH</td>
</tr>
<tr>
<td>3</td>
<td>OPER</td>
<td>17</td>
<td>1</td>
<td>ASSEM</td>
<td>IDMS</td>
</tr>
<tr>
<td>4</td>
<td>ICDVPL02</td>
<td>344</td>
<td>1</td>
<td>ADSO</td>
<td>IDMS</td>
</tr>
<tr>
<td>5</td>
<td>DML0</td>
<td>299</td>
<td>43</td>
<td>ASSEM</td>
<td>IDMS</td>
</tr>
<tr>
<td>6</td>
<td>10:49:17</td>
<td>68</td>
<td>45</td>
<td>ASSEM</td>
<td>IDMS</td>
</tr>
<tr>
<td>7</td>
<td>LOGD</td>
<td>9:13:06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>ICDVPL02</td>
<td>19:33:58</td>
<td>1</td>
<td>ADSO</td>
<td>IDMS</td>
</tr>
<tr>
<td>9</td>
<td>QUED</td>
<td>3:03:25</td>
<td>2</td>
<td>ASSEM</td>
<td>IDMS</td>
</tr>
<tr>
<td>10</td>
<td>DOI</td>
<td>23:24:54</td>
<td>1</td>
<td>ASSEM</td>
<td>IDMS</td>
</tr>
<tr>
<td>11</td>
<td>LOGD</td>
<td>18:16:03</td>
<td>47</td>
<td>ASSEM</td>
<td>IDMS</td>
</tr>
<tr>
<td>12</td>
<td>OPER</td>
<td>9:08:34</td>
<td>293</td>
<td>ASSEM</td>
<td>IDMS</td>
</tr>
<tr>
<td>13</td>
<td>QUED</td>
<td>10:47:41</td>
<td>2</td>
<td>ASSEM</td>
<td>IDMS</td>
</tr>
<tr>
<td>14</td>
<td>ICDVPL02</td>
<td>8:30:09</td>
<td>140</td>
<td>ADSO</td>
<td>IDMS</td>
</tr>
<tr>
<td>15</td>
<td>MAKL0</td>
<td>10:31:27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>LOGD</td>
<td>18:05:49</td>
<td>186</td>
<td>ASSEM</td>
<td>BTCH</td>
</tr>
<tr>
<td>17</td>
<td>USKEVNT</td>
<td>10:55:28</td>
<td>352</td>
<td>ASSEM</td>
<td>IDMS</td>
</tr>
<tr>
<td>18</td>
<td>LOGD</td>
<td>10:57:38</td>
<td>365</td>
<td>ASSEM</td>
<td>IDMS</td>
</tr>
<tr>
<td>19</td>
<td>ICDVPL01</td>
<td>11:03:42</td>
<td>401</td>
<td>ASSEM</td>
<td>IDMS</td>
</tr>
<tr>
<td>20</td>
<td>USKEVNT</td>
<td>19:20:23</td>
<td>337</td>
<td>ADSO</td>
<td>IDMS</td>
</tr>
</tbody>
</table>

Ranking Report:

---

CA IDMS Task Analyzer Input Parameter Report

Contents
CA IDMS - 19.0

- Report Fields (see page 794)

The CA IDMS Task Analyzer Input Parameter Report contains a list of the parameters input to CA IDMS Task Analyzer and the messages that result from processing.

Report Fields

These fields make up the CA IDMS Task Analyzer Input Parameter Report (see the Input Parameter Report). This report and the fields that appear on the report vary, depending on the parameters input to CA IDMS Task Analyzer.

* -- Processing messages

** -- Input parameters

<table>
<thead>
<tr>
<th>ID</th>
<th>TASK ANALYZER</th>
<th>RELEASE</th>
<th>DATE</th>
<th>CA-IDMS TIME</th>
<th>REPORT</th>
<th>PAGE</th>
<th>PAGE</th>
<th>mm/dd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rnn.nn</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>yy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hh:mm:ss</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

V.+++...1+++....

* USF0001I INPUT PARAMETER STATEMENT...............

*PROCESS DCYSRUN=N IDMSXXXX=N **

*PROCESS CVNUM=0019 PLANID=PLAN0001

REPORT=ABND LEVEL=DET RUTYPE=@RUNAME=@ALL

REPORT=ADSO LEVEL=DET RUTYPE=@RUNAME=@ALL

REPORT=BILL LEVEL=DET RUTYPE=@RUNAME=@ALL

REPORT=INDX LEVEL=DET RUTYPE=@RUNAME=@ALL

REPORT=LOAD LEVEL=DET RUTYPE=@RUNAME=@ALL

REPORT=PROG LEVEL=DET RUTYPE=@RUNAME=@ALL

REPORT=RANK ORDER=D WHAT=CPU

Input Parameter Report:

CA IDMS Task Analyzer Parameters

CA IDMS Task Analyzer is parameter-driven: you control the output by supplying the proper parameters which are used as input to a batch job that extracts the information from the CA IDMS log or SMF file and then formats the information to your specifications. CA IDMS Task Analyzer parameters let you select the report types you want to produce, the level of detail, the kind of detail, the time interval, as well as other useful selections.

- Parameters and Their Uses (see page 795)
- Order of Parameter Statements (see page 795)
- Maximum Number of Reports Possible Per Execution (see page 796)

This section is divided into the following sections:
Parameters and Their Uses

Two parameters control CA IDMS Task Analyzer output: PROCESS and REPORT. The PROCESS parameter initiates CA IDMS Task Analyzer processing. The REPORT parameter specifies which CA IDMS Task Analyzer report is to be printed.

The PROCESS parameter is mandatory and should precede all report parameters. It supplies certain global parameters that initiate all processing performed by CA IDMS Task Analyzer.

The REPORT parameter specifies which type of CA IDMS Task Analyzer report is to be created and defines the data that is to be printed. Up to 210 reports can be requested for each execution of CA IDMS Task Analyzer.

A parameter summary and examples of all statements are shown in Parameter Summary at the end of this section.

Order of Parameter Statements

The PROCESS Statement must be entered first. The spelling of the REPORT statement keywords is not important. The keywords, however, must be entered in the order presented.
Maximum Number of Reports Possible Per Execution

A total of 210 reports can be requested during each execution of CA IDMS Task Analyzer. A total of 30 reports per report group (Billing, Program, CA ADS, Abend, Program Loads, Integrated Index, or Ranking) can be requested during each execution of CA IDMS Task Analyzer. This means you can choose, for example, 30 Program Detail Reports, 30 CA ADS Summary Reports, and 30 Billing System Summary Reports, or any combination of reports and options available through CA IDMS Task Analyzer.

PROCESS

All keywords are written in UPPERCASE. Those portions of the keyword that must be entered are UNDERSCORED. When part of a keyword is not underscored, you may omit it without altering the meaning of the statement. You must, however, enter all values for variables within the columns indicated.

REPORT=BILL

A keyword phrase is made up of a major keyword followed by an equal sign (=), followed by a minor keyword or a variable. A keyword phrase cannot be split between two parameter cards.

NAME=name

Variables appear in lower case. Substitute an appropriate value for each variable if the keyword phrase is required.

[ LEVL=DET ]

Brackets indicate optional keyword phrases. If you omit the entire parameter, CA IDMS Task Analyzer will supply a default value.

/BILL \ ADSO
REPORT = <ABEND>
\INDEX /

Braces enclose two or more options in a column. You must choose one of them. The last option listed in the column is the default value (unless otherwise stated). The PROCESS and REPORT parameters for CA IDMS Task Analyzer are positional: keywords and values must be entered in the designated columns, as identified in the following pages. For examples of correctly entered PROCESS and REPORT statements, see the Parameter Summary.

Billing Report Parameters

Contents
- How RTYPE, RUNAME, and NAME Parameters Interrelate (see page 799)

The Billing Reports are available at three levels: details, summary, and system. To generate these reports, use the parameter syntax listed here.

The parameters for CA IDMS Task Analyzer are positional. The following parameters for a REPORT statement example are:
where:

REPORT = BILL
indicates that CA IDMS Task Analyzer is to create and print a BILLING Report.

Rule: You must enter BILL in columns 8 through 11.

EXAMPLE: REPORT STATEMENT, BILLING REPORT
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----
REPORT=BILL LEVL=DET RUTYPE=D RUNAME=@ALL NAME=* / DET \ LEVL = < SUM > \ SYS /

Use this parameter to specify the level of reporting that you want printed.

- DET -- indicates that you want CA IDMS Task Analyzer to print the Billing Details Report. (A Billing Summary Report and a Billing System Summary Report also will be produced for each time interval.)

- SUM -- indicates that you want a Billing Summary Report by program name. (A Billing System Summary also will be produced for each time interval.)

- SYS -- indicates that you want only system summaries to be produced for each time interval.

Default: DET

Rule: The exact spelling of this keyword (LEVL) is not important. You must, however, enter DET, SUM, or SYS in columns 18 through 20.

/ D \ RUTYPE = < V > B \ @ /

Use this parameter to specify the origin of execution for the tasks you want reported.
• **D** -- indicates that you want CA IDMS Task Analyzer to report on tasks with CA IDMS/DC as the origin of execution.

• **C** -- indicates that you want CA IDMS Task Analyzer to report on tasks with CICS as the origin of execution.

• **V** -- indicates that you want CA IDMS Task Analyzer to report on tasks with z/VM as the origin of execution.

• **B** -- indicates that you want CA IDMS Task Analyzer to report on tasks with Batch as the origin of execution.

• **@** -- indicates that you want CA IDMS Task Analyzer to report on tasks with all of the above origins of execution.

**Default:** @

**Rule:** The exact spelling of this keyword (RUTYPE) is not important. You must, however, enter D, C, V, B, or @ in column 29.

```
/ OPER \  
TERM 
RUNAME = < TASK >
GRUP  
@ALL /
```

An individual task may be identified in a number of ways. The way CA IDMS Task Analyzer identifies a task is determined by three things:

• Whether this is a request for a Billing, Program, CA ADS, Program Loads, Abend, or Integrated Index Report.

• Whether execution of this task originates from an online transaction or a batch transaction.

• Whether this task has been invoked by an operator, from a terminal, by task code, or by a group.

Online and batch tasks can be identified by operator, terminal, task, or group.

• **OPER** -- specifies that tasks are to be identified by operator ID.

• **TERM** -- specifies that tasks are to be identified by logical terminal ID.

• **TASK** -- specifies that tasks are to be specified by task code.

• **GRUP** -- specifies that tasks are to be specified by a pre-defined group ID. For CA IDMS Task Analyzer purposes, group ID is in bytes 17-32 of Installation Code in the USER entity of the dictionary.

• **@ALL** -- specifies that CA IDMS Task Analyzer reports on all tasks invoked by operator ID, terminal ID, task code, and group ID.

**Default:** @ALL
Rule: The exact spelling of this keyword (RUNAME) is not important. You must, however, enter the OPER, TERM, TASK, GRUP, or @ALL in columns 38 through 41.

**NAME = name**

This parameter lets you select only those task records that have a specific (or generic) task name. The field that will contain this name is specified by the RUNAME parameter.

Use this parameter to specify the actual (or generic) operator ID, terminal ID, task code, or group ID that a task must have in order for that task to be selected for analysis on the Billing Report.

**Note:** CA IDMS Task Analyzer will perform generic processing. For example, if you key in an asterisk (*) before a name field (that is, NAME = *ABC), all tasks (as specified by the RUNAME parameter) whose name field begins with ABC will be included in the report.

**Default:** None. When no characters are entered in columns 48 through 63, CA IDMS Task Analyzer searches for an operator, terminal, or group that has "blanks" (no characters) for an ID. If RUNAME=TASK is specified and NAME="blanks" (no characters), CA IDMS Task Analyzer returns an error message.

**Rules:**

- The exact spelling of this keyword (NAME) is not important. You must, however, enter the name of the operator, terminal, task, or group beginning in column 48 and at the maximum ending in column 63.

- A maximum of sixteen characters can be entered for the OPER, TERM, TASK, or GRUP name.

- You can enter an asterisk followed by 0 to 15 characters. When an asterisk is entered followed by blanks (no characters), CA IDMS Task Analyzer reports on all tasks as specified by RUNAME. When an asterisk followed by 1 to 15 characters is specified, CA IDMS Task Analyzer reports on all tasks with names beginning with the 1 to 15 characters.

How RUTYPE, RUNAME, and NAME Parameters Interrelate

RUTYPE indicates what processing environment is the origin of execution for the task you want CA IDMS Task Analyzer to report on; the default for RUTYPE is @, indicating CA IDMS Task Analyzer is to report on tasks executing in the processing environments CA IDMS/DC, CICS, z/VM, and batch.

RUNAME indicates the category of tasks you want CA IDMS Task Analyzer to report on, that is, by operator, terminal, task code, or group. The default for RUNAME is @ALL, indicating CA IDMS Task Analyzer is to report on all categories of tasks: operator, terminal, task code, and group. NAME indicates the actual name of the operator, logical terminal, task, or group that you want CA IDMS Task Analyzer to report on; there is no default for the NAME parameter (see the explanation under NAME).

When RUTYPE=@ is specified (or selected by default), CA IDMS Task Analyzer reports on all tasks originating from the four processing environments (CA IDMS/DC, CICS, z/VM, and Batch) for the specified RUNAME and NAME. When RUNAME=@ALL is specified (or selected by default), CA IDMS Task Analyzer reports on all tasks originating in all processing environments previously specified, for all four categories of tasks: operator ID, logical terminal ID, task code, or group ID. Also when
RUNAME=@ALL is specified, CA IDMS Task Analyzer overrides all characters specified after the NAME parameter, and reports on all occurrences of all tasks for all four categories of tasks. When NAME=@ is specified, CA IDMS Task Analyzer reports on all occurrences of tasks identified by the RUNAME.

Program Report Parameter

The Program Reports are available at three levels: details, summary, and system. To generate these reports, use the parameter syntax listed here.

The parameters for CA IDMS Task Analyzer are *positional*. The following example shows a REPORT statement example:

```
REPORT = PROG
\ / DET \nLEVL = < < SUM >
\ \ SYS /
\ / D \nC
\ / RUTYPE = < < V >
\ \ B
\ \ @ /
\ / OPER \nTERM
RUNAME = < < TASK >
\ \ GRUP
\ \ @ALL /
\ [NAME = name]
```

where:

**REPORT = PROG**

indicates that CA IDMS Task Analyzer is to create and print a PROGRAM Report.

**Rule:** You must enter PROG in columns 8 through 11.

**EXAMPLE:** REPORT STATEMENT, PROGRAM REPORT

---

REPORT=PROG LEVL=SUM RUTYPE=C RUNAME=GRUP NAME=EXAMY

\ / DET \ LEVL = < < SUM > \ SYS /

Use this parameter to specify the level of reporting that you want printed.

- **DET** -- indicates that you want CA IDMS Task Analyzer to print the Program Details Report. (A Program Summary Report and a Program System Summary Report also will be produced for each time interval.)

- **SUM** -- indicates that you want a Program Summary Report by program name. (A Program System Summary also will be produced for each time interval.)

- **SYS** -- indicates that you want only system summaries to be produced for each time interval.
Default: DET

Rule: The exact spelling of this keyword (LEVL) is not important. You must, however, enter DET, SUM, or SYS in columns 18 through 20.

\ / D \ C
RUTYPE = < V >
\ / B
\ @ /

Use this parameter to specify the origin of execution for the tasks you want reported.

- **D** -- indicates that you want CA IDMS Task Analyzer to report on the first program in tasks with CA IDMS/DC as the origin of execution.
- **C** -- indicates that you want CA IDMS Task Analyzer to report on the first program in tasks with CICS as the origin of execution.
- **V** -- indicates that you want CA IDMS Task Analyzer to report on the first program in tasks with z/VM as the origin of execution.
- **B** -- indicates that you want CA IDMS Task Analyzer to report on the first program in tasks with Batch as the origin of execution.
- **@** -- indicates that you want CA IDMS Task Analyzer to report on the first program in tasks with all of the above origins of execution.

Default: @

Rule: The exact spelling of this keyword (RUTYPE) is not important. You must, however, enter D, C, V, B, or @ in column 29.

\ / OPER \ TERM
RUNAME = < TASK >
\ @ALL /

An individual task may be identified in a number of ways. The way CA IDMS Task Analyzer identifies a task is determined by three things:

- Whether this is a request for a Billing, Program, CA ADS, Program Loads, Abend, or Integrated Index Report.
- Whether execution of this task originates from an online transaction or a batch transaction.
- Whether this task has been invoked by an operator, from a terminal, by task code, or by a group.

Online and batch tasks can be identified by operator, terminal, task, or group.

- **OPER** -- specifies that tasks are to be identified by operator ID.
- **TERM** -- specifies that tasks are to be identified by logical terminal ID.
- **TASK** -- specifies that tasks are to be specified by task code.
CA IDMS - 19.0

- **GRUP** -- specifies that tasks are to be specified by a pre-defined group ID. For CA IDMS Task Analyzer purposes, group ID is in bytes 17-32 of Installation Code in the USER entity of the dictionary.

- **@ALL** -- specifies that CA IDMS Task Analyzer reports on all tasks invoked by operator ID, terminal ID, task code, and group ID.

Rule: The exact spelling of this keyword (RUNAME) is not important. You must, however, enter OPER, TERM, TASK, GRUP, or @ALL in columns 38 through 41.

**NAME = name**

This parameter lets you select only those task records that have a specific (or generic) task name. The field that is to contain this name is specified by the RUNAME parameter.

Use this parameter to specify the actual (or generic) operator ID, logical terminal ID, task code, or group ID that a task must have in order for that task to be selected for analysis on the Program Report.

⚠️ **Note:** CA IDMS Task Analyzer will perform generic processing. For example, if you key in an asterisk (*) before a name field (that is, NAME = *ABC), all tasks (as specified by the RUNAME parameter) whose name field begins with ABC will be included in the report.

**Default:** None. When no characters are entered in columns 48 through 63, CA IDMS Task Analyzer searches for an operator, terminal, or group that has "blanks" (no characters) for an ID. If RUNAME=TASK is specified and NAME="blanks" (no characters), CA IDMS Task Analyzer returns an error message.

**Rules:**

- The exact spelling of this keyword (NAME) is not important. You must, however, enter the name of the operator, terminal, task, or group beginning in column 48 and at the maximum ending in column 63.

- A maximum of sixteen characters can be entered for the OPER, TERM, TASK, or GRUP name.

- You can enter an asterisk followed by 0 to 15 characters. When an asterisk is entered followed by blanks (no characters), CA IDMS Task Analyzer reports on all tasks as specified by RUNAME. When an asterisk followed by 1 to 15 characters is specified, CA IDMS Task Analyzer reports on all tasks with names beginning with the 1 to 15 characters.

---

**CA ADS Report Parameters**

The CA ADS Reports are available at three levels: details, summary, and system. To generate these reports, use the parameter syntax listed here.

The parameters for CA IDMS Task Analyzer are **positional**. See the following parameters for a REPORT statement example.
The parameters are:

```
REPORT = ADSO
\ / DET \ LEVL = < SUM > \ SYS / \ D \ C
RUTYPE = < V > \ @ / \ OPER \ TERM
RUNAME = < TASK > GRUP \ @ALL / [NAME = name]
```

where:

**REPORT = ADSO**

indicates that CA IDMS Task Analyzer is to create and print an CA ADS Report.

Rule: You must enter ADSO in columns 8 through 11.

**EXAMPLE:** REPORT STATEMENT, CA ADS
```
- - - - - - - 1 - - - - - - 2 - - - - - - 3 - - - - - - 4 - - - - - - - - - - - 5 - - - - - - - - - - - - 6 - - - - - - - - - - - 7 -
REPORT=ADSO LEVL=SYS RUTYPE=V RUNAME=TASK NAME=EPOSED
\ / DET \ LEVL = < SUM > \ SYS /
```

Use this parameter to specify the level of reporting that you want printed.

- **DET** -- indicates that you want CA IDMS Task Analyzer to print the CA ADS Details Report. (A CA ADS Summary Report and a CA ADS System Summary Report will also be produced for each time interval.)

- **SUM** -- indicates that you want an CA ADS Summary Report by dialog name. (A CA ADS System Summary Report will also be produced for each time interval.)

- **SYS** -- indicates that you want only system summaries to be produced for each time interval.

**Default:** DET

Rule: The exact spelling of this keyword (LEVL) is not important. You must, however, enter DET, SUM, or SYS in columns 18 through 20.

```
\ / DET \ LEVL = < SUM > \ SYS /
```

Use this parameter to specify the origin of execution for the tasks (that have as their first program an CA ADS dialog) you want reported.
• D -- indicates that you want CA IDMS Task Analyzer to report on the first CA ADS dialog in tasks with CA IDMS/DC as the origin of execution.

• C -- indicates that you want CA IDMS Task Analyzer to report on the first CA ADS dialog in tasks with CICS as the origin of execution.

• V -- indicates that you want CA IDMS Task Analyzer to report on the first CA ADS dialog in tasks with z/VM as the origin of execution.

• @ -- indicates that you want CA IDMS Task Analyzer to report on the first CA ADS dialog in tasks with all of the above origins of execution.

Default: @

Rule: The exact spelling of this keyword (RUTYPE) is not important. You must, however, enter D, C, V, or @ in column 29.

/ OPER \  
TERM
RUNAME = < TASK >  
GRUP  
@ALL /

An individual task may be identified in a number of ways. How CA IDMS Task Analyzer identifies a task is determined by three things:

• Whether this is a request for a Billing, Program, CA ADS, Program Loads, Abend, or Integrated Index Report.

• Whether execution of this task originates from an online transaction or a batch transaction.

• Whether this task has been invoked by an operator, from a terminal, by task code, or by a group.

Online and batch tasks can be identified by operator, terminal, task, or group.

• OPER -- specifies that tasks are to be identified by operator ID.

• TERM -- specifies that tasks are to be identified by logical terminal ID.

• TASK -- specifies that tasks are to be specified by task code.

• GRUP -- specifies that tasks are to be specified by a pre-defined group ID. For CA IDMS Task Analyzer purposes, group ID is in bytes 17-32 of Installation Code in the USER entity of the dictionary.

• @ALL -- specifies that CA IDMS Task Analyzer reports on all tasks invoked by operator ID, terminal ID, task code, and group ID.

Default: @ALL

Rule: The exact spelling of this keyword (RUNAME) is not important. You must, however, enter OPER, TERM, TASK, GRUP, or @ALL in columns 38 through 41.
NAME = name

This parameter lets you select only those task records that have a specific (or generic) task name. The field that will contain this name is specified by the RUNAME parameter.

Use this parameter to specify the actual (or generic) operator ID, logical terminal ID, task code, or group ID that a task must have in order for that task to be selected for analysis on the CA ADS Report.

Note: CA IDMS Task Analyzer will perform generic processing. For example, if you key in an asterisk (*) before a name field (that is, NAME = *ABC), all tasks (as specified by the RUNAME parameter) whose name field begins with ABC will be included in the report.

Default: None. When no characters are entered in columns 48 through 63, CA IDMS Task Analyzer searches for an operator, terminal, or group that has "blanks" (no characters) for an ID. If RUNAME=TASK is specified and NAME="blanks" (no characters), CA IDMS Task Analyzer returns an error message.

Rules:

- The exact spelling of this keyword (NAME) is not important. You must, however, enter the name of the operator, terminal, task, or group beginning in column 48 and at the maximum ending in column 63.
- A maximum of sixteen characters can be entered for the OPER, TERM, TASK, or GRUP name.
- You can enter an asterisk followed by 0 to 15 characters. When an asterisk is entered followed by blanks (no characters), CA IDMS Task Analyzer reports on all tasks as specified by RUNAME. When an asterisk followed by 1 to 15 characters is specified, CA IDMS Task Analyzer reports on all tasks with names beginning with the 1 to 15 characters.

Abend Report Parameters

The Abend Report is available at one level: detail. To generate this report, use the parameter syntax listed on this page.

The parameters for CA IDMS Task Analyzer are positional. See the following parameters for a REPORT statement example.

The parameters are:

```
REPORT = ABND
  [LEVL = DET ]
  / D \
  / C
RUTYPE = < V >
  B
  \ @ /
  / OPER \ T
```
RUNAME = < TASK >
\ @ALL /

(NAME = name)

where:

REPORT = ABND

indicates that CA IDMS Task Analyzer is to create and print an ABEND Report.

Rule: You must enter ABND in columns 8 through 11.

EXAMPLE: REPORT STATEMENT, ABEND REPORT
- - - - - - - - 1 - - - - - - - - 2 - - - - - - - - 3 - - - - - - - - 4 - - - - - - - - 5 - - - - - - - - 6 - - - - - - - - 7 -
REPORT=ABND LEVL=DET RUTYPE=B RUNAME=@ALL NAME=TERKK

LEVL = DET

Use this parameter to specify that you want printed the Abend Details Report.

- DET -- indicates that you want CA IDMS Task Analyzer to print the Abend Details Report.

Default: DET

Rule: The exact spelling of this keyword (LEVL) is not important. You must, however, enter DET in columns 18 through 20.

/ D \
C

RUTYPE = < V >
B
\ @ /

Use this parameter to specify the origin of execution for the tasks you want reported. A report is generated for the specified tasks if they abend while processing under CA IDMS, even though processing began within CICS, z/VM, or Batch.

- D -- indicates that you want CA IDMS Task Analyzer to report on the tasks with CA IDMS/DC as the origin of execution.

- C -- indicates that you want CA IDMS Task Analyzer to report on the tasks with CICS as the origin of execution.

- V -- indicates that you want CA IDMS Task Analyzer to report on the tasks with z/VM as the origin of execution.

- B -- indicates that you want CA IDMS Task Analyzer to report on the tasks with Batch as the origin of execution.

- @ -- indicates that you want CA IDMS Task Analyzer to report on the tasks with all of the above origins of execution.

Default: @
Rule: The exact spelling of this keyword (RUTYPE) is not important. You must, however, enter D, C, V, B, or @ in column 29.

```
/ OPER \  
TERM

RUNAME = < TASK >  
GRUP  
\ @ALL /
```

- **OPER** -- specifies that tasks are to be identified by operator ID.
- **TERM** -- specifies that tasks are to be identified by logical terminal ID.
- **TASK** -- specifies that tasks are to be specified by task code.
- **GRUP** -- specifies that tasks are to be specified by a pre-defined group ID. For CA IDMS Task Analyzer purposes, group ID is in bytes 17-32 of Installation Code in the USER entity of the dictionary.
- **@ALL** -- specifies that CA IDMS Task Analyzer reports on all tasks invoked by operator ID, terminal ID, task code, and group ID.

**Default:** @ALL

Rule: The exact spelling of this keyword (RUNAME) is not important. You must, however, enter OPER, TERM, TASK, GRUP, or @ALL in columns 38 through 41.

**NAME = name**

This parameter lets you select only those task records that have a specific (or generic) task name. The field that will contain this name is specified by the RUNAME parameter.

Use this parameter to specify the actual (or generic) operator ID, logical terminal ID, task code, or group ID that a task must have in order for that task to be selected for analysis on the Abend Report.

⚠️ **Note:** CA IDMS Task Analyzer will perform generic processing. For example, if you key in an asterisk (*) before a name field (that is, NAME = *ABC), all tasks (as specified by the RUNAME parameter) whose name field begins with ABC will be included in the report.

**Default:** None. When no characters are entered in columns 48 through 63, CA IDMS Task Analyzer searches for an operator, terminal, or group that has "blanks" (no characters) for an ID. If RUNAME=TASK is specified and NAME="blanks" (no characters), CA IDMS Task Analyzer returns an error message.

**Rules:**

- The exact spelling of this keyword (NAME) is not important. You must, however, enter the name of the operator, terminal, task, or group beginning in column 48 and at the maximum ending in column 63.
A maximum of sixteen characters can be entered for the OPER, TERM, TASK, or GRUP name.

You can enter an asterisk followed by 0 to 15 characters. When an asterisk is entered followed by blanks (no characters), CA IDMS Task Analyzer reports on all tasks as specified by RUNAME. When an asterisk followed by 1 to 15 characters is specified, CA IDMS Task Analyzer reports on all tasks with names beginning with the 1 to 15 characters.

Program Loads Report Parameters

The Program Loads Report is available at one level: detail. To generate this report, use the parameter syntax listed on this page.

The parameters for CA IDMS Task Analyzer are *positional*. See the following parameters from a REPORT statement example.

The parameters are:

\[
\text{REPORT} = \text{LOAD} \\
\text{[LEVL = DET ]} \\
\text{/ D \} \\
\text{C} \\
\text{RUTYPE} = < V > \\
\text{B} \\
\text{\ @ /} \\
\text{/ OPER \} \\
\text{TERM} \\
\text{RUNAME} = < \text{TASK} > \\
\text{GRUP} \\
\text{\ @ALL /} \\
\text{[NAME = name]}
\]

where:

\text{REPORT} = \text{LOAD}

indicates that CA IDMS Task Analyzer is to create and print a Program Loads Report.

Rule: You must enter LOAD in columns 8 through 11.

\text{EXAMPLE: REPORT STATEMENT, PROGRAM LOADS REPORT}

\text{REPORT=LOAD LEVL=DET RUTYPE=@ RUNAME=OPER NAME=KIKKS}

\text{LEVL = DET}

Use this parameter to specify that you want printed the Program Loads Details Report.

- \text{DET} -- indicates that you want CA IDMS Task Analyzer to print the Program Loads Details Report.

\text{Default: DET}

Rule: The exact spelling of this keyword (LEVL) is not important. You must, however, enter DET in columns 18 through 20.
CA IDMS - 19.0

Use this parameter to specify the origin of execution for the tasks you want reported.

- D -- indicates that you want CA IDMS Task Analyzer to report on the tasks with CA IDMS/DC as the origin of execution.
- C -- indicates that you want CA IDMS Task Analyzer to report on the tasks with CICS as the origin of execution.
- V -- indicates that you want CA IDMS Task Analyzer to report on the tasks with z/VM as the origin of execution.
- B -- indicates that you want CA IDMS Task Analyzer to report on the tasks with Batch as the origin of execution.
- @ -- indicates that you want CA IDMS Task Analyzer to report on the tasks with all of the above origins of execution.

Default: @

Rule: The exact spelling of this keyword (RUTYPE) is not important. You must, however, enter D, C, V, B, or @ in column 29.

OPER -- specifies that tasks are to be identified by operator ID.

TERM -- specifies that tasks are to be identified by logical terminal ID.

TASK -- specifies that tasks are to be specified by task code.

GRUP -- specifies that tasks are to be specified by a pre-defined group ID. For CA IDMS Task Analyzer purposes, group ID is in bytes 17-32 of Installation Code in the USER entity of the dictionary.

@ALL -- specifies that CA IDMS Task Analyzer reports on all tasks invoked by operator ID, terminal ID, task code, and group ID.

Default: @ALL

Rule: The exact spelling of this keyword (RUNAME) is not important. You must, however, enter OPER, TERM, TASK, GRUP, or @ALL in columns 38 through 41.

NAME = name

This parameter lets you select only those task records that have a specific (or generic) task name. The field that will contain this name is specified by the RUNAME parameter.
Use this parameter to specify the actual (or generic) operator ID, logical terminal ID, task code, or group ID that a task must have in order for that task to be selected for analysis on the Program Loads Report.

**Note:** CA IDMS Task Analyzer will perform generic processing. For example, if you key in an asterisk (*) before a name field (that is, NAME = *ABC), all tasks (as specified by the RUNAME parameter) whose name field begins with ABC will be included in the report.

**Default:** None. When no characters are entered in columns 48 through 63, CA IDMS Task Analyzer searches for an operator, terminal, or group that has "blanks" (no characters) for an ID. If RUNAME=TASK is specified and NAME="blanks" (no characters), CA IDMS Task Analyzer returns an error message.

**Rules:**

- The exact spelling of this keyword (NAME) is not important. You must, however, enter the name of the operator, terminal, task, or group beginning in column 48 and at the maximum ending in column 63.

- A maximum of sixteen characters can be entered for the OPER, TERM, TASK, or GRUP name.

- You can enter an asterisk followed by 0 to 15 characters. When an asterisk is entered followed by blanks (no characters), CA IDMS Task Analyzer reports on all tasks as specified by RUNAME. When an asterisk followed by 1 to 15 characters is specified, CA IDMS Task Analyzer reports on all tasks with names beginning with the 1 to 15 characters.

### Integrated Index Report Parameters

The Integrated Index Reports are available at three levels: details, summary, and system. To generate these reports, use the parameter syntax listed here.

The parameters for CA IDMS Task Analyzer are *positional*. See the following parameters for a REPORT statement example.

The parameters are:

```
REPORT = INDEX
/ DET \
LEVL = < SUM >
/ SYS /
/ D \\C
RUTYPE = < V >
/ B \ @ /
/ OPER \\ 
TERM
RUNAME = < TASK >
```

\ @ALL /  

[NAME = name]  

where:  

**REPORT = INDX**  

indicates that CA IDMS Task Analyzer is to create and print an Integrated Index Report.  

Rule: You must enter INDX in columns 8 through 11.  

**EXAMPLE:**  REPORT STATEMENT, INTEGRATED INDEX REPORT  

--+-+---1---+---2---+---3---+---4---+---5---+---6---+---7-  
REPORT=INDX LEVL=SUM RUTYPE=D RUNAME=@ALL NAME=*  
/ DET \ LEVL = < SUM > \ SYS /  

Use this parameter to specify the level of reporting that you want printed.  

- **DET** -- indicates that you want CA IDMS Task Analyzer to print the Integrated Index Details Report.  
  (An Integrated Index Summary Report and an Integrated Index System Summary Report also will  
  be produced for each time interval.)  

- **SUM** -- indicates that you want an Integrated Index Summary Report by program name. (An  
  Integrated Index System Summary also will be produced for each time interval.)  

- **SYS** -- indicates that you want only system summaries to be produced for each time interval.  

Default: DET  

Rule: The exact spelling of this keyword (LEVL) is not important. You must, however, enter DET, SUM,  

or SYS in columns 18 through 20.  

/ D \  
C  

**RUTYPE = < V >**  

/ @ /  

Use this parameter to specify the origin of execution for the tasks you want reported.  

- **D** -- indicates that you want CA IDMS Task Analyzer to report on tasks with CA IDMS/DC as the  
  origin of execution.  

- **C** -- indicates that you want CA IDMS Task Analyzer to report on tasks with CICS as the origin of  
  execution.  

- **V** -- indicates that you want CA IDMS Task Analyzer to report on tasks with z/VM as the origin of  
  execution.  

- **B** -- indicates that you want CA IDMS Task Analyzer to report on tasks with Batch as the origin of  
  execution.  

- **@** -- indicates that you want CA IDMS Task Analyzer to report on tasks with all of the above  
  origins of execution.
Default: @

Rule: The exact spelling of this keyword (RUTYPE) is not important. You must, however, enter D, C, V, B, or @ in column 29.

\ / OPER \ \
TERM

RUNAME = < TASK >
GRUP
\ @ALL /

- OPER -- specifies that tasks are to be identified by operator ID.
- TERM -- specifies that tasks are to be identified by logical terminal ID.
- TASK -- specifies that tasks are to be specified by task code.
- GRUP -- specifies that tasks are to be specified by a pre-defined group ID. For CA IDMS Task Analyzer purposes, group ID is in bytes 17-32 of Installation Code in the USER entity of the dictionary.
- @ALL -- specifies that CA IDMS Task Analyzer reports on all tasks invoked by operator ID, terminal ID, task code, and group ID.

Default: @ALL

Rule: The exact spelling of this keyword (RUNAME) is not important. You must, however, enter OPER, TERM, TASK, GRUP, or @ALL in columns 38 through 41.

NAME = name

This parameter lets you select only those task records that have a specific (or generic) task name. The field that will contain this name is specified by the RUNAME parameter.

Use this parameter to specify the actual (or generic) operator ID, logical terminal ID, task code, or group ID that a task must have in order for that task to be selected for analysis on the Integrated Index Report.

**Note:** CA IDMS Task Analyzer will perform generic processing. For example, if you key in an asterisk (*) before a name field (that is, NAME = *ABC), all tasks (as specified by the RUNAME parameter) whose name field begins with ABC will be included in the report.

Default: None. When no characters are entered in columns 48 through 63, CA IDMS Task Analyzer searches for an operator, terminal, or group that has "blanks" (no characters) for an ID. If RUNAME=TASK is specified and NAME="blanks" (no characters), CA IDMS Task Analyzer returns an error message.

Rules:
The exact spelling of this keyword (NAME) is not important. You must, however, enter the name of the operator, terminal, task, or group beginning in column 48 and at the maximum ending in column 63.

A maximum of sixteen characters can be entered for the OPER, TERM, TASK, or GRUP name.

You can enter an asterisk followed by 0 to 15 characters. When an asterisk is entered followed by blanks (no characters), CA IDMS Task Analyzer reports on all tasks as specified by RUNAME. When an asterisk followed by 1 to 15 characters is specified, CA IDMS Task Analyzer reports on all tasks with names beginning with the 1 to 15 characters.

Ranking Report Parameters

To generate Ranking Reports, use the parameter syntax listed here.

The parameters for CA IDMS Task Analyzer are positional. The following is an example of a Ranking REPORT statement example.

The parameters are:

\begin{itemize}
  \item \texttt{REPORT = RANK}
  \item \texttt{ORDER = < A > \ D /}
  \item \texttt{[NUMBER= nnn]}
  \item \texttt{[WHAT = item]}
  \item \texttt{[VALUE = nnnnnnnnnnnn]}
\end{itemize}

where:

\begin{itemize}
  \item \texttt{REPORT = RANK}
\end{itemize}

indicates that CA IDMS Task Analyzer is to create and print a Ranking Report.

Rule: You must enter RANK in columns 8 through 11.

\begin{verbatim}
EXAMPLE: REPORT STATEMENT, RANKING REPORT
---+---1---+---2---+---3---+---4---+---5---+---6---+---7---
REPORT=RANK ORDER=D NUMBER=050 HOW=GT WHAT=TERMREAD VALUE=0000000001000
ORDER = < A > \ D /
\end{verbatim}

Use this parameter to specify the order in which the tasks are to be reported.
A -- indicates that you want CA IDMS Task Analyzer to print the Ranking Report in ascending order, with the task with the lowest value printed first and the task with the highest value printed last.

D -- indicates that you want CA IDMS Task Analyzer to print the Ranking Report in descending order, with the task with the highest value printed first and the task with the lowest value printed last.

Default: A (for ascending)

Rule: The exact spelling of this keyword (ORDER) is not important. You must, however, enter A or D in column 19.

NUMBER=nnn

After all of the tasks have been selected and ranked, you may also request how many you want to see on the Ranking Report. For example, if you only want to see the first 10 when ranked according to your ORDER and HOW parameters, specify NUMBER = 010.

Use this parameter to specify the number of items to be reported on the Ranking Report.

Default: 020 (that is, 20 tasks will be listed on the report)

Rules:

- The maximum number of tasks that can be reported is 999.
- The exact spelling of this keyword (NUMBER) is not important. You must, however, enter a three digit number in columns 28 through 30.
- You must include leading zeroes (’0’s). For example, if you want to indicate 5 tasks, you must specify ”005”.

/ AB \  
EQ \  
HOW = < LT >  
LE \ GT  
\ GE /

Use this keyword to specify how the attribute you select is to be ranked. There are six methods to choose from.

- AB -- indicates that tasks are to be ranked by the value of the attribute from the specified task. If you specify AB, you will obtain a report on all tasks compared by the item following the WHAT parameter.
- EQ -- the attribute will be ranked from a value EQUAL to the specified value.
- LT -- the attribute will be ranked from a value LESS THAN the specified value.
- LE -- the attribute will be ranked from a value LESS than or EQUAL to the specified value.
- GT -- the attribute will be ranked from a value GREATER THAN the specified value.
- **GE** -- the attribute will be ranked from a value GREATER than or EQUAL to the specified value.

**Default:** AB (for ABSOLUTE)

Rule: The exact spelling of this keyword is not important. You must, however, enter **AB, EQ, LT, LE, GT**, or **GE** in columns 36 and 37.

**WHAT = item**

specifies which attribute is to be ranked in the report (see the following table).

**Default:** None. You **must** select one of the available items.

Rules:

- The exact spelling of this keyword (WHAT) is not important. You must, however, enter the item in columns 44 through 52.

- You **must** spell the item exactly as it is presented in the following table.

<table>
<thead>
<tr>
<th>Item</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTREE</td>
<td>Number of integrated index searches</td>
</tr>
<tr>
<td>BTREELEVL</td>
<td>Number of levels required to complete a search</td>
</tr>
<tr>
<td>CLOCK</td>
<td>Clock time</td>
</tr>
<tr>
<td>CPU</td>
<td>CPU time</td>
</tr>
<tr>
<td>DBCALL</td>
<td>DML verbs</td>
</tr>
<tr>
<td>DETPUTNEW</td>
<td>New details written ( pageable maps)</td>
</tr>
<tr>
<td>DETPUTCUR</td>
<td>Current details written ( pageable maps)</td>
</tr>
<tr>
<td>DETGET</td>
<td>Details read ( pageable maps)</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>Number of display commands</td>
</tr>
<tr>
<td>INVOKE</td>
<td>Number of invoke commands</td>
</tr>
<tr>
<td>I/O</td>
<td>Physical I/Os</td>
</tr>
<tr>
<td>LEAVE</td>
<td>Number of leave commands</td>
</tr>
<tr>
<td>LINK</td>
<td>Number of link commands</td>
</tr>
<tr>
<td>LINKMAX</td>
<td>Maximum number of links</td>
</tr>
<tr>
<td>LINKMIN</td>
<td>Minimum number of links</td>
</tr>
<tr>
<td>ORPHANS</td>
<td>SR8 orphans adopted</td>
</tr>
<tr>
<td>PREMAP</td>
<td>Premap processes</td>
</tr>
<tr>
<td>QUEUEDEL</td>
<td>Records deleted ( queue)</td>
</tr>
<tr>
<td>QUEUEGET</td>
<td>Records read ( queue)</td>
</tr>
<tr>
<td>QUEUEPUT</td>
<td>Records written ( queue)</td>
</tr>
<tr>
<td>RBBMAX</td>
<td>Maximum size of record buffer blocks</td>
</tr>
<tr>
<td>RBBMIN</td>
<td>Minimum size of record buffer blocks</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>RESPONSE</td>
<td>Response processes</td>
</tr>
<tr>
<td>RETURN</td>
<td>Number of return commands</td>
</tr>
<tr>
<td>SCRTCHDEL</td>
<td>Records deleted (scratch)</td>
</tr>
<tr>
<td>SCRTCHGET</td>
<td>Records read (scratch)</td>
</tr>
<tr>
<td>SCRTCHPUT</td>
<td>Records written (scratch)</td>
</tr>
<tr>
<td>SR7ERASED</td>
<td>SR7s erased</td>
</tr>
<tr>
<td>SR7STORED</td>
<td>SR7s stored</td>
</tr>
<tr>
<td>SR8ERASED</td>
<td>SR8s erased</td>
</tr>
<tr>
<td>SR8SPAWN</td>
<td>SR8 spawns occurring</td>
</tr>
<tr>
<td>SR8SPLIT</td>
<td>SR8 splits occurring</td>
</tr>
<tr>
<td>SR8STORED</td>
<td>SR8s stored</td>
</tr>
<tr>
<td>STORACQUR</td>
<td>Storage acquired</td>
</tr>
<tr>
<td>STORALLOC</td>
<td>Storage allocated</td>
</tr>
<tr>
<td>STORKEPT</td>
<td>Storage kept</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>System time</td>
</tr>
<tr>
<td>TERMERROR</td>
<td>Terminal errors</td>
</tr>
<tr>
<td>TERMI/O</td>
<td>Terminal I/Os</td>
</tr>
<tr>
<td>TERMREAD</td>
<td>Terminal reads</td>
</tr>
<tr>
<td>TERMWRITE</td>
<td>Terminal writes</td>
</tr>
<tr>
<td>TRANSFER</td>
<td>Number of transfer commands</td>
</tr>
<tr>
<td>USER</td>
<td>User time</td>
</tr>
<tr>
<td>WAIT</td>
<td>CPU wait time</td>
</tr>
</tbody>
</table>

**VALUE = nnnnnnnnnnnn**

indicates what numeric value is to be used for comparison, where

- **nnnnnnnnnnnn** represents a twelve digit number. The number can indicate quantity or time, depending upon the attribute specified after the **WHAT** parameter (columns 44 through 52). When time is indicated, units are presented in 1/10,000ths of a second.

**Default:** blank. This indicates that all tasks are to be ranked by the ABSOLUTE value, as specified after the **HOW** parameter (columns 36 and 37).

**Rules:**

- The exact spelling of this keyword (VALUE) is not important. You must, however, enter a twelve digit number in columns 60 through 71.

- You must include leading and trailing zeroes ('0's). For example, if you want to indicate 10 seconds, you must specify "000000100000".
If you leave the columns 60 through 71 blank, then you must specify after the **HOW** parameter (columns 36 and 37); or, leave columns 36 and 37 blank. Otherwise, no ranking report will be generated.

- Do **not** include commas or decimal points.

### Parameter Summary:

```plaintext
PROCESS [DCSYSRUN={(Y)}] [IDMSXXXX={(Y)}] [START=mmddyyhhmm] [STOP=mmddyyhhmm]

[*PROCESS] [CVNUM={(ALL)}] [PLANID={(ALL)}]

[BILL] [LEVL={(SUM)}] [RUTYPE={(V)}] [RUNAME={(TASK)}] [NAME=name]

[PROGREP] [LEVL={(SUM)}] [RUTYPE={(V)}] [RUNAME={(TASK)}] [NAME=name]

[ADSO] [LEVL={(SUM)}] [RUTYPE={(V)}] [RUNAME={(TASK)}] [NAME=name]

[ABND] [LEVL=DET] [RUTYPE={(V)}] [RUNAME={(TASK)}] [NAME=name]

[LOAD] [LEVL=DET] [RUTYPE={(V)}] [RUNAME={(TASK)}] [NAME=name]

[INDEX] [LEVL={(SUM)}] [RUTYPE={(V)}] [RUNAME={(TASK)}] [NAME=name]

[RANK] [ORDER={(A)}] [NUMBER=nnn] [HOW={(LT)}] [HOW={(GT)}]

[WHAT=item] [VALUE=nnnnnnnnnnnn]
```

**Note:** In the START and STOP keywords, the year is assumed to be 2000 if you specify less than 69; the year is assumed to be 1900 if you specify 69 or greater.

**EXAMPLE:** PROCESS STATEMENT

```
- - - - - - - 1 - - - - - - 2 - - - - - - 3 - - - - - - 4 - - - - - - 5 - - - - - - 6 - - - - - - 7 -

PROCESS DCSYSRUN=Y IDMSXXXX=Y START=0420862030 STOP=0420862359
```
Task Analyzer Operations

This section describes about z/OS, z/VSE and z/VM operations by the CA IDMS Task Analyzer.

- CA IDMS Task Analyzer Operating Requirements (see page 818)
- CA IDMS Task Analyzer Statistics Plan Options (see page 820)
- Task Analyzer z/OS Operation (see page 823)
- CA IDMS Task Analyzer z/VSE Operations (see page 831)
- CA IDMS Task Analyzer z/VM Operations (see page 835)

CA IDMS Task Analyzer Operating Requirements

The following are the operating requirements of CA IDMS Task Analyzer:

- **Terminal Type**: All 3270-type terminals, models 2 through 5
Storage Requirement:

- **Program Storage:** RHDCUXIT increases by 14K, and USFAOPT (Statistics Plan program) needs 11K of reentrant pool program storage.

- **Storage Pool Usage:** For exit processing, each CA IDMS task acquires the following number of bytes:

  \[
  440 + (128 \times \text{maximum load/links})
  \]

  For the default of 16 maximum load/link levels, this would be 2488 bytes per CA IDMS task. USFAOPT (Statistics Plan program) needs 3656 bytes.

- **Queue Storage:** CA IDMS Task Analyzer uses 120 bytes of queue storage, having a storage ID of CA IDMS Task Analyzer.

- **User Exit Considerations:** If you add or subtract exit 0, 3, 4, 5, 13, or 15 after your initial installation of CA IDMS Task Analyzer, you must regenerate the USFUEXT table to reflect that change. If you add or subtract an exit other than 0, 3, 4, 5, 13, or 15 after your initial installation of CA IDMS Task Analyzer, you must regenerate the CA version of RHDCUXIT to reflect that change. For more information, see the Installation Sections.

- **GSISVCX module customization:** GSISVCX, the CA IDMS TOOLS version of the assembler language module IDMSSVCX is designed to create a 40-position extension to the CA IDMS External Request Element control block (ERE). The type of data that is placed into these 40 positions by the module depends, in part, on whether the run-unit being processed is identified by CA IDMS as BATCH or CICS.

  - **For any type of run-unit** -- the JOBNAME, the run-unit start date and time, and the step start time are moved into the ERE by GSISVCX.

  - **For BATCH run-units** -- up to 16 bytes of information contained in the account field of the jobcard are moved into the ERE by GSISVCX.

  - **For CICS run-units** -- the transaction ID, terminal ID, and operator ID are moved into the ERE by GSISVCX. You will need to customize this module if your installation already uses a version of IDMSSVCX and that function must be retained for continued use: if your account number is not in the first field of the z/OS JOB ACCT parameter; if your installation uses a TP monitor other than CICS; or if the data moved into the ERE is not sufficient for your billing system requirements.

  To customize GSISVCX, your systems programmer must make the desired changes to the source code. For more information, see the Installation Sections.

  When altering the source code for GSISVCX, follow these sectionlines:

  - The ERE may be defined as any length between 40 and 32767 but only the first 40 positions will be written by CA IDMS to the Task Statistics Record.

  - After the GSISVCX source code is modified, the CA IDMS SVC macro must be identified to the CA IDMS SVC. For more information, see the CA IDMS Installation and Maintenance Section.
CA IDMS Task Analyzer Statistics Plan Options

Contents
- Screen Fields (see page 820)
- Enable CA-IDMS Task Analyzer User Exits (see page 822)

CA IDMS Task Analyzer has an online front-end with which you specify options to control the collection and writing of statistics. Selection of these options creates your statistics plan for CA IDMS Task Analyzer.

One of the main features of the statistics plan is the ability to assign a Plan ID to any collection run. If you make changes to CA IDMS/DC (for example, maintenance tapes, new applications) and assign a different Plan ID to the collection run, you can generate reports by the new Plan ID and compare them to reports of other collection runs.

The Statistics Control Planning screen is divided into three areas: collection control, exit control, and write control.

The collection control area lets you specify what statistics are to be collected and what, if any, programs and/or tasks can be excluded from collection.

The exit control area lets you enable or disable CA IDMS/DC user exits in three ways: all user exits, CA IDMS Task Analyzer exits, and specific user exits. In addition, hours of operation can be specified for the CA IDMS Task Analyzer exits as a group.

The write control area lets you specify whether to write statistics to the CA IDMS/DC Log and/or the SMF file, and also gives you several SMF options.

For more information, see the following Task Analyzer Statistics Collection Planning Screen and the descriptions.

To specify statistics collection options, call up the Statistics Collection Planning screen by entering task USFAOPT from the CA IDMS/DC prompt. Then tab to each field you want to change from the default. To update the plan with options you have specified, press ENTER. If errors have been made, a general error message is displayed. To get help on a specific error, press the PF1 key. To exit the screen, press the CLEAR key or the PF3 key.

Screen Fields

Here is a description of the fields that appear on the following Statistics Collection Planning screen.

**PLAN ID** -- Enter 8 alpha-numeric characters to identify the statistics plan. PLAN ID may be used to identify the type of environment that CA IDMS Task Analyzer is operating in. For example, you may want to use PLAN ID to identify the CA IDMS maintenance level currently installed or to identify the type of CA IDMS Task Analyzer processing.

**DATABASE** -- Specify Y (yes) if you want CA IDMS Task Analyzer to collect database statistics including those for integrated indexes or N (no) if you do not want CA IDMS Task Analyzer to collect database statistics.
CA ADS -- Specify Y (yes) if you want CA IDMS Task Analyzer to collect CA ADS statistics or N (no) if you do not want CA IDMS Task Analyzer to collect CA ADS statistics.

PROGRAM -- Specify Y (yes) if you want CA IDMS Task Analyzer to collect program statistics or N (no) if you do not want CA IDMS Task Analyzer to collect program statistics.

Rnn.nn CA-IDMS/Task Analyzer hh:mm mm/dd/yy
USFMPT STATISTICS COLLECTION PLANNING SCREEN PLAN ID: PLAN0001

COLLECTION CONTROL:
DATABASE: Y ADS: Y PROGRAM: N DC/INTERNAL: N
LOAD/LINK: Y ERUS: Y MAXIMUM LOAD/LINK LEVELS: 129

EXCLUDE:
OPTION NAME OPTION NAME OPTION NAME OPTION NAME
P CRUAL _ _

EXIT CONTROL:
GLOBAL STATUS: _ => ACTIVE
COLLECTION STATUS: _ FROM: 00:00 TO: 24:00 => ACTIVE
(SUN/MON/TUE/WED/THU/FRI/SAT) DAY OF WEEK: WED TO FRI
0 EXIT MODE NAME STATUS 0 EXIT MODE NAME STATUS
 04 DC SSK2IT04 ACTIVE 13 (NONE) DISABLED
 05 (NONE) DISABLED 15 (NONE) DISABLED

WRITE CONTROL:
LOG FILE: Y LOG TYPE: 2 DC STATISTICS: Y OS/SMF: N
SMF OPTIONS:
NUMBER: 129 BLOCK MODE: 1 BLOCK SIZE: 04096 JES ID: XE06
AUTHORIZATION MODE: 1 AUTHORIZATION SVC: 247

CA IDMS Task Analyzer Statistics Collection Planning Screen:

DC/INTERNAL -- Specify Y (yes) if you want CA IDMS Task Analyzer to collect CA IDMS/DC internal tasks statistics or N (no) if you do not want CA IDMS Task Analyzer to collect CA IDMS/DC internal tasks statistics.

LOAD/LINK -- Specify Y (for yes) if you want CA IDMS Task Analyzer to collect program load statistics. Specify N (for no) if you do not want CA IDMS Task Analyzer to collect program load statistics.

ERUS -- Specify Y (for yes) if you want CA IDMS Task Analyzer to collect external run unit statistics. Specify N (for no) if you do not want CA IDMS Task Analyzer to collect external run unit statistics.

MAXIMUM LOAD/LINK LEVELS -- Specify the program load threshold for tasks. The program load threshold is the maximum number of modules loaded by a task acceptable in your environment. If the maximum number of modules is exceeded, the task that exceeds the threshold is identified on the Program Loads Report. The default is 16.

EXCLUDE -- If you want to specifically exclude programs and/or tasks from CA IDMS Task Analyzer statistics collection (thereby reducing overhead), you may identify them here. To exclude a program, specify P and a program name. To exclude a task, specify T and a task name. To exclude groups of related tasks and/or programs, specify a T or P and a mask for the name. For example, to exclude all programs with names beginning with 'ABC', specify P and ABC##### for the name.

CA IDMS Task Analyzer will bypass statistical collection for programs having a matching value in PDEPGMID of the Program Definition Element. CA IDMS Task Analyzer will bypass statistical collection for tasks having a matching value in TDTSKCD of the Task Definition Element.

To remove programs or tasks from the exclusion list, specify D and the appropriate name.
GLOBAL STATUS -- Specify E to enable or D to disable all CA IDMS/DC user exits, including CA IDMS Task Analyzer collection exits. If you disable all exits, CA IDMS/DC statistics collection is disabled. The default is E.

COLLECTION STATUS -- Specify E to enable or D to disable the CA IDMS Task Analyzer collection exits. You may also choose the hours of operation. You may also choose the days of operation. The default is E.

O (for Option) -- Specify E to enable or D to disable specific user exits that were previously specified in the USFUEXT table. Each installed exit is identified with its specific exit number, its mode ((D) for CA IDMS/DC or (S) for IBM), its name, and its current status (active or disabled). The default is E.

LOG FILE -- Specify Y (yes) if you want CA IDMS Task Analyzer to write its statistics to the CA IDMS Log. Specify N (no) if you do not want CA IDMS Task Analyzer to write its statistics to the CA IDMS Log. The default is Y. LOG TYPE -- Specify the record type to be used by CA IDMS Task Analyzer when writing records to the CA IDMS/DC Log. Record types of 1 (messages), 2 (trace), or 4 (snap/dump) may be specified. The default is 2 (trace).

DC STATISTICS -- Specify Y (yes) if you want the CA IDMS/DC task statistics gathered by CA IDMS written to the CA IDMS/DC Log. Specify N (no) if you do not want the CA IDMS/DC task statistics gathered by CA IDMS written to the CA IDMS/DC Log. The default is Y. However, if CA IDMS/DC task statistics gathered by CA IDMS are written to the Log, they will duplicate many of the CA IDMS/DC task statistics collected by CA IDMS Task Analyzer.

z/OS -- Specify Y (yes) if you want CA IDMS Task Analyzer to write its statistics to the SMF file. Specify N (no) if you do not want CA IDMS Task Analyzer to write its statistics to the SMF file. The default is N. This option applies to z/OS only.

NUMBER -- Specify the z/OS SMF file user record number. The default is 129.

BLOCK MODE -- Specify the block mode for writing statistics to the SMF file. Specify 1 for maximum blocking efficiency. Specify 2 for blocking by transaction (that is, only those statistics for a specific transaction are included in a single SMF record).

BLOCK SIZE -- Specify the block size for the SMF file. The default is 4096. If block mode is 1, you must specify a minimum block size of (150 * the maximum number of links).

JES ID -- The JES ID that CA IDMS Task Analyzer includes on the records written to SMF is displayed here.

AUTHORIZATION MODE -- Specify the mode in which the SMF records are written. Specify 1 if the CA IDMS CV region is authorized to write to SMF. Specify 2 if the CA IDMS Task Analyzer SVC for SMF is installed.

AUTHORIZATION SVC -- If the authorization mode is 2, specify the specific SVC number for the CA IDMS Task Analyzer SMF SVC.

Enable CA-IDMS Task Analyzer User Exits

To enable the CA-IDMS Task Analyzer Exits, change the startup JCL for your CV(s) to include the required SYSIDMS parameter.

TASK_ANALYZER_EXITS = ON | OFF
ON
Enables the CA IDMS Task Analyzer User Exits. ON must be specified if CA IDMS Task Analyzer is to be used in the CV.

OFF
Disables the CA IDMS Task Analyzer User Exits.
Default: OFF

Task Analyzer z/OS Operation

Depending upon the collection options you specify on the Statistics Collection Planning screen, you may want to consider how frequently you archive the CA IDMS Log, or whether you should use multiple logs to store large quantities of statistics. To collect batch run-unit statistics, install the CA IDMS SVC exit module GSISVCX supplied with CA IDMS Task Analyzer. To use authorization mode 2 with SMF, the CA IDMS Task Analyzer SVC (USFASVC) must be installed.

- JCL to Create an Extract File from the CA IDMS Log (see page 824)
- JCL to Create an Extract File from the SMF File (see page 826)
- Report Execution JCL (see page 827)

CA IDMS Task Analyzer report programs are written in CA Culprit. To customize CA IDMS Task Analyzer reports, copy the report program modules from your dictionary, and then modify the programs to create reports that meet your current needs. The following list contains the report program module names and the reports each program generates:

- USFPBILL -- Billing Reports
- USFPPROG -- Program Reports
- USFPADSO -- CA ADS Reports
- USFPABND -- Abend Reports
- USFLOAD -- Program Loads Reports
- USFPINDEX -- Integrated Index Reports
- USFRANK -- Ranking Reports

The following JCL required to direct CA IDMS Task Analyzer was cataloged during installation:

1. To create an Extract File from the CA IDMS Log, modify and run the CA IDMS Task Analyzer Extract JCL contained in Target or Distribution source library member USFEXLOG.

2. To create an Extract File from the SMF File, modify and run the CA IDMS Task Analyzer Extract JCL contained in Target or Distribution source library member USFEXSMF.

3. To create CA IDMS Task Analyzer reports from the Extract File, modify and run the CA IDMS Task Analyzer Report Execution JCL contained in Target or Distribution source library member USFREPT.
JCL samples are shown on the following pages.

### JCL to Create an Extract File from the CA IDMS Log

Use this JCL to extract data from the CA IDMS Archive Log file and reformat it for input into the CA IDMS Task Analyzer reporting process. A key to the required variables (shown in bold) follows the JCL.

```jcl
//USFEXLOG JOB  (job card parameters)
//******************************************************************************
//* EXTRACTS DATA FROM THE CA-IDMS ARCHIVE LOG FILE AND REFORMATS IT *
//* FOR INPUT INTO THE CA-IDMS/Task Analyzer REPORTING PROCESS *
//* THE FOLLOWING VARIABLES MUST BE SUPPLIED: *
//* your.idms.loadlib - The load library where CA-IDMS was *
//* installed. *
//* printout - SYSPUT print class. *
//* errorout - ERROR print class. *
//* wrkunit - Work unit. *
//* your.idms.sysctl - SYSCTL when running in CV mode. *
//* local-dmcl - Local DMCL name. *
//* dictname - DBNAME of dictionary containing *
//* CA-IDMS/Task Analyzer Culprit source. *
//* your.sortlib - System SORT library. *
//* your.idms.plog - CA-IDMS archive PLOG. *
//* your.taska.extract - CA-IDMS/Task Analyzer extract file. *
******************************************************************************

//* OPTIONAL STEP TO DELETE CA-IDMS/Task Analyzer EXTRACT FILE.
//* IF THIS STEP IS NOT USED, AND YOU INTEND ON USING A PREVIOUSLY
//* CREATED FILE, STEP CULL4 MUST BE CHANGED TO MAKE THIS FILE
//* DISP=SHR.

/*DELETE EXEC PGM=IEFBR14

/*DDI DD DSN=your.taska.extract,DISP=(MOD,DELETE),
//UNIT=DISK,SPACE=(TRK,0)

/*CULL0 EXEC PGM=CULP0,REGION=4048K

//STEPLIB DD DISP=SHR,DSN=your.idms.loadlib//SYSOUT DD SYSOUT=printout//SYSPRINT
// DD SYSOUT=printout//SYSUDUMP DD SYSOUT=errorout//SYS004 DD SYSOUT=printout,
// DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYS005 DD DISP=(NEW,PASS),
// DSN=&UPRMWORK,UNIT=wrkunit,
// SPACE=(CYL,(10,5),RLSE),
// DCB=(RECFM,F,LRECL=320,BLKSIZE=320)
//SYSCTL DD DISP=SHR,DSN=your.idms.sysctl//SYSIDMS DD *
//DMCL=local-dmcl DBNAME=dictname/*

//SYSDN DD *
PARAM=NOLIST
COPY 'USFPEXTR' 1

/*CULL1 EXEC PGM=SORT,PARM='MSG,AP',REGION=500K

//SORTWK01 DD DISP=SHR,DSN=your.sortlib//SORTWK01 DD UNIT=wrkunit,SPACE=(CYL,(5),,
//CONTIG)
//SORTWK02 DD UNIT=wrkunit,SPACE=(CYL,(5),,CONTIG)
//SORTWK03 DD UNIT=wrkunit,SPACE=(CYL,(5),,CONTIG)
//SORTWK04 DD UNIT=wrkunit,SPACE=(CYL,(5),,CONTIG)
//SORTOUT DD DSN=&UPRMWORK,UNIT=wrkunit,
// SPACE=(CYL,(5),,CONTIG),
// DISP=(NEW,PASS),
// DCB=(RECFM=F,LRECL=320,BLKSIZE=320)
//SYSOUT DD SYSOUT=printout//SYSPRINT DD SYSOUT=printout//SYSUDUMP DD SYSOUT=error
//SORTIN DD *
//SORT Fields=(1,69,A),FORMAT=BI
//RECORD TYPE=F,LENGTH=(320,,320)
END */
```
//CULL2 EXEC PGM=CULL,REGION=4048K
//STEPLIB DD DISP=SHR,DSN=your.idms.loadlib//SYS010 DD DISP=SHR,DSN=your.idms.
plog//SYSCTL DD DISP=SHR,DSN=your.idms.sysctl//SYSIDMS DD *
IDsMBUG=OFF
/*
//SYSUDUMP DD SYSOUT=errorout//SYS004 DD SYSOUT=printout,
// DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYS005 DD DSN=&&SPRMWORK,DISP=(OLD,DELETE)
//SYS006 DD DISP=(NEW,PASS)
// DSN=&&UEXTWORK,UNIT=wrkunit,
// SPACE=(CYL,(10,5),RLSE),
// DCB=(RECFM=FB,LRECL=1000,BLKSIZE=1004)
//SYS007 DD DISP=(NEW,PASS)
// DSN=&&SRTPWORK,UNIT=wrkunit,
// SPACE=(TRK,(1,1),RLSE),
// DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYS008 DD DISP=(NEW,PASS)
// DSN=&&NSRTWORK,UNIT=wrkunit,
// SPACE=(CYL,(10,5),RLSE),
// DCB=(RECFM=FB,LRECL=1000,BLKSIZE=1004)
/*
//CULL3 EXEC PGM=SORT,PARM='MSG,AP',REGION=500K
//SORTLIB DD DISP=SHR,DSN=your.sortlib//SORTWK01 DD UNIT=wrkunit,SPACE=(CYL,(5),CONTIG)
//SORTWK02 DD UNIT=wrkunit,SPACE=(CYL,(5),CONTIG)
//SORTWK03 DD UNIT=wrkunit,SPACE=(CYL,(5),CONTIG)
//SORTWK04 DD UNIT=wrkunit,SPACE=(CYL,(5),CONTIG)
//SORTOUT DD DSN=&&SEXTWORK,UNIT=SYSDA,
// SPACE=(CYL,(5,5),CONTIG),
// DISP=(NEW,PASS)
// DCB=(RECFM=FB,LRECL=1000,BLKSIZE=1004)
//SYSOUT DD SYSOUT=printout//SYSPRINT DD SYSOUT=printout//SYSDUMP DD SYSOUT=errorout//SORTIN DD DISP=(OLD,DELETE),DSN=&&UEXTWORK
//SYSIN DD DISP=(OLD,DELETE),DSN=&&SRTPWORK
/*
//CULL4 EXEC PGM=CULE,REGION=548K
//STEPLIB DD DISP=SHR,DSN=your.idms.loadlib//SYS020 DD DSN=your.taska.extract
// UNIT=DISK
// VOL=SER=WRK06A,
// SPACE=(CYL,(10,5),RLSE),
// DISP=(NEW,CATLG,DELETE),
// DCB=(RECFM=FB,LRECL=200,BLKSIZE=6200)
//SYSCTL DD DISP=SHR,DSN=your.idms.sysctl//SYSIDMS DD *
IDsMBUG=OFF
/*
//SYSUDUMP DD SYSOUT=errorout//SYS004 DD SYSOUT=printout,
// DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYS006 DD DISP=(OLD,DELETE),DSN=&&SEXTWORK
//SYS008 DD DISP=(OLD,DELETE),DSN=&&NSRTWORK
//SYSABEND DD SYSOUT=*,OUTLIM=0
//SYSIN4 DD DUMMY

⚠️ **Note:** For more information on the CA IDMS Task Analyzer CA Culprit procedure used by USFEXLOG, see the appendix "CA Culprit Procedure JCL."

- **job card parameters** -- The job card parameters required at your installation.
- **your.idms.loadlib** -- The load library where CA IDMS was installed.
- **printout** -- SYSOUT print class.
- **errorout** -- ERROR print class.
**wrkunit** -- Work unit.

**your.idms.sysctl** -- The name of the SYSCTL when running in CV mode.

**local-dmcl** -- The local DMCL name.

**dictname** -- The DBNAME of the dictionary containing CA IDMS Task Analyzer CA Culprit source.

**your.sortlib** -- The system SORT library.

**your.idms.plog** -- CA IDMS archive PLOG.

**your.taska.extract** -- CA IDMS Task Analyzer Extract file.

---

**JCL to Create an Extract File from the SMF File**

Use this JCL to extract CA IDMS Task Analyzer record from an SMF dump file and format the data for input into the CA IDMS Task Analyzer reporting process. A key to the variables (shown in bold) and the return codes follows the JCL.

```plaintext
//USFEXSMF JOB (job card parameters)

//*******************************************************************************
// EXTRACTS CA-IDMS/Task Analyzer RECORDS FROM AN SMF DUMP FILE AND *
// FORMATS THE DATA FOR INPUT INTO THE CA-IDMS/Task Analyzer *
// REPORTING PROCESS *
// THE FOLLOWING VARIABLES MUST BE SUPPLIED: *
//*******************************************************************************
// *** smfrecid - The SMF file user record number as specified on the USFAOPT screen. ***
// *** your.loadlib - The appropriate STEPLIB DSNAMe(s). ***
// *** printout - SYSOUT print class. ***
// *** errorout - ERROR print class. ***
// *** your.taska.extract - CA-IDMS/Task Analyzer extract file. ***
// *** wrkunit - Work unit. ***
// *** blkout - CA-IDMS/Task Analyzer extract block size. ***
// *** your.smf.dump.file - The DSNAMe of the SMF dump file to be processed. ***
//*******************************************************************************
// RETURN CODES: 0 Successful creation of CA-IDMS/Task Analyzer file *
// 4 No SMF rec id parameter was supplied, Defaults to '129' *
// 8 No records were extracted from the SMF dump file *
// 16 Condition code indicates that an error occurred during this run - See report listing for error messages *
//*******************************************************************************
// OPTIONAL STEP TO DELETE CA-IDMS/Task Analyzer EXTRACT FILE. *
// IF THIS STEP IS NOT USED, AND YOU INTEND ON USING A PREVIOUSLY *
// CREATED FILE, STEP SMFEXTR MUST BE CHANGED TO MAKE THIS FILE *
// DISP=SHR. *
//*******************************************************************************
//DELETE EXEC PGM=IEFBR14
//DD1 DD DSN=your.taska.extract,DISP=(MOD,DELETE),UNIT=DISK,SPACE=(TRK,0)
//SMFEXTR EXEC PGM=USFSMFEX,REGION=256K,PARM='smfrecid'/STEPLIB DD DISP=SHR,DSN=your.loadlib/
//REPORT DD SYSOUT=printout//SYSUDUMP DD SYSOUT=errorout//OUTFILE DD DSN=your.
```
job card parameters -- The job card parameters required at your installation.

smfrecid -- The SMF file user record number as specified on the USFAOPT screen.

your.loadlib -- The appropriate STEPLIB DSNAME(s).

printout -- The SYSOUT print class.

errorout -- The ERROR print class.

your.taska.extract -- The CA IDMS Task Analyzer extract file.

wrkunit -- Work unit.

blksize -- The CA IDMS Task Analyzer extract block size.

your.smf.dump.file -- The DSNAME of the SMF dump file to be processed.

Return Codes:

- 0 -- Successful creation of an CA IDMS Task Analyzer file.
- 4 -- No SMF rec ID parameter was supplied. The parameter defaults to '129'.
- 8 -- No records were extracted from the SMF dump file.
- 16 -- Condition indicates that an error occurred during this run. See the report listing for error messages.

Report Execution JCL

Use this JCL to generate CA IDMS Task Analyzer reports. A key to the variables (shown in bold) follows the JCL.

```csh
//USFREPT JOB (job card parameters)
/***************
* GENERATES CA-IDMS/Task Analyzer REPORTS. *
* THE FOLLOWING VARIABLES MUST BE SUPPLIED: *
* your.idms.loadlib - The load library where CA-IDMS was installed. *
* your.idms.sysctl - SYSCTL when running in CV mode. *
* printout - SYSOUT print class. *
* errorout - ERROR print class. *
* wrkunit - Work unit. *
* primary - Primary SORT CYL(inder) allocation. *
* secondary - Secondary SORT CYL(inder) allocation. *
* your.taska.extract - CA-IDMS/Task Analyzer extract file. *
* your.sorted.taska.extract - Sorted CA-IDMS/Task Analyzer extract file. *
* primout - Sorted CA-IDMS/Task Analyzer extract primary CYL(inder) allocation. *
```
/* secout - Sorted CA-IDMS/Task Analyzer extract */
/* blkout - Sorted CA-IDMS/Task Analyzer extract */
/* local-dmcl - Local DMCL name. */
/* dictionary - DBNAME of dictionary containing */
/* your.sortlib - System SORT library. */
*******************************************************************************/
/* OPTIONAL STEP TO DELETE CA-IDMS/Task Analyzer EXTRACT FILE. */
/* IF THIS STEP IS NOT USED, AND YOU INTEND ON USING A PREVIOUSLY */
/* CREATED FILE, STEP SORT MUST BE CHANGED TO MAKE THIS FILE */
/* DISP=SHR. */
/*DELETE EXEC PGM=IEFBR14
 *DD I DD DSN=your.sorted.taska.extract,DISP=(MOD,DELETE),
 *       UNIT=DISK,SPACE=(TRK,0)
 */
/*SORT EXEC PGM=SORT,PARM='RC16=ABE'
 *STEPLIB DD DISP=SHR,DSN=
 *SYSPRINT DD SYSOUT=printout/SYSOUT=printout/SYSUDUMP DD SYSOUT=error
 *SORTWK01 DD UNIT=wrkunit,SPACE=(CYL,(primary,secondary))
 *SORTWK02 DD UNIT=wrkunit,SPACE=(CYL,(primary,secondary))
 *SORTWK04 DD UNIT=wrkunit,SPACE=(CYL,(primary,secondary))
 *SORTWK05 DD UNIT=wrkunit,SPACE=(CYL,(primary,secondary))
 *SORTWK06 DD UNIT=wrkunit,SPACE=(CYL,(primary,secondary))
 *SORTIN DD DISP=SHR,DSN=your.taska.extract/SORTOUT DD DSN=your.sorted.taska.
 extract,
 / 
 / 
 /SYSIN DD *
 *SORT FIELDS=(7,14,A),FORMA=BI,EQUALS
 */
/*CULL0 EXEC PGM=CULP0,REGION=4048K
 *STEPLIB DD DISP=SHR,DSN=
 *SYSCTL DD DISP=SHR,DSN=
 *SYS004 DD SYSOUT=printout/SYSOUT=printout,
 / 
 /DCB=(RECFM=FBA,LRECL=133,BKSIZE=133)
 *SYSUDUMP DD SYSOUT=errorout/SYS005 DD DISP=(NEW,PASS),
 / 
 /DSN=&&UPRMWORK,UNIT=wrkunit,
 / 
 /DCB=(RECFM=F,LRECL=320,BKSIZE=320)
 *SYSIDMS DD *
 *DMCL=local-dmcl DBNAME=dictionary/*
 */
/*USFPMAIN and USFPWORK are always needed */
/* Others needed based on requested reports */
/*SY SIN DD *
 *PARAM=NOLIST
 =COPY 'USFPMAIN'
 =COPY 'USFPWORK'
 =COPY 'USFPBILL'
 =COPY 'USFPABND'
 =COPY 'USFPADSO'
 =COPY 'USFPINDX'
 =COPY 'USFPLOAD'
 =COPY 'USFPFPROG'
 =COPY 'USFPFRAWK'
/* */
/*CULL1 EXEC PGM=SORT,PARM='MSG,AP',REGION=500K
 *STEPLIB DD DISP=SHR,DSN=
 *SORTWK01 DD UNIT=wrkunit,SPACE=(CYL,(5),CONTIG)
 *SORTWK02 DD UNIT=wrkunit,SPACE=(CYL,(5),CONTIG)
 *SORTWK03 DD UNIT=wrkunit,SPACE=(CYL,(5),CONTIG)
 *SORTWK04 DD UNIT=wrkunit,SPACE=(CYL,(5),CONTIG)
//SORTOUT DD DSN=&SPRMWORK,UNIT=wrkunit,  
//SPACE=(CYL,(5,5),,CONTIG),  
//DISP=(NEW,PASS),  
//DCB=(RECFM=F,LRECL=320,BLKSIZE=320)  
//SYSOUT DD SYSOUT=printout/SYSPRINTF DD SYSOUT=printout/SYSUDUMP DD SYSOUT=errout/SORTIN DD DISP=(OLD,DELETE),DSN=&UPRMWORK  
//SYSIN DD *  
SORT FIELDS=(1,69,A),FORMAT=BF  
RECORD TYPE=F,LENGTH=(320,,320)  
END  
/*  
//CULL2 EXEC PGM=CULL,REGION=4048K  
//STEPLIB DD DISP=SHR,DSN=your.idms.loadlib//SYS011 DD DISP=SHR,DSN=your.sorted. taska.extract//SYSCTL DD DISP=SHR,DSN=your.idms.sysctl//SYSUDUMP DD SYSOUT=errorout/SYSIDMS DD *  
IDMSDBUG=OFF  
*/  
//SYS004 DD SYSOUT=printout,  
//DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)  
//SYS005 DD DSN=&SPRMWORK,DISP=(OLD,DELETE)  
//SYS006 DD DISP=(NEW,PASS),  
//DSN=&UEXWTWORK,UNIT=wrkunit,  
//SPACE=(CYL,(10,5),RLSE),  
//DCB=(RECFM=V8,LRECL=1000,BLKSIZE=1004)  
//SYS007 DD DISP=(NEW,PASS),  
//DSN=&SRTPWORK,UNIT=wrkunit,  
//SPACE=(TRK,(1,1),RLSE),  
//DCB=(RECFM=F,LRECL=80,BLKSIZE=80)  
//SYS008 DD DISP=(NEW,PASS),  
//DSN=&NSRTWORK,UNIT=wrkunit,  
//SPACE=(CYL,(10,5),RLSE),  
//DCB=(RECFM=V8,LRECL=1000,BLKSIZE=1004)  
/*  
* PROCESS and REPORT request statements  
*  
//SYS010 DD *  
PROCESS DCSYSRUS=N IDMSXXXX=N START=0101870000 STOP=1231002359  
*PROCESS CNVNUM=@ALL PLANID=@ALL  
REPORT=BILL LEVEL=DET RUTYPE=@ RUNAME=@ALL NAME=*  
REPORT=ABND LEVEL=DET RUTYPE=@ RUNAME=@ALL NAME=*  
REPORT=ADSO LEVEL=DET RUTYPE=@ RUNAME=@ALL NAME=*  
REPORT=INDX LEVEL=DET RUTYPE=@ RUNAME=@ALL NAME=*  
REPORT=LOAD LEVEL=DET RUTYPE=@ RUNAME=@ALL NAME=*  
REPORT=PROG LEVEL=DET RUTYPE=@ RUNAME=@ALL NAME=*  
REPORT=RANK ORDER=D NUMBER=035 HOW=GT WHAT=TERMREAD VALUE=000000001000  
*/  
/*  
//CULL3 EXEC PGM=SORT,PARM='MSG,AP',REGION=508K  
//STEPLIB DD DISP=SHR,DSN=your.idms.loadlib//SORTLIB DD DISP=SHR,DSN=your.sortlib/ 
//SORTWK01 DD UNIT=wrkunit,SPACE=(CYL,(5).,CONTIG)  
//SORTWK02 DD UNIT=wrkunit,SPACE=(CYL,(5).,CONTIG)  
//SORTWK03 DD UNIT=wrkunit,SPACE=(CYL,(5).,CONTIG)  
//SORTWK04 DD UNIT=wrkunit,SPACE=(CYL,(5).,CONTIG)  
//SORTOUT DD DSN=&SEXITWORK,UNIT=wrkunit,  
//SPACE=(CYL,(5),,CONTIG),  
//DISP=(NEW,PASS),  
//DCB=(RECFM=V8,LRECL=1000,BLKSIZE=1004)  
//SYSOUT DD SYSOUT=printout/SYSPRINTF DD SYSOUT=printout/SYSUDUMP DD SYSOUT=errout/SORTIN DD DISP=(OLD,DELETE),DSN=&SEXITWORK  
//SYSIN DD DISP=(OLD,DELETE),DSN=&SRTWPWORK  
*/  
/*  
//CULL4 EXEC PGM=CULE,REGION=548K  
//STEPLIB DD DISP=SHR,DSN=your.idms.loadlib//SYSCTL DD DISP=SHR,DSN=your.idms.sysctl//SYSUDUMP DD SYSOUT=errorout/SYSIDMS DD *  
IDMSDBUG=OFF  
*/  
/*  
//SYS004 DD SYSOUT=printout,  
//DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)  
//SYS006 DD DISP=(OLD,DELETE),DSN=&SEXITWORK  
//SYS008 DD DISP=(OLD,DELETE),DSN=&SRTWPWORK  
*/
Note: For more information on the CA IDMS Task Analyzer CA Culprit procedure used by USFREPT, see the appendix "CA Culprit Procedure JCL."

- **job card parameters** -- The job card parameters required at your installation.
- **your.idms.loadlib** -- The load library where CA IDMS was installed.
- **your.idms.sysctl** -- The SYSCTL when running in CV mode.
- **printout** -- The SYSOUT print class.
- **errorout** -- The ERROR print class.
- **wrkunit** -- Work unit.
- **primary** -- The primary SORT CYL(inder) allocation.
- **secondary** -- The secondary SORT CYL(inder) allocation.
- **your.taska.extract** -- The CA IDMS Task Analyzer Extract file.
- **your.sorted.taska.extract** -- The sorted CA IDMS Task Analyzer Extract file.
- **primout** -- The sorted CA IDMS Task Analyzer Extract primary CYL(inder) allocation.
- **secout** -- The sorted CA IDMS Task Analyzer Extract secondary CYL(inder) allocation.
- **blkout** -- Sorted CA IDMS Task Analyzer Extract block size.
- **local-dmcl** -- The local DMCL name.
- **dictname** -- The DBNAME of the dictionary containing the CA IDMS Task Analyzer CA Culprit source.
CA IDMS - 19.0

- your.sortlib -- The system SORT library.

CA IDMS Task Analyzer z/VSE Operations

Depending upon the collection options you specify on the Statistics Plan screen, you may want to consider how frequently you archive the CA IDMS Log, or whether you should use multiple logs to store large quantities of statistics. To collect batch run-unit statistics, install the CA IDMS SVC exit module GSISVCX supplied with CA IDMS Task Analyzer.

- z/VSE File Assignments (see page 831)

CA IDMS Task Analyzer report programs are written in CA Culprit. To customize CA IDMS Task Analyzer reports, copy the report program modules from your dictionary, and then modify the programs to create reports that meet your current needs. The following list contains the report program module names and the reports each program generates:

- USFPBILL -- Billing Reports
- USFPPROG -- Program Reports
- USFPADSO -- CA ADS Reports
- USFPABND -- Abend Reports
- USFPLOAD -- Program Loads Reports
- USFPINDEX -- Integrated Index Reports
- USFPANK -- Ranking Reports

The JCL required to direct CA IDMS Task Analyzer was cataloged during installation. The use of the JCL and the names of the TOOLJCL library members containing the JCL are as follows:

1. To create an Extract File from the CA IDMS Log, modify and run the CA IDMS Task Analyzer Extract JCL contained in TOOLJCL library member USFEXTRC.S (z/VSE).
2. To create CA IDMS Task Analyzer reports from the Extract File, modify and run the CA IDMS Task Analyzer Report Execution JCL contained in TOOLJCL library member USFEXEC.S (z/VSE).

Both sets of JCL are shown on the following pages.

z/VSE File Assignments

Even if you use a storage management tool such as CA DYNAM, CA IDMS Task Analyzer requires an ASSGN statement for every file except SORTWKnn. This ASSGN is required because CA IDMS Task Analyzer has its own device-independent support which dynamically builds a DTF based on the device type indicated by the ASSGN. Unless the ASSGN specifies VSAM or BDAM, the file may be defined with either DLBL or TLBL.

```plaintext
// (job card parameters)
* FOR z/VSE USE THE FOLLOWING STATEMENTS
// DLBL  lib-filename,'your.idms.library'
// EXTENT ,volser// LIBDEF SOURCE,SEARCH=lib-filename.sublib// LIBDEF PHASE,SEARCH=lib
```

16-Jan-2018 831/898
* FOR /VSE USE THE FOLLOWING STATEMENTS
// DLBL  IJSSYSL,'idms.corelib'
// EXTENT ,volser// DLBL IJSSYSL,'yourHLQ.CAGJSRC'
// EXTENT ,volser// LIBDEF SL,SEARCH=IJSSYSL
// LIBDEF CL, SEARCH=IJSSYSL
* ************************************************************
// UPSI 1
// OPTION LOG,PARTDUMP
* ******************************************************************
* If running in LOCAL mode, include dataset containing
* the DDLML area of the dictionary containing Culprit modules
* **************************************************************
// DLBL DICTDB,'your.dict.ddldml'
// EXTENT SYSnnn,volser,,rel-trk-blk,amount// ASSGN SYSnnn,DISK,VOL=volser,SHR
* ******************************************************************
* *************
* FOR CV runs specify:
// DLBL SYSCTL,'your.sysctl.file'
// EXTENT SYSnnn,volser,,starttrack,#tracks// ASSGN SYSnnn,DISK,VOL=volser,SHR
* ******************************************************************
// ASSGN SYS004,SYSLST
// DLBL SYS005,'CULPRIT.PARMS',0
// EXTENT SYS005,volser,,rel-trk-blk,amount// ASSGN SYS005,DISK,VOL=volser,SHR
// DLBL SYS006,'CULPRIT.EXTRACT',0
// EXTENT SYS006,volser,,rel-trk-blk,amount// ASSGN SYS006,DISK,VOL=volser,SHR
// DLBL SYS007,'CULPRIT.SORTCARD',0
// EXTENT SYS007,volser,,rel-trk-blk,amount// ASSGN SYS007,DISK,VOL=volser,SHR
// DLBL SYS008,'CULPRIT.NOSORT',0
// EXTENT SYS008,volser,,rel-trk-blk,amount// ASSGN SYS008,DISK,VOL=volser,SHR
// DLBL SORTWK1,'CULSORT.WORK',0
// EXTENT SYS001,volser,,rel-trk-blk,amount// ASSGN SYS001,DISK,VOL=volser,SHR
// DLBL SYS010,'your.archive.log'
// EXTENT SYS010,volser,,rel-trk-blk,amount// ASSGN SYS010,DISK,VOL=volser,SHR
// DLBL SYS020,'your.extract.file',0
// EXTENT SYS020,volser,,rel-trk-blk,amount// ASSGN SYS020,DISK,VOL=volser,SHR
// DLBL SYSDMDS,'#SYSIPT',0,5D
// EXEC CULPRIT,SIZE=1024K
// PARAM=NOLIST
=COPY 'USFPEXTR'
/*
 Fill in appropriate Rnn.n SYSIDMS parameters
 * ECHO=ON OR OFF
 LOCAL=OFF OR ON
 DMCL=your-dmcl-name DICTNAME=your-dictionary-name*/
&

Use the CA IDMS Task Analyzer EXTRACT JCL to:

1. Select data from the CA IDMS Log.
2. Reformat the data for use by CA IDMS Task Analyzer report programs.
3. Copy the data to the Extract File.

4. **job card parameters** -- The job card parameters required at your company.

5. **lib-filename** -- The file name of the /VSE library where your CA IDMS executable phases and source reside.

6. **your.idms.library** -- The data set name of the library where your CA IDMS executable phases and source reside.
7. **volser** -- The volume serial number or generic assignment of the disk volume on which the library or file, specified in the preceding DLBL statement, resides.

8. **sublib** -- The name of the sublibrary of the z/VSE library specified in the preceding DLBL statement.

9. **idms.corelib** -- The name of your CA IDMS core image library.

10. **yourHLQ.CAGJSRC** -- The name of your CA IDMS source statement library.

11. **rel-trk-blk** -- The starting position on the DASD for storage of the work file. z/VSE Users: do not start track assignment at 000000.

12. **amount** -- The number of tracks or blocks you need for storage of the work file.

13. **your.archive.log** -- The name of the CA IDMS Log from which the records are to be extracted.

14. **your.extract.file** -- The name of the extract file to which the CA IDMS Task Analyzer records from the CA IDMS Log are to be copied.

```
// JOB USFEXEC (job card parameters)
* **** CREATE A SYSIDMS PARAMETER FILE (nn.n) *****
// UPSI 1
// OPTION LOG,PARTDUMP
// DLBL anyname,'work.file.SYSIDMS',0,SD
// EXTENT SYS060,volser,,rel-trk-blk,amount// ASSGN SYS060,DISK,VOL=volser,SHR
// EXEC DITTO
$$DITTO CSQ FILEOUT=anyname* Rnn.n SYSIDMS parameters.
* For Local Mode specify :
* DMCL=dmcl-name,LOCAL=ON,JOURNAL=OFF,DBNAME=your.dbname*
* For CV runs specify :
* DMCL=dmcl-name,LOCAL=OFF,JOURNAL=OFF,DBNAME=your.dbname*/
$$DITTO EOJ
/
// UPSI 0
* FOR z/VSE USE THE FOLLOWING STATEMENTS
// DLBL lib-filename,'your.idms.library'
// EXTENT ,volser// LIBDEF SOURCE,SEARCH=lib-filename.sublib// LIBDEF PHASE,SEARCH=lib-filename.sublib* *
* FOR z/VSE USE THE FOLLOWING STATEMENTS
// DLBL IJSYSCL,'idms.corelib'
// EXTENT ,volser// DLBL IJSYSSL,'yourHLQ.CAGJSRC'
// EXTENT ,volser// LIBDEF SL,SEARCH=IJSYSSL
// LIBDEF CL,SEARCH=IJSYSCL
* ***************************************************
* OPTION LOG,PARTDUMP
// ASSGN SYS004,SYSLST
// ASSGN SYS030,SYSLST
// ASSGN SYS031,SYSLST
// ASSGN SYS032,SYSLST
// ASSGN SYS033,SYSLST
// ASSGN SYS034,SYSLST
// ASSGN SYS035,SYSLST
// ASSGN SYS036,SYSLST
// ASSGN SYS037,SYSLST
// ASSGN SYS038,SYSLST
// ASSGN SYS039,SYSLST
// ASSGN SYS040,SYSLST
// ASSGN SYS041,SYSLST
// ASSGN SYS042,SYSLST
```
Use the CA IDMS Task Analyzer REPORT EXECUTION JCL to:

1. Identify and run CA IDMS Task Analyzer reports. For more information on the use of parameters, see the beginning of this section.

2. Use the CA IDMS Task Analyzer report programs.

3. **job card parameters** -- The job card parameters required at your company.

4. **anynname** -- Any suitable name for your SYSIDMS file. Note the name chosen must be identical to the FILEOUT value in the DITTO copy step.

5. **volser** -- The volume serial number or generic assignment of the disk volume on which the library or file, specified in the preceding DLBL statement, resides.

6. **rel-trk-blk** -- The starting position on the DASD for storage of the work file. z/VSE Users: do not start track assignment at 000000.
7. **amount** -- The number of tracks or blocks you need for storage of the work file.

8. **lib-filename** -- The file name of the z/VSE library where your CA IDMS executable phases and source reside.

9. **your.idms.library** -- The data set name of the library where your CA IDMS executable phases and source reside.

10. **sublib** -- The name of the sublibrary of the z/VSE library specified in the preceding DLBL statement.

11. **idms.corelib** -- The name of your CA IDMS core image library.

12. **yourHLQ.CAGJSRC** -- The name of your CA IDMS source statement library.

13. **your.dict.ddldml** -- The file ID of the dictionary into which the CULPRIT report statements have been loaded.

14. **your.sysctl.file** -- The file ID of your SYSCTL file.

15. **your.extract.file** -- The name of the extract file to which the CA IDMS Task Analyzer records from the CA IDMS Log are to be copied.

16. **work.file.SYSIDMS** -- The file ID of your SYSIDMS work file.

17. **xxxx** -- The suffix of the report program name: BILL, PROG, ADSO, ABND, LOAD, INDX, or RANK.

⚠️ **Note:** The Report Execution JCL, contained in USFEXEC, contains PROCESS and REPORT statement examples. For more information on the use of parameters, see the section "Parameters."

---

**CA IDMS Task Analyzer z/VM Operations**

Before operating CA IDMS Task Analyzer in a z/VM environment, review the following considerations. Depending upon the collection options you specify on the Statistics Plan Screen, you may want to consider how frequently you archive the CA IDMS Log, or whether you should use multiple logs to store large quantities of statistics. To collect batch run-unit statistics, install the CA IDMS SVC exit module GSISVCX supplied with CA IDMS Task Analyzer.

- The Extract EXEC USFXTRCT (see page 836)
- The Report Execution EXEC USFRPRT (see page 838)

CA IDMS Task Analyzer report programs are written in CA Culprit. To customize CA IDMS Task Analyzer reports, copy the report program modules from your dictionary, and then modify the programs to create reports that meet your current needs. The following list contains the report program module names and the reports each program generates:
USFPBILL -- Billing Reports
USFPROG -- Program Reports
USFPADSO -- CA ADS Reports
USFPABND -- Abend Reports
USFPLOAD -- Program Loads Reports
USFPINDEX -- Integrated Index Reports
USFPRANK -- Ranking Reports

USFXTRCT and USFRPRT, the EXECs required to direct CA IDMS Task Analyzer, were cataloged during installation and are shown on the following pages.

The Extract EXEC USFXTRCT

USFXTRCT, the Extract EXEC, is shown on the next two pages. To create an Extract File from the CA IDMS Log:

1. Create a SYSIN file with the following CA Culprit parameters:

   DATABASE DICTNAME=your.dict.name
   PARAM=NOLIST
   =COPY 'USFPEXTR'

   where:
   your.dict.name is the name of the dictionary in which the Extract programs reside.
   If you do not use a secondary dictionary, do not create a DATABASE parameter.

2. Modify and run the CA IDMS Task Analyzer Extract EXEC contained in source library member USFXTRCT.

   /* */
   TRACE OFF; SIGNAL ON ERROR
   /* */
   IDMS_LOADLIB_FN = 'idms.loadlib'
   SORT_TXTLIB_FN = 'your.sortlib'
   /* */
   /* Link and access the Minidisks containing the required library(s) */
   /* */
   'CP SPOOL PRINTER NOCONT CLOSE'
   'GLOBAL LOADLIB ' IDMS_LOADLIB_FN
   'GLOBAL TXTLIB ' SORT_TXTLIB_FN
   /* */
   /* Restart Dataset */
   /* */
   /* 'FILEDEF SYS004 DUMMY' */
   /* */
   /* Input Program Listing */
   /* */
   'FILEDEF SYS004 DISK FILE SYS004 fm4 (RECFM FBA LRECL 133 BLKSIZE 133'
Use the CA IDMS Task Analyzer EXTRACT EXEC to:
1. Select data from the CA IDMS Log.

2. Reformat the data for use by CA IDMS Task Analyzer report programs.

3. Copy the data to the Extract File.

4. `idms.loadlib` -- The file name of the load library containing your CA IDMS modules.

5. `your.sortlib` -- The file name of the library containing your sort module.

6. `dcb` -- DCB information for the archive log tape: record format, logical record length, and block size.

7. `fm4` -- The file mode of the relevant file. File mode 4 indicates z/OS file simulation.

The Report Execution EXEC USFRPRT

USFRPRT, the Report Execution EXEC, is shown on the next two pages. To create CA IDMS Task Analyzer reports from the Extract File:

1. Create a SYSIN file with the following CA Culprit parameters:

   ```
   DATABASE DICTNAME=your.dict.name PARM=NOLIST
   =COPY 'USFPMAIN'
   =COPY 'USFPWORK'
   =COPY 'USFPxxxx'
   /*
   TRACE OFF; SIGNAL ON ERROR
   /* USFRPRT
   /*
   IDMS_LOADLIB_FN = 'idms.loadlib'
   SORT_TXTLIB_FN = 'your.sortlib'
   /*
   /* Link and access the Minidisks containing the required librarie(s) */
   /*
   'CP SPOOL PRINTER NOCONT CLOSE'
   'GLOBAL LOADLIB ' IDMS_LOADLIB_FN
   'GLOBAL TXTLIB ' SORT_TXTLIB_FN
   /*
   'FILEDEF SYSLST DISK FILE SYSLST fm'
   'FILEDEF SYSIN4 DISK FILE SYSIN4 fm'
   'FILEDEF SYSPRINT DISK FILE SYSPRINT fm'
   'FILEDEF SYSDUMP DISK FILE SYSDUMP fm'
   ```

   where:
   `your.dict.name` is the name of the dictionary into which the CA IDMS Task Analyzer Report programs were loaded.
   
   If you do not use a secondary dictionary, do not create the DATABASE parameter.
   `xxxx` is the suffix of the report program name: BILL, PROG, ADSO, ABND, LOAD, INDX, or RANK.

2. Create a SYS010 file for the CA IDMS Task Analyzer parameters. Use as parameters the PROCESS and REPORT statements, supplying values in the correct positions. For more information on the use of parameters, see the section "Parameters."

3. Modify and run the CA IDMS Task Analyzer Report Execution EXEC contained in source library member USFRPRT.
'FILEDEF SORTPRNT DISK FILE SORTPRNT fm'
'FILEDEF SORTMSGS DISK FILE SORTMSGS fm'
'FILEDEF SYSOUT DISK FILE SYSOUT fm'
/* Input Program Listing */
/* Work File */
/* Work File */
/* Idms10 Sort Parameters */
/* Extract Work File */
/* Work File */
/* Work File */
/* Work File */
/* CA-IDMS/Task Analyzer Parameter File */
/* Extract Work File */
/* CA-IDMS/Task Analyzer Extract File (Created in Extract EXEC) */
/* CULPRIT SYSIN File */
/* Report Files */
/* You must create a file 'SYSIDMS INPUT A' containing the SYSIDMS */
/* parameters you use to specify your runtime environment. */
/* 'FILEDEF SYSIDMS DISK SYSIDMS INPUT A' */
SIGNAL OFF ERROR
SAY 'STARTING CA-IDMS/Task Analyzer CULPRIT REPORT PROCESSING'
'EXECOS OSRUN CULPRIT'
USFRPRT_RC = RC
Use the CA IDMS Task Analyzer REPORT EXECUTION EXEC to:

1. Identify and run CA IDMS Task Analyzer reports.
2. Use the CA IDMS Task Analyzer report programs.
3. idms.loadlib -- The name of your CA IDMS load library.
4. your.sortlib -- The name of your sort text library.
5. fn ft fm -- The file name, file type, and file mode of the relevant file.
6. fn ft fm4 -- The file name, file type, and file mode of the relevant file. File mode 4 indicates z/OS file simulation.

Note: For more information on the use of parameters, see the section "Parameters."

External Request Element Extension 1

This section provides a description of the External Request Element (ERE) extension. Altering the ERE description is necessary if you want to tailor the CA IDMS Log Analyzer Billing Reports. To change the ERE extension you must alter GSISSVCX, USLBILX, and USLRPT5.

```
*--------------------------------------------------------------*
* ERE LAYOUT (AS CREATED BY GSISVCX)                         *
*                                                          *
* NOTE: THESE FIELDS ARE CONTAINED WITHIN                    *
* THE LOG STATISTICS                                         *
* TYPE '28' RECORD (TST - TASK - STATISTICS)                 *
*--------------------------------------------------------------*
10  TST-STATS-BLOCK-ID
15  TST-ERE
```
CA Culprit Procedure JCL

The JCL for the CA Culprit procedure used in CA IDMS Task Analyzer is shown here. The CA Culprit procedure, found in source library member USFPCULP, is used to generate reports.

When the CA IDMS Task Analyzer Report Execution JCL is run, it executes the CA Culprit procedure contained in the source member USFPCULP. The JCL for the CA Culprit procedure is shown in this appendix.

```jcl
//USFPCULP PROC PRINT=A, ERML=D, UNIT=3380, SYSTCL='YOUR.IDMS.SYSCTL', IDMSLD2='YOUR.IDMS.LOADLIB', IDMSRC='YOUR.IDMS.SRCLIB', SORTLIB='YOUR.SYS1.SRTLIB'
//*****************************************************************************/
//* CA-Culprit PROCEDURE
//*****************************************************************************/
//CULPO EXEC PGM=CULPO,REGION=320K, DSN=IDSMXLDB,DISP=SHR
//SYS004 DD SYSOUT=6,PRINT,DCB=(RECFM=FBA,LRECL=133,BLKSZ=133)
//SYS005 DD DSN=UPRMWORK,UNIT=6,DISP=(NEW,PASS), SPACE=(CYL,(10,5),RLSE,
//DCB=(RECFM=F,LRECL=320,BLKSZ=320)
//SYSCTL DD DSN=SYSCTL,DISP=SHR
//*
//CULP1 EXEC PGM=SORT,REGION=320K,PARM='MSG=AP'
//SORTLIB DD DSN=SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=6,SPACE=(CYL,(5),CONTIG)
//SORTWK02 DD UNIT=6,SPACE=(CYL,(5),CONTIG)
//SORTWK03 DD UNIT=6,SPACE=(CYL,(5),CONTIG)
//SORTWK04 DD UNIT=6,SPACE=(CYL,(5),CONTIG)
```
//SORTOUT DD DSN=66SPRMW0RK,UNIT=6UNIT,Vol=6VOLSER, DISP=(NEW,PASS),
  SPACE=(CYL,(5),CONTIG),
  DCB=(RECFM=F,LRECL=320,BLKSIZE=320)
//SYSOUT DD SYSOUT=6PRINT
//SYSPRINT DD SYSOUT=6PRINT
//SORTIN DD DSN=66UPRMW0RK,DISP=(OLD,DELETE)
//SYSIN DD DSN=6IDMSRC.(SORT1),DISP=SHR
//*
//CULP2 EXEC PGM=CUL,REGION=320K,
//STEPLIB DD DSN=6IDSM0LD,DISP=SHR
//SYS004 DD SYSDT=6&ERML,DCB=(RECFM=FBA,LRECL=133,
  BLKSIZE=133)
//SYS005 DD DSN=66SPRMW0RK,DISP=(OLD,DELETE)
//SYS006 DD DSN=66UEXTW0RK,UNIT=6UNIT,
  DISP=(NEW,PASS),
  SPACE=(CYL,(10,5),RLSE),
  DCB=(RECFM=VB,LRECL=1000,BLKSIZE=1004)
//SYS008 DD DSN=66NSRTW0RK,UNIT=6UNIT,
  DISP=(NEW,PASS),
  SPACE=(CYL,(10,5),RLSE),
  DCB=(RECFM=VB,LRECL=1000,BLKSIZE=1004)
//SYS007 DD DSN=66SRTPW0RK,UNIT=6UNIT,
  DISP=(NEW,PASS),
  SPACE=(CYL,(10,5),RLSE),
  DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYS009 DD DUMMY
//SYSJRNL DD DUMMY
//*
//CULP3 EXEC PGM=SORT,REGION=320K,PARM='MSG=AP'
//SORTLIB DD DSN=6SORTLIB,DISP=SHR
//SORTWK01 DD UNIT=6UNIT,
  SPACE=(CYL,(5),CONTIG)
//SORTWK02 DD UNIT=6UNIT,
  SPACE=(CYL,(5),CONTIG)
//SORTWK03 DD UNIT=6UNIT,
  SPACE=(CYL,(5),CONTIG)
//SORTWK04 DD UNIT=6UNIT,
  SPACE=(CYL,(5),CONTIG)
//SORTOUT DD DSN=66UEXTW0RK,UNIT=6UNIT,Vol=6VOLSER,
  DISP=(NEW,PASS),
  SPACE=(CYL,(15,5),RLSE),
  DCB=(RECFM=VB,LRECL=1000,BLKSIZE=1004)
//SYSOUT DD SYSOUT=6PRINT
//SYSPRINT DD SYSOUT=6PRINT
//SORTIN DD DSN=66UEXTW0RK,DISP=(OLD,DELETE)
//SYSIN DD DSN=66SRTPW0RK,DISP=(OLD,DELETE)
//*
//CULP4 EXEC PGM=CULE,REGION=512K
//STEPLIB DD DSN=6IDSM0LD,DISP=SHR
//SYS004 DD SYSDT=6PRINT,DCB=(RECFM=FBA,LRECL=133,
  BLKSIZE=133)
//SYS006 DD DSN=66UEXTW0RK,DISP=(OLD,DELETE)
//SYS008 DD DSN=66NSRTW0RK,DISP=(OLD,DELETE)
//SYS030 DD SYSDT=6PRINT,DCB=(RECFM=FBA,LRECL=133,
  BLKSIZE=133)
//SYS031 DD SYSDT=6PRINT,DCB=(RECFM=FBA,LRECL=133,
  BLKSIZE=133)
//SYS032 DD SYSDT=6PRINT,DCB=(RECFM=FBA,LRECL=133,
  BLKSIZE=133)
//SYS033 DD SYSDT=6PRINT,DCB=(RECFM=FBA,LRECL=133,
  BLKSIZE=133)
//SYS034 DD SYSDT=6PRINT,DCB=(RECFM=FBA,LRECL=133,
  BLKSIZE=133)
//SYS035 DD SYSDT=6PRINT,DCB=(RECFM=FBA,LRECL=133,
  BLKSIZE=133)
//SYS036 DD SYSDT=6PRINT,DCB=(RECFM=FBA,LRECL=133,
  BLKSIZE=133)
//SYS037 DD SYSDT=6PRINT,DCB=(RECFM=FBA,LRECL=133,
  BLKSIZE=133)
//SYS038 DD SYSDT=6PRINT,DCB=(RECFM=FBA,LRECL=133,
// SYS039 DD SYSOUT=&PRINT,DCB=(RECFM=FBA,LRECL=133, BLKSIZE=133)
// SYS040 DD SYSOUT=&PRINT,DCB=(RECFM=FBA,LRECL=133, BLKSIZE=133)
// SYS041 DD SYSOUT=&PRINT,DCB=(RECFM=FBA,LRECL=133, BLKSIZE=133)
// SYS042 DD SYSOUT=&PRINT,DCB=(RECFM=FBA,LRECL=133, BLKSIZE=133)
// SYS043 DD SYSOUT=&PRINT,DCB=(RECFM=FBA,LRECL=133, BLKSIZE=133)
// SYS044 DD SYSOUT=&PRINT,DCB=(RECFM=FBA,LRECL=133, BLKSIZE=133)
// SYS045 DD SYSOUT=&PRINT,DCB=(RECFM=FBA,LRECL=133, BLKSIZE=133)
// SYS046 DD SYSOUT=&PRINT,DCB=(RECFM=FBA,LRECL=133, BLKSIZE=133)
// SYSABEND DD DUMMY,OUTLIM=0
// SYSIN DD DUMMY,DCB=BLKSIZE=80
// PEND **** END OF PROCEDURE ****
Commands to execute CA IDMS reports in local mode are shown as follows. To execute the reports in local mode, take one of the following actions:

- Code LOCAL as an option on the DATABASE parameter. For more information on this option, see the CA Culprit for CA IDMS Reference section.
- Code PARM=\"LOCAL\" on the OSRUN command used to invoke the program. This option is valid only if the OSRUN command is issued from the System Product Interpreter or an EXEC2 file.
- Link edit the program with an IDMSOPTI module that specifies CENTRAL=NO.

CULPRIT (z/VM)

*------------------ CA IDMSRPTS *
GLOBAL TXTLIB sortlib
FILEDEF SYS004 PRINTER (RECFM FBA LRECL 133 BLKSIZE 133
FILEDEF SYS005 DISK uprwork file a (RECFM FB LRECL 320 BLKSIZE 1600
FILEDEF CULSRTII DISK IDMSLIB MACLIB A (MEMBER SORT1
FILEDEF SYS006 DISK uextwork file a (RECFM VB LRECL 1024 BLKSIZE 4096
FILEDEF SYS007 DISK srtpwork file a (RECFM F LRECL 80 BLKSIZE 80
FILEDEF SYS008 DISK nsrtwork file a (RECFM VB LRECL 1024 BLKSIZE 4096
FILEDEF SYS010 DISK input file a
FILEDEF SYS020 DISK nonprint file a
FILEDEF SYS020 DISK card output a
FILEDEF SYS020 DISK card output a
FILEDEF SYS04 DISK sysipt data a
FILEDEF CULLIB DISK srclib MACLIB a
FILEDEF sysjnl DUMMY
FILEDEF SYSIDMS DISK sysidms parms a (RECFM F LRECL pppp BLKSIZE pppp
EXEC IDMSFD
OSRUN CULPRIT

<table>
<thead>
<tr>
<th>card</th>
<th>filename, filetype, and filemode of the file that contains the card output (required for DREPORT 051 and CREPORT 051)</th>
</tr>
</thead>
<tbody>
<tr>
<td>output</td>
<td>DREPORT 051 and CREPORT 051</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDMSFD Name of the exec provided at installation that contains the file definitions for CA IDMS dictionaries, sample databases, and disk journal files, TXTLIBs, and LOADLIBs</th>
</tr>
</thead>
<tbody>
<tr>
<td>input</td>
</tr>
<tr>
<td>nnnn</td>
</tr>
<tr>
<td>nonprint</td>
</tr>
<tr>
<td>nsrtwork</td>
</tr>
<tr>
<td>pppp</td>
</tr>
<tr>
<td>restart</td>
</tr>
<tr>
<td>sortlib</td>
</tr>
<tr>
<td>srclib</td>
</tr>
</tbody>
</table>
file name of PDS containing parameters to be copied (necessary only if USE =COPY, or =MACRO is used)

srtpwor  filename, filetype, and filemode of the sort control parameter file

sysipt   filename, filetype, and filemode of the file that contains the CULPRIT parameters for the run

data a  Note: Information about creating this file appears later in this appendix.

sysidms  filename, filetype, and filemode of the SYSIDMS parameters file

parms    Note: Information about creating this file appears later in this appendix.

sysjrnl  file name for the tape journal file, as assigned in the DMCL definition

uextwo   filename, filetype, and filemode of the unsorted extracted item data set

rk file a

uprmw    filename, filetype, and filemode of the unsorted parameter file

ork file a

Note: CULPRIT requires an external sort package (other than the z/VM and z/VM SORT command) that can be loaded dynamically.

Central Version Modifications

CA IDMS reports that run under central version can access an IDMS CV/DC system that is running in a z/VM and z/VM virtual machine. To identify the IDMS DC/UCF system to be accessed, take one of the following actions:

Specify CVMACH=cv-machine-name on the DATABASE parameter, where cv-machine-name is a 1- through 8-character user identifier of the z/VM and z/VM virtual machine in which the DC/UCF system is executing. For more information, see the CA Culprit for CA IDMS Reference section.

Code PARM='CVMACH=cv-machine-name' in the OSRUN command used to invoke the program. This option is valid only if the OSRUN command is issued from the System Product Interpreter or from an EXEC2 file.

Link edit the utility with an IDMSOPTI module that specifies CVMACH=cv-machine-name.

Creating the SYSIPT or SYSIDMS file

To create the SYSIPT file, enter these z/VM and z/VM commands:

XEDIT sysipt data a (NOPROF INPUT
.
.
.Source statements .
To create the SYSIDMS parameter file, substitute *sysidms parms a* in the example above.

⚠️ **Note:** For more information on all SYSIDMS parameters, see the *CA IDMS Common Facilities section.*
Endevor and DB for IDMS Reporting

This section provides more information on Reports, see the following topics:
- CCDB Reporting (see page 847)
- CA Culprit Modules (see page 847)
- Running Reports (see page 849)
- Report Formats (see page 856)

CCDB Reporting

The CA Endevor/DB Change Monitor creates an audit trail of all dictionary updates in the CCBD associated with any dictionary that has monitoring active. Extensive reporting is supplied in both batch and online modes. This section contains instructions for using the batch reporting facilities.

You can think of change log entries as a sequential log of all updates made to a dictionary. These log records are always associated with a dictionary and an entity occurrence. Optionally, they can be associated with a CCID, a user ID, or both. Batch reports are available to display the Change Log by Dictionary, User, or CCID, in both summarized and detailed forms.

Reporting modules are provided for the following CCDB entities:
- User
- Status
- Management group
- Security class
- CCID

CA Culprit Modules

Contents
- Common Modules (see page 848)
- Input Modules (see page 848)
- Output Modules (see page 848)

The following CA Culprit reporting modules are supplied with the CA Endevor/DB system:
- Common
- Input
Common Modules

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDVRPT10</td>
<td>CA Endevor/DB global fields. This module must be copied into all reports.</td>
</tr>
<tr>
<td>NDVRNA</td>
<td>A module that contains the company name. This name appears on all reports. The delivered module contains the following: <em><strong>PUT YOUR COMPANY NAME HERE</strong></em> Modify the module to reflect the name you wish to appear in all reports.</td>
</tr>
</tbody>
</table>

Input Modules

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>=COPY NDVRRCLG</td>
<td>Change-Log records input module</td>
</tr>
<tr>
<td>=COPY NDVRRCCI</td>
<td>CCID records input module</td>
</tr>
<tr>
<td>=COPY NDVRRUSR</td>
<td>User records input module</td>
</tr>
<tr>
<td>=COPY NDVRRSTA</td>
<td>Status records input module</td>
</tr>
<tr>
<td>=COPY NDVRRMGP</td>
<td>MANAGEMENT-GROUP records input module</td>
</tr>
<tr>
<td>=COPY NDVRRCLS</td>
<td>CCID/Entity/Status records input module</td>
</tr>
<tr>
<td>=COPY NDVRRSGN</td>
<td>SIGNIN/SIGNOUT records input module</td>
</tr>
<tr>
<td>=COPY NDVRRMGC</td>
<td>MGR/CCID relationships input module</td>
</tr>
</tbody>
</table>

Output Modules

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDVRPT01</td>
<td>Chronological Change Log Detail Report</td>
</tr>
<tr>
<td>NDVRPT02</td>
<td>User Change Log Detail Report</td>
</tr>
<tr>
<td>NDVRPT03</td>
<td>CCID Change Log Detail Report</td>
</tr>
<tr>
<td>NDVRPT04</td>
<td>Entity Change Log Detail Report</td>
</tr>
<tr>
<td>NDVRPT05</td>
<td>CCID Change Log Summary Report</td>
</tr>
<tr>
<td>NDVRPT06</td>
<td>CCID List</td>
</tr>
<tr>
<td>NDVRPT07</td>
<td>User Change Log Summary Report</td>
</tr>
<tr>
<td>NDVRPT08</td>
<td>User List</td>
</tr>
<tr>
<td>NDVRPT09</td>
<td>Status List</td>
</tr>
<tr>
<td>NDVRPT10</td>
<td>Management Group List</td>
</tr>
</tbody>
</table>
# Running Reports

**Contents**

- Selecting Input (see page 850)
  - Legend (see page 850)
- Report Groups (see page 851)
- Selection Criteria (see page 851)
- Selection Criteria Coding Rules (see page 852)
- Creating an Archive of Change Log Entries (see page 853)
- Using an Archive of Change Log Entries (see page 853)
- Coding Examples (see page 853)
- Sample JCL (see page 854)
  - Legend (see page 856)

Reports are requested by including the CA Endevor/DB global fields module (NDVRPT00), the company name (NDVRNAME), the input specification (NDVRRxxx), and the output specification (NDVRPTnn) indicating the desired reports. Optional selection criteria can also be applied to all reports. The =COPY command is used to include modules into an CA Culprit run. CA Endevor/DB allows you to run reports separately or in groups for substantial time savings.

To run a single report, modules must be specified as follows in the SYSIN data set:
To run multiple reports during a single session, list each report using an output specification (NDVRPTnn) as follows:

=COPY NDVRPT00
=COPY NDVRFxxx
=COPY NDVRPTnn

Selecting Input

The SYS010 file contains commands that direct the CA Endevor/DB reporting module to the appropriate input files. The SYS010 file can be omitted completely to report against the CCDB for the default dictionary or if appropriate parameters are specified in the SYSIDMS file. The following syntax can be specified in the SYS010 file:

---

DBName is dbname

SYSystem name is system

INPut is DATABASE
   ARCHIVE
   BOTH

Legend

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbn</td>
<td>DBNAME of the CCDB to report against. This parameter can be omitted for the primary dictionary. Optional.</td>
</tr>
<tr>
<td>syst</td>
<td>System Identifier of the CCDB to report against. Optional.</td>
</tr>
<tr>
<td>DAT</td>
<td>For Group 1 reports (See below), specify DATABASE to report against the current CCDB contents. DATABASE is the default.</td>
</tr>
<tr>
<td>ARC</td>
<td>For Group 1 reports (See below), specify ARCHIVE to report only against Change Log Entry files created by NDVRARCO (CA Endevor/DB archive/compress utility) or NDVRPT90 (CA Endevor/DB machine-readable report output).</td>
</tr>
<tr>
<td>BOT</td>
<td>For Group 1 reports (See below), specify BOTH to run reports against the current database and output from prior NDVRPT90 and/or NDVRARCO runs together. The CA Endevor/DB input module will read the archived files after processing the database.</td>
</tr>
</tbody>
</table>
Note: When BOTH or ARCHIVE are specified, a SYS011 DD statement must be included for input. See sample JCL below.

Report Groups

CA Endevor/DB reports are broken into eight general groups. Group 1 reports are all related to Change Log Activity, and can be run against the CCDB directly or from sequential Change Log Entry files created by CA Endevor/DB utilities. Report Groups 2-8 are lists of major CCDB entities.

<table>
<thead>
<tr>
<th>Report Group</th>
<th>Allowed Command</th>
<th>Input Module</th>
<th>Output Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DATABASE</td>
<td>=COPY NDVRR</td>
<td>=COPY NDVRPT01 =COPY NDVRPT02 =COPY NDVRPT03 =COPY NDVRPT04 =COPY NDVRPT05 =COPY NDVRPT07 =COPY NDVRPT11 =COPY NDVRPT12 =COPY NDVRPT14 =COPY NDVRPT15 =COPY NDVRPT16 =COPY NDVRPT17 =COPY NDVRPT24 =COPY NDVRPT90</td>
</tr>
<tr>
<td>2</td>
<td>DATABASE</td>
<td>=COPY NDVRR</td>
<td>=COPY NDVRPT06 =COPY NDVRPT22</td>
</tr>
<tr>
<td>3</td>
<td>DATABASE</td>
<td>=COPY NDVRR</td>
<td>=COPY NDVRPT08 =COPY NDVRPT21</td>
</tr>
<tr>
<td>4</td>
<td>DATABASE</td>
<td>=COPY NDVRR</td>
<td>=COPY NDVRPT09</td>
</tr>
<tr>
<td>5</td>
<td>DATABASE</td>
<td>=COPY NDVRR</td>
<td>=COPY NDVRPT10</td>
</tr>
<tr>
<td>6</td>
<td>DATABASE</td>
<td>=COPY NDVRR</td>
<td>=COPY NDVRPT13</td>
</tr>
<tr>
<td>7</td>
<td>DATABASE</td>
<td>=COPY NDVRR</td>
<td>=COPY NDVRPT18 =COPY NDVRPT19 =COPY NDVRPT20 =COPY NDVRPT25 =COPY NDVRPT26 =COPY NDVRPT27</td>
</tr>
<tr>
<td>8</td>
<td>DATABASE</td>
<td>=COPY NDVRR</td>
<td>=COPY NDVRPT23</td>
</tr>
</tbody>
</table>

Selection Criteria

Reports will display all records in the CCDB unless additional selection criteria are given to limit the volume. Selection criteria are specified in the SYSIN data set according to standard CA Culprit conventions. Commonly used selection criteria are outlined below. Additionally, any field in the database may be used as a selector. Appendix B contains the record layout used by CA Endevor/DB reports.
### Criteria Description

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOG-DATE</td>
<td>Date of change log entries. Rules:</td>
</tr>
<tr>
<td></td>
<td>Specified as yyyymmdd</td>
</tr>
<tr>
<td></td>
<td>Must not be surrounded by single quotes</td>
</tr>
<tr>
<td></td>
<td>Example: LOG-DATE GT 19861201</td>
</tr>
<tr>
<td>ENTITY-NAME</td>
<td>Name of entity in change log entries. Rules:</td>
</tr>
<tr>
<td></td>
<td>1-40 character name</td>
</tr>
<tr>
<td></td>
<td>Must be surrounded by single quotes</td>
</tr>
<tr>
<td></td>
<td>Example: ENTITY-NAME EQ 'ADSO-STAT'</td>
</tr>
<tr>
<td>ENTITY-VERSION</td>
<td>Version of entity in change log entries. Rules:</td>
</tr>
<tr>
<td></td>
<td>4-digit number 0-9999</td>
</tr>
<tr>
<td></td>
<td>Must not be surrounded by single quotes</td>
</tr>
<tr>
<td></td>
<td>Example: ENTITY-VERSION NE 1</td>
</tr>
<tr>
<td>USER</td>
<td>User name in change log entries. Rules:</td>
</tr>
<tr>
<td></td>
<td>1-32 character name</td>
</tr>
<tr>
<td></td>
<td>Must be surrounded by single quotes</td>
</tr>
<tr>
<td></td>
<td>Example: USER EQ 'JONES'</td>
</tr>
<tr>
<td>ACTION</td>
<td>Action in change log entries. Rules:</td>
</tr>
<tr>
<td></td>
<td>1-character code: A, M, D, R, S, T, P, C (add, modify, delete, status, reset status set, status terminate, purged, and compressed)</td>
</tr>
<tr>
<td></td>
<td>Must be surrounded by single quotes</td>
</tr>
<tr>
<td></td>
<td>Example: ACTION EQ 'D'</td>
</tr>
</tbody>
</table>

### Selection Criteria Coding Rules

The following is a listing of selection criteria coding rules:

- SEL statement must occur after the last output module specified (See JCL below) in the input stream.
- SEL must be coded in column 2.
- The phrase WHEN must follow the word SEL (i.e., SEL WHEN).
- Selection parameters can immediately follow the selection phrase (i.e., SEL WHEN USER EQ 'JONES').
- Boolean logic for selection criteria is supported as follows:
  - LT -- Less than
  - LE -- Less than or equal
  - GT -- Greater than
  - GE -- Greater than or equal
  - NE -- Not equal to
  - EQ -- Equal to
  - AND -- And
  - OR -- Or
- To continue a selection request to multiple lines, place an "*" in column 1 of the second, third, and subsequent selection cards.
Once the SEL has been coded in column 2, the coding of all subsequent selection criteria is free-form.

Creating an Archive of Change Log Entries

CA Endevor/DB module NDVRPT90 creates a machine-readable sequential output file consisting of Change Log Entries formatted as described in Appendix B. Change Log Entries archived through NDVRPT90 are not deleted from the CCDB. To archive as well as delete from the CCDB, the NDVRARCO utility should be run by the CCDB administrator. When running the NDVRPT90 module, archived Change Log Entries will be placed in the SYS020 file during Step 2 of the CA Culprit run. See sample JCL below.

Using an Archive of Change Log Entries

All CA Endevor/DB Group I reports are capable of reading the current CCDB or sequential Change Log Entry files created by NDVRARCO or NDVRPT90. To include previously archived output in reports, include an 'INPUT IS ARCHIVE' or 'INPUT IS BOTH' clause in the SYS010 file in CA Culprit Step 2.

Coding Examples

Example 1
To select all Change Log Entries between December 30, 1986 and January 5, 1987 from the CCDB for the primary dictionary, specify the following input to CA Culprit Step 1:

```
//SYSIN DD *
=Copy NDVRPT00
=Copy NDVRNAME
=Copy NDVRRCLG
=Copy NDVRPT01
Sel When LOG-DATE GE 19861230 AND
* LOG-DATE LE 19870115
```

Example 2
To request a report showing the Change Log Entries created by 'JONES' since November 15, 1986 in a secondary dictionary named NTSTDICT, specify the following input:

```
In Step 1:

//SYSIN DD *
=Copy NDVRPT00
=Copy NDVRNAME
=Copy NDVRRCLG
=Copy NDVRPT02
Sel When USER EQ 'JONES' AND
* LOG-DATE GE 19861115

In Step 2:

//SYS010 DD *
DBNAME = NTSTDICT
```
Example 3
To detect archive Change Log Entries older than November 1, 1986 from a dictionary named NTSTDICT, specify:

In Step 1:

```plaintext
//SYSIN DD *
COPY NDVRPT00
COPY NDVRNAME
COPY NDVRRCLG
COPY NDVRPT90
SEL WHEN LOG-DATE LT 19861101
```

In Step 2:

```plaintext
//SYS010 DD *
DBNAME IS NTSTDICT
//SYS020 DD DSN=OLD.CLE.ARCHIVE,DISP=(OLD,KEEP)
```

Example 4
To report against an archived change log for any dictionary:

In Step 1:

```plaintext
//SYSIN DD *
COPY NDVRPT00
COPY NDVRNAME
COPY NDVRRCLG
COPY NDVRPT04
```

In Step 2:

```plaintext
//SYS010 DD *
INPUT IS ARCHIVE
//SYS011 DD DSN=OLD.CLE.ARCHIVE,DISP=OLD
```

Sample JCL

The example below illustrates multiple-step CA Culprit execution. When using one-step CA Culprit, the ddnames SYS010, SYS011, and/or SYS020 are added to the single step as required.

```plaintext
//JOBNAME JOB YOUR.JOBCARD.INFORMATION//JOBLIB DD DISP=SHR,DSN=usercv.loadlib// DD DISP=SHR,DSN=ndvrdb.loadlib// DD DISP=SHR,DSN=ndms.
loadlib//
**********************************************************
//* CA ENDEVOR/DB CULPRIT REPORTS *
**********************************************************
//* ******************************************************************
//* CULP0 EXEC PGM=CULP0,REGION=2048K
//* SYSCALL DD DISP=SHR,DSN=ndms.sysctl//CULLIB DD DISP=SHR,DSN=ndv rdb.srclib//SYS004 DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
//* SYS005 DD DSN=6.6PRMWORK.,
// DISP=(NEW,PASS),UNIT=disk,
// DCB=(RECFM=FB,LRECL=320,BLKSIZE=1600),
// SPACE=(CYC,(2,1),RLSE,,ROUND)
//* SYSDUMP DD SYSOUT=*
//* SYSIDMS DD *
//* DMCL=dmcl-nameDICTNAME=dictionary-nameOTHER OPTIONAL SYSIDMS PARAMETERS/*
//* SYSIN DD *
//* COPY NDVRPT00
//* COPY NDVRNAME
//* COPY NDVRxx=COPY NDVRPTnn/*
```
CA IDMS - 19.0

/*
*CULP1
EXEC PGM=SORT,PARM='MSG=AP',REGION=2048K
/SORTLIB DD DSN=sortlib,SYSLIB DD SHR,DSN=yourHLQ.CAGJSRC(SORT1)
/*
/SORTIN DD DISP=(OLD,DELETE),DSN=&.&UPRMWORK.
/SORTWK01 DD DSN=&.&WRKAWORK.,UNIT=disk,
// SPACE=(CYL,(2,1),RLSE,ROUND)
/SORTWK02 DD DSN=&.&WRKBWORK.,UNIT=disk,
// SPACE=(CYL,(2,1),RLSE,ROUND)
/SORTWK03 DD DSN=&.&WRKCWORK.,UNIT=disk,
// SPACE=(CYL,(2,1),RLSE,ROUND)
/SORTWK04 DD DSN=&.&WRKDWORK.,UNIT=disk,
// SPACE=(CYL,(2,1),RLSE,ROUND)
/SORTOUT DD DSN=&.&SPRMWORK.,
// DISP=(NEW,PASS),UNIT=disk,
// DCB=(RECFM=F,LRECL=320,BLKSIZE=320),
// SPACE=(CYL,(2,1),RLSE,ROUND)
//SYSDUMP DD SYSOUT=* SYSPRINT DD SYSOUT=*
/*
*CULP2
EXEC PGM=CULL,REGION=2048K
/SYSCTL DD DISP=SHR,DSN=idms.sysctl//SYS004 DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
//SYS005 DD DISP=(OLD,DELETE),DSN=&.&SPRMWORK.
//SYS006 DD DSN=&.&UEXTWORK.,
// DISP=(NEW,PASS),UNIT=disk,
// DCB=(RECFM=VB,LRECL=1024,BLKSIZE=1028),
// SPACE=(CYL,(2,1),RLSE,ROUND)
//SYS008 DD DSN=&.&WSRTWORK.,
// DISP=(NEW,PASS),UNIT=disk,
// DCB=(RECFM=VB,LRECL=1024,BLKSIZE=1028),
// SPACE=(CYL,(2,1),RLSE,ROUND)
//SYS007 DD DSN=&.&SPRTWORK.,
// DISP=(NEW,PASS),UNIT=disk,
// DCB=(RECFM=F,LRECL=80,BLKSIZE=80),
// SPACE=(CYL,(2,1),RLSE,ROUND)
//SYS011 DD DSN=indsn,DISP=OLD ARCHIVE INPUT FILE
//SYS020 DD DSN=outsdsn,DISP=(NEW,KEEP), ARCHIVE OUTPUT FILE
// DCB=(LRECL=288,BLKSIZE=14400,RECFM=FB),
// UNIT=tape//SYSDUMP DD SYSOUT=* SYSPRINT DD SYSOUT=*
/*
SYSIDMS DD *
DMCL=dmcl-name DICTNAME=dictionary-name OTHER OPTIONAL SYSIDMS PARAMETERS/*
//SYS010 DD * DBNAME IS DBNAME INPUT IS BOTH/*
/*
*CULP3
EXEC PGM=SORT,PARM='MSG=AP',REGION=2048K
/SORTLIB DD DSN=sortlib,DISP=SHR
/SYSIN DD DSN=&.&SPRTWORK.,DISP=(OLD,DELETE)
/SORTIN DD DSN=&.&UEXTWORK.,DISP=(OLD,DELETE)
/SORTWK01 DD DSN=&.&WRKAWORK.,UNIT=disk,
// SPACE=(CYL,(2,1),RLSE,ROUND)
/SORTWK02 DD DSN=&.&WRKBWORK.,UNIT=disk,
// SPACE=(CYL,(2,1),RLSE,ROUND)
/SORTWK03 DD DSN=&.&WRKCWORK.,UNIT=disk,
// SPACE=(CYL,(2,1),RLSE,ROUND)
/SORTWK04 DD DSN=&.&WRKDWORK.,UNIT=disk,
// SPACE=(CYL,(2,1),RLSE,ROUND)
/SORTOUT DD DSN=&.&SRTWORK.,
// DISP=(NEW,PASS),UNIT=disk,
// DCB=(RECFM=FB,LRECL=1024,BLKSIZE=1028),
// SPACE=(CYL,(2,1),RLSE,ROUND)
//SYSOUT DD SYSOUT=* SYSPRINT DD SYSOUT=*
/*
*CULP4
EXEC PGM=CULE,REGION=2048K
//SYS004 DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133)
//SYS006 DD DISP=(OLD,DELETE),DSN=&.&SRTWORK.
Legend

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usercv.loadlib</td>
<td>Load library for DMCL, subschema, DBNAME table, or load modules for a particular CA IDMS/CV platform.</td>
</tr>
<tr>
<td>ndvrdb.loadlib</td>
<td>CA Endevor/DB load library.</td>
</tr>
<tr>
<td>idms.loadlib</td>
<td>CA IDMS load library.</td>
</tr>
<tr>
<td>idms.sysctl</td>
<td>Name of the SYSCTL file as defined in the CA IDMS system generation.</td>
</tr>
<tr>
<td>dmcl-name</td>
<td>Name of the DMCL used by CA IDMS/CV.</td>
</tr>
<tr>
<td>dictname-name</td>
<td>CCDB dictname whose contents are to be reported.</td>
</tr>
<tr>
<td>sortlib</td>
<td>Name of your sort package load library.</td>
</tr>
<tr>
<td>yourHLQ.</td>
<td>Name of the CA IDMS install source library</td>
</tr>
<tr>
<td>CAGJSRC</td>
<td></td>
</tr>
<tr>
<td>ndvrdb.srclib</td>
<td>Source library containing CA Endevor/DB install library containing the CA Culprit source modules.</td>
</tr>
<tr>
<td>disk</td>
<td>Installation unit name for temporary disk space.</td>
</tr>
<tr>
<td>indsn</td>
<td>Data set name for an input sequential archive file.</td>
</tr>
<tr>
<td>outdsn</td>
<td>Data set name for an output sequential archive file.</td>
</tr>
<tr>
<td>NDVRRxxx</td>
<td>Input module name</td>
</tr>
<tr>
<td>NDVRRPTnn</td>
<td>Output module name</td>
</tr>
</tbody>
</table>

Report Formats

For more Report Formats, see the following topics:
- Report No. 00 Reporting Commands and Messages (see page 857)
- Report No. 01 Change Log Detail (see page 857)
- Report No. 02 User Change Log Detail (see page 858)
- Report No. 03 CCID Change Log Detail (see page 859)
- Report No. 04 Entity Change Log Detail (see page 860)
- Report No. 05 CCID Change Log Summary (see page 860)
- Report No. 06 CCID List (see page 861)
- Report No. 07 User Change Log Summary (see page 861)
- Report No. 08 User List (see page 862)
CA IDMS - 19.0

- Report No. 09 Status List (see page 863)
- Report No. 10 Management Group List (see page 863)
- Report No. 11 Entity / CCID Change Log Summary (see page 864)
- Report No. 12 Entity Change Log Summary (see page 865)
- Report No. 13 CCID / Entity Status (see page 866)
- Report No. 14 Entity Status History (see page 867)
- Report No. 15 Post Migration Activity (see page 867)
- Report No. 16 Source Migration Summary (see page 868)
- Report No. 17 Target Migration Summary (see page 869)
- Report No. 18 Entity Signout Detail (see page 869)
- Report No. 19 Entity Signout by User (see page 870)
- Report No. 20 Entity Signout by CCID (see page 870)
- Report No. 21 Security Class/User (see page 871)
- Report No. 22 Security Class/CCID (see page 871)
- Report No. 23 Management Group/CCID (see page 871)
- Report No. 24 Entity/User Change Log Summary (see page 872)
- Report No. 25 Entity Preauthorization Detail Report (see page 873)
- Report No. 26 Entity Preauthorization by User Report (see page 874)
- Report No. 27 Entity Preauthorization by CCID Report (see page 874)

Report No. 00 Reporting Commands and Messages

volser

RELEASE nn.n
/DB

mm/dd/yy 12:30:49 0001

REPORTING COMMANDS AND MESSAGES

CA ENDEVOR/DB nn.n volser

CHANGE LOG FOR DICTIONARY: SRCNDVR SYSTEM: SYSTEM74
DATE --- TIME --- ACTION -------- ENTITY TYPE ---- ENTITY IDENTIFICATION ------
mm/dd/yy 07:51: 55 O (SIGNOUT) MODULE NDVRPT01
1 DBADMIN

SIGNED TO: DBADMIN

mm/dd/yy 07:51: 56 O (SIGNOUT) MODULE NDVRPT02
1 DBADMIN

SIGNED TO: DBADMIN

mm/dd/yy 09:31: 50 M (COMPRESSED) DICTIONARY SRCNDVR

Report No. 01 Change Log Detail

This report contains one line for each IDD or CCDB entity change made. Change Log Entries are displayed in chronological order with any associated CCIDs or USERIDs. The dates shown here are mm/dd/yy. These are dates in the actual report.

REPORT NO. 01

CA ENDEVOR/DB nn.n volser

CHANGE LOG DETAIL REPORT

** PUT YOUR COMPANY NAME HERE **

mm/dd/yy 07:51: 55 O (SIGNOUT) MODULE NDVRPT01
1 DBADMIN

SIGNED TO: DBADMIN

mm/dd/yy 07:51: 56 O (SIGNOUT) MODULE NDVRPT02
1 DBADMIN

SIGNED TO: DBADMIN

mm/dd/yy 09:31: 50 M (COMPRESSED) DICTIONARY SRCNDVR
**Report No. 02 User Change Log Detail**

This report contains one line for each IDD or CCDB entity update made. Change Log Entries are displayed in user and chronological sequence with any associated CCIDs.

### Changes Made by User: DBADMIN

<table>
<thead>
<tr>
<th>mm/dd/yy</th>
<th>Action</th>
<th>Time</th>
<th>Entity Type</th>
<th>Entity Identification</th>
<th>DICT</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>07:51:55</td>
<td>SIGNOUT</td>
<td>07:51:51</td>
<td>MODULE</td>
<td>NDVRPT01</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SRCNDVR</td>
<td></td>
<td></td>
<td>SYSTEM74</td>
<td></td>
<td></td>
<td>DBADMIN</td>
</tr>
<tr>
<td>07:51:56</td>
<td>SIGNOUT</td>
<td>07:51:51</td>
<td>MODULE</td>
<td>NDVRPT02</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SRCNDVR</td>
<td></td>
<td></td>
<td>SYSTEM74</td>
<td></td>
<td></td>
<td>DBADMIN</td>
</tr>
<tr>
<td>07:51:56</td>
<td>SIGNOUT</td>
<td>07:51:51</td>
<td>MODULE</td>
<td>NDVRPT03</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SRCNDVR</td>
<td></td>
<td></td>
<td>SYSTEM74</td>
<td></td>
<td></td>
<td>DBADMIN</td>
</tr>
<tr>
<td>07:51:56</td>
<td>SIGNOUT</td>
<td>07:51:51</td>
<td>MODULE</td>
<td>NDVRPT04</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SRCNDVR</td>
<td></td>
<td></td>
<td>SYSTEM74</td>
<td></td>
<td></td>
<td>EDB-DEVELOP</td>
</tr>
<tr>
<td>07:51:56</td>
<td>SIGNOUT</td>
<td>07:51:51</td>
<td>MODULE</td>
<td>NDVRPT05</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SRCNDVR</td>
<td></td>
<td></td>
<td>SYSTEM74</td>
<td></td>
<td></td>
<td>EDB-DEVELOP</td>
</tr>
<tr>
<td>09:53:30</td>
<td>COMPRESSED</td>
<td></td>
<td>DICTIONARY</td>
<td>SRCNDVR</td>
<td>1</td>
<td>SYSTEM74</td>
</tr>
</tbody>
</table>

---

**DATE: mm/dd/yy**
**TIME: 06:44:**

**DATE: mm/dd/yy**
**TIME: 07:36:**

**DATE: mm/dd/yy**
**TIME: 07:36:**

**DATE: mm/dd/yy**
**TIME: 07:36:**
### Report No. 03 CCID Change Log Detail

This report contains one line for each IDD or CCDB entity update made under a given CCID. Change Log Entries are displayed in entity name, user, and chronological sequence with any associated USERID.

<table>
<thead>
<tr>
<th>DATE</th>
<th>TIME</th>
<th>ACTION</th>
<th>ENTITY TYPE</th>
<th>ENTITY IDENTIFICATION</th>
<th>SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>mm/dd/yy 06:02:03</td>
<td>D (COMPRESSED)</td>
<td>ENDEVOR-USER</td>
<td>1</td>
<td>DBADMIN SRCNDVR SYSTEM74</td>
<td></td>
</tr>
<tr>
<td>mm/dd/yy 15:05:</td>
<td>M (MODIFY)</td>
<td>DICTIONARY</td>
<td>1</td>
<td>SRCNDVR</td>
<td></td>
</tr>
<tr>
<td>mm/dd/yy 14:18:01</td>
<td>A (ADD)</td>
<td>ELEMENT</td>
<td>1</td>
<td>SRCNDVR</td>
<td></td>
</tr>
<tr>
<td>mm/dd/yy 14:18:01</td>
<td>A (ADD)</td>
<td>RECORD</td>
<td>1</td>
<td>SRCNDVR</td>
<td></td>
</tr>
<tr>
<td>mm/dd/yy 06:33:14</td>
<td>C (MIGR AUDIT)</td>
<td>ELEMENT</td>
<td>1</td>
<td>SRCNDVR</td>
<td></td>
</tr>
</tbody>
</table>

**CHANGES MADE UNDER CCID: EDB-DEVELOP**

**PUT YOUR COMPANY NAME HERE**
Report No. 04 Entity Change Log Detail

This report contains one line for each IDD or CCDB update grouped by entity. Change Log Entries are displayed in entity name and chronological sequence with any associated USERID and CCID available.

CHANGE LOG FOR RECORD EDB-REC VERSION 1 WITHIN DICTIONARY: SRCNDVR SYSTEM: SYSTEM74

DATE --- TIME --- ACTION ------- USER --- CCID ------- ADDITIONAL CHANGE-RELATED INFORMATION

mm/dd/yy 14:18:01 A DBADMIN UPDATE PROGRAM: IDMSDDDL
mm/dd/yy 06:33:14 C (MIGR AUDIT) DBADMIN DATE: mm/dd/yy TIME: 07:36:

Report No. 05 CCID Change Log Summary

This report contains one entry for each update made to an entity under a given CCID. Multiple lines indicate more than one CCID updating for a given time interval.

CHANGE MADE UNDER CCID: EDB-SYSADMIN WITHIN DICTIONARY: SRCNDVR SYSTEM: SYSTEM74
Report No. 06 CCID List

This report contains one entry for each Change Control Identifier (CCID) defined in the data dictionary. CCIDs are automatically associated with work done in a IDD by the Dynamic Change Monitor.

REPORT NO. 06

CA ENDEVOR/DB nn.n volser

CCID LIST

** PUT YOUR COMPANY NAME HERE **

CCDB FOR DICTIONARY: SRCNDVR

SYSTEM: SYSTEM74

<table>
<thead>
<tr>
<th>CCID NAME</th>
<th>SECURITY CLASS</th>
<th>LOCKED</th>
<th>TYPE</th>
<th>DATE LAST</th>
<th>TIME LAST</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDB-DCADMIN NDVR-GLOBAL</td>
<td>PUBLIC</td>
<td>mm/dd/yy</td>
<td>07:49:22 E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/DB DC ADMINISTRATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDB-DEVELOP DEVELOPMENT</td>
<td>PUBLIC</td>
<td>mm/dd/yy</td>
<td>06:37:23 E/DB nn.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>n DEVELOPMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDB-QA</td>
<td>QA</td>
<td>mm/dd/yy</td>
<td>06:37:24 E/DB nn.n QA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDB-SUPPORT SUPPORT</td>
<td>PUBLIC</td>
<td>mm/dd/yy</td>
<td>06:37:24 E/DB nn.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>n TECHNICAL SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDB-SYSADMIN NDVR-GLOBAL</td>
<td>PUBLIC</td>
<td>mm/dd/yy</td>
<td>07:48:07 E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/DB SYSTEM ADMINISTRATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Report No. 07 User Change Log Summary

This report contains one line for each update made to an entity by a given user. Multiple lines for an entity signify a break in the sequence of Change Log Entries for that entity by that user.
Report No. 08 User List

This report contains one entry for each user defined to the CCDB. Each user entry is followed by the CCID(s) that will automatically be associated with work done in the data dictionary by the Dynamic Change Monitor.
Report No. 09 Status List

This report contains one line for each change control status contained in the CCDB. Status codes are associated with dictionary entities as milestones are reached in the development process.

REPORT NO. 09
CA, INC.
CA ENDEVOR/DB nn.n volser
status list

CCDB FOR DICTIONARY: SRCNDVR SYSTEM: SYSTEM74

STATUS LAST DATE LAST TIME COMMENT
DEBUGGING Y mm/dd/yy 08:03:
09 STATUS FOR PRIVATE DEBUGGING
EDB-ISSUE N mm/dd/yy 14:50:08 E/DB OPEN STAR ISSUE
MANUAL-ENTITY N mm/dd/yy 08:03:
12 STATUS FOR MANUALLY DEFINED ENTITIES
REPORTING N mm/dd/yy 08:18:31 E/DB nn.n REPORTING
TESTING N mm/dd/yy 08:03:
07 STATUS FOR PUBLIC TESTING

Report No. 10 Management Group List

This report contains one line for each management group contained in the CCDB. Management groups are a mechanism for grouping related CCIDs together, and are used for release and migration configuration purposes.

REPORT NO. 10
CA, INC.
CA ENDEVOR/DB nn.n volser
management group list

CCDB FOR DICTIONARY: SRCNDVR SYSTEM: SYSTEM74

MANAGEMENT GROUP LAST DATE LAST TIME COMMENT
PRODUCTION CTRL mm/dd/yy 08:34:
47 PRODUCTION CONTROL GROUP
DEVELOPMENT CTRL mm/dd/yy 08:35:
00 DEVELOPMENT CONTROL GROUP
QA CTRL mm/dd/yy 08:45:47 QA CONTROL GROUP
Report No. 11 Entity / CCID Change Log Summary

This report contains one line for each update made to an entity under a given CCID. More than one line under an entity signifies multiple CCID updates for the reported time interval.

<table>
<thead>
<tr>
<th>TYPE</th>
<th>CONTROL</th>
<th>RETRIEVE</th>
<th>ARCHIVED</th>
<th>DATE LAST</th>
<th>VERS</th>
<th>CCID</th>
<th>ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOAD-</td>
<td>ENTITY</td>
<td>ACTION</td>
<td>ACTION</td>
<td>ACTION</td>
<td>ACTION</td>
<td>ACTION</td>
<td></td>
</tr>
<tr>
<td>MODULE</td>
<td>ADSCSELB</td>
<td>mm/dd/yy</td>
<td>mm/dd/yy</td>
<td>mm/dd/yy</td>
<td>100</td>
<td>1</td>
<td>EDB-</td>
</tr>
<tr>
<td></td>
<td>SYSSADMIN</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>1</td>
</tr>
<tr>
<td>TABLE</td>
<td>ADSCSELB</td>
<td>mm/dd/yy</td>
<td>mm/dd/yy</td>
<td>mm/dd/yy</td>
<td>100</td>
<td>1</td>
<td>EDB-</td>
</tr>
<tr>
<td></td>
<td>SYSSADMIN</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>1</td>
</tr>
<tr>
<td>DIALOG</td>
<td>AUTODIAG</td>
<td>mm/dd/yy</td>
<td>mm/dd/yy</td>
<td>mm/dd/yy</td>
<td>100</td>
<td>1</td>
<td>EDB-</td>
</tr>
<tr>
<td></td>
<td>SYSSADMIN</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>1</td>
</tr>
<tr>
<td>MODULE</td>
<td>AUTUSER-FLD-HELP</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SYSSADMIN</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>1</td>
</tr>
<tr>
<td>HELP</td>
<td>AUTUSER-FLD-HELP</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SYSSADMIN</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>1</td>
</tr>
<tr>
<td>MODULE</td>
<td>AUTUSER-MAP-HELP</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SYSSADMIN</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>1</td>
</tr>
<tr>
<td>HELP</td>
<td>AUTUSER-MAP-HELP</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SYSSADMIN</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>1</td>
</tr>
<tr>
<td>RECORD</td>
<td>CUSTOMER</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>SYSSADMIN</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td>FILE</td>
<td>CUSTOMER-FILE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SYSSADMIN</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>1</td>
</tr>
<tr>
<td>MESSAGE</td>
<td>DC601086</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>SYSSADMIN</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>1</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>EMPINQ</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>SYSSADMIN</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>1</td>
</tr>
<tr>
<td>TASK</td>
<td>EMPINQ</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>SYSSADMIN</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>1</td>
</tr>
<tr>
<td>RECORD</td>
<td>EMPLOYEE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>SYSSADMIN</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>1</td>
</tr>
</tbody>
</table>
Report No. 12 Entity Change Log Summary

This report contains one line for each entity known to the CCDB and an accompanying update count and last update time.

<table>
<thead>
<tr>
<th>RECORD</th>
<th>EMPPMAP-WORK-RECORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 0 0 1 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>mm/dd/yy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SYSADMIN</th>
<th>EDB-SYSADMIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 0 0 0</td>
<td>0 1 0 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOAD-MODULE</th>
<th>EMPMAPP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 0 1 0</td>
<td>1 17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>mm/dd/yy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MAP</th>
<th>EDB-SYSADMIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1</td>
<td>0 1 0 0 1 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MODULE</th>
<th>EMPMAPP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>1 17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>mm/dd/yy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

**Report No. 12 Entity Change Log Summary**

This report contains one line for each entity known to the CCDB and an accompanying update count and last update time.

**REPORT NO. 12**

<table>
<thead>
<tr>
<th>CA, INC.</th>
</tr>
</thead>
</table>

**mm/dd/yy PAGE 1**

**CA ENDEVOR/DB nn.n volser**

**ENTITY CHANGE LOG SUMMARY REPORT**

**CCDB FOR DICTIONARY: SRCNDVR**

**SYSTEM: SYSTEM74**

**CHANGE CO**

**NTROL RETRIEVE ARCHIVED DATE LAST**

**TIONS ACTIONS ENTITY ACTIONS ACTION**

**VERSION ACTIONS AC**

<table>
<thead>
<tr>
<th>LOAD-</th>
<th>ADSCSELB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0</td>
<td>100 1 0 2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE</th>
<th>ADSCSELB</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 1</td>
<td>100 2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DIALOG</th>
<th>AUTODIAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>100 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MODULE</th>
<th>AUTODIAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>HELP</td>
<td>100 1 2 1 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MODULE</th>
<th>AUTODIAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>HELP</td>
<td>100 2 2 1 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RECORD</th>
<th>CUSTOMER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0</td>
<td>1 1 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FILE</th>
<th>CUSTOMER</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>1 1 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MESSAGE</th>
<th>DC601086</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>1 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>EMPINQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0</td>
<td>1 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TASK</th>
<th>EMPINQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>1 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RECORD</th>
<th>EMPPMAP-WORK-RECORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0</td>
<td>1 17 11 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MAP</th>
<th>EMPMAPP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>1 14</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MODULE</th>
<th>EMPMAPP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0</td>
<td>1 17</td>
</tr>
</tbody>
</table>

**mm/dd/yy**

<table>
<thead>
<tr>
<th>0</th>
</tr>
</thead>
</table>

16-Jan-2018
### Report No. 13 CCID / Entity Status

This report contains one line for each entity modified under a CCID and that entity's current status under that CCID, and its current "base" status. An entity may have one status for each CCID it has been modified under and one "base" status. Status codes can be used for workflow tracking/reporting, and promotion fine-tuning.

**Report No. 13 CCID / Entity Status**

<table>
<thead>
<tr>
<th>ENTITY STATUS CHANGED UNDER CCID: EMPDEMO WITHIN DICTIONARY: SRCNDVR SYST EM: SYSTEM74</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE LAST ACTION</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>RECORD</td>
</tr>
<tr>
<td>ELEMENT</td>
</tr>
<tr>
<td>ELEMENT</td>
</tr>
<tr>
<td>ELEMENT</td>
</tr>
<tr>
<td>ELEMENT</td>
</tr>
<tr>
<td>ELEMENT</td>
</tr>
<tr>
<td>ELEMENT</td>
</tr>
</tbody>
</table>
### Report No. 14 Entity Status History

This report displays one entry for each time the status of an entity has been set, reset, or terminated by CA Endevor/DB facilities.

<table>
<thead>
<tr>
<th>DATE</th>
<th>TIME</th>
<th>ACTION</th>
<th>STATUS</th>
<th>CCID</th>
<th>QUALIFIER</th>
<th>USER</th>
</tr>
</thead>
<tbody>
<tr>
<td>mm/dd/yy</td>
<td>15:41:03</td>
<td>S</td>
<td>STANDARD-MIGRATE</td>
<td>EMPDEMO</td>
<td>DBADMIN</td>
<td></td>
</tr>
</tbody>
</table>

### Report No. 15 Post Migration Activity

This report contains one line for each entity that was modified in the target dictionary since it was last migrated in. It is instrumental in identifying production or quality assurance "fixes" made independently of the development system. The Correlation Processor uses a similar extraction path when identifying the potential reversion of these fixes by new migrations.

<table>
<thead>
<tr>
<th>ACTION</th>
<th>DATE LAST</th>
<th>TIME LAST</th>
<th>ENTITY</th>
<th>TYPE</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOAD-MODULE</td>
<td>mm/dd/yy</td>
<td>07:42:10</td>
<td>EMPMAPP1</td>
<td>TAX-REC</td>
<td>1</td>
</tr>
</tbody>
</table>

**Note:** The table entries are extracted from the scanned document and may not be exact.
Report No. 16 Source Migration Summary

This report is run against the source system and contains one line for each entity received by the target system, the time it was received, and the time it was originally selected for migration (and optionally Signed out). Data for this report was originally contained in the confirmation file transmitted to the source system by the reception process on the target system. Also identified are the system name and dictionary name of the target. All entities participating in a migration are displayed together in entity name sequence.

<table>
<thead>
<tr>
<th>TYPE</th>
<th>TARGET</th>
<th>DATE LAST</th>
<th>VERSION</th>
<th>EXPORTED</th>
<th>EXPORTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE</td>
<td>/yy SYSTEM74</td>
<td>mm/dd/yy</td>
<td>100</td>
<td>1</td>
<td>mm/dd/yy</td>
</tr>
<tr>
<td>TABLE</td>
<td>/yy SYSTEM74</td>
<td>mm/dd/yy</td>
<td>100</td>
<td>1</td>
<td>mm/dd/yy</td>
</tr>
<tr>
<td>MODULE</td>
<td>/yy SYSTEM74</td>
<td>mm/dd/yy</td>
<td>HELP</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>MODULE</td>
<td>/yy SYSTEM74</td>
<td>mm/dd/yy</td>
<td>HELP</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>RECORD</td>
<td>/yy SYSTEM74</td>
<td>mm/dd/yy</td>
<td>100</td>
<td>1</td>
<td>mm/dd/yy</td>
</tr>
<tr>
<td>ELEMENT</td>
<td>/yy SYSTEM74</td>
<td>mm/dd/yy</td>
<td>1</td>
<td>1</td>
<td>mm/dd/yy</td>
</tr>
<tr>
<td>RECORD</td>
<td>/yy SYSTEM74</td>
<td>mm/dd/yy</td>
<td>1</td>
<td>1</td>
<td>mm/dd/yy</td>
</tr>
<tr>
<td>MESSAGE</td>
<td>/yy SYSTEM74</td>
<td>mm/dd/yy</td>
<td>1</td>
<td>1</td>
<td>mm/dd/yy</td>
</tr>
<tr>
<td>LOAD-MODULE</td>
<td>/yy SYSTEM74</td>
<td>mm/dd/yy</td>
<td>1</td>
<td>1</td>
<td>mm/dd/yy</td>
</tr>
<tr>
<td>MAP</td>
<td>/yy SYSTEM74</td>
<td>mm/dd/yy</td>
<td>1</td>
<td>1</td>
<td>mm/dd/yy</td>
</tr>
<tr>
<td>SCHEMA</td>
<td>/yy SYSTEM74</td>
<td>mm/dd/yy</td>
<td>100</td>
<td>1</td>
<td>mm/dd/yy</td>
</tr>
<tr>
<td>SUBSCHEMA</td>
<td>/yy SYSTEM74</td>
<td>mm/dd/yy</td>
<td>100</td>
<td>1</td>
<td>mm/dd/yy</td>
</tr>
</tbody>
</table>
Report No. 17 Target Migration Summary

This report is produced on the target system and contains one line for each entity received from the source system, the date and time it was selected for migration on the source system, and the time it was received on the target. The Reception Processor created the data necessary for this report. All entities participating in a migration are displayed in entity name sequence.

Report No. 18 Entity Signout Detail

This report displays each entity currently signed out, and the user or CCID to which that entity is signed out.
### Report No. 19 Entity Signout by User

This report displays each entity currently signed out, by user.

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ENTITY</th>
<th>VERS</th>
<th>OUT ON</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODULE</td>
<td>NDVRPT01</td>
<td>1</td>
<td>mm/dd</td>
<td></td>
</tr>
<tr>
<td>/yy CCID</td>
<td>SIGNOUT CULPRIT</td>
<td>1</td>
<td>mm/dd</td>
<td></td>
</tr>
</tbody>
</table>

### Report No. 20 Entity Signout by CCID

This report displays each entity currently signed out, by CCID.

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ENTITY</th>
<th>VERS</th>
<th>OUT ON</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODULE</td>
<td>NDVRPT02</td>
<td>1</td>
<td>mm/dd</td>
<td></td>
</tr>
<tr>
<td>/yy CCID</td>
<td>SIGNOUT CULPRIT</td>
<td>1</td>
<td>mm/dd</td>
<td></td>
</tr>
</tbody>
</table>

---
Report No. 21 Security Class/User

This report displays each user that is associated to a given security class.

```
REPORT NO. 21  mm/dd/yy PAGE 1  CA, INC.
CA ENDEVOR/DB nn.n volser            SECURITY CLASS/USER REPORT
USERS DEFINED WITH SECURITY CLASS: NDVR-GLOBAL IN DICTIONARY: SRCNDVR SYSTEM: SYSTEM74
-----------------------------------------------
USER NAME ----------------------  COMMENT ------------------
-------------------------------
EDB-DC-ADMINISTRATOR          E/DB DC ADMINISTRATOR
EDB-SYSTEM-ADMINISTRATOR      E/DB SYSTEM ADMINISTRATOR
DBADMIN                        DATABASE ADMINISTRATOR
```

Report No. 22 Security Class/CCID

This report displays each CCID that is associated to a given security class.

```
REPORT NO. 22  mm/dd/yy PAGE 1  CA, INC.
CA ENDEVOR/DB nn.n volser            SECURITY CLASS/CCID REPORT
CCIDS DEFINED WITH SECURITY CLASS: NDVR-GLOBAL IN DICTIONARY: SRCNDVR SYSTEM: SYSTEM74
-----------------------------------------------
CCID -------  COMMENT --------------------------
--------------------------
EDB-DCADMIN            E/DB DC ADMINISTRATION
EDB-SYSDADMIN          E/DB SYSTEM ADMINISTRATION
```

Report No. 23 Management Group/CCID

This report displays each CCID that is associated to a given management group.

```
REPORT NO. 23  mm/dd/yy PAGE 1  CA, INC.
CA ENDEVOR/DB nn.n volser            MANAGEMENT GROUP/CCID REPORT
CCIDS WITHIN MANAGEMENT GROUP: QA_CTRL IN DICTIONARY: SRCNDVR SYSTEM: SYSTEM74
-----------------------------
CCID -------  DATE LAST  TIME LAST  COMMENT -----------------
-------------------------------
EDB-DEVELOP  mm/dd/yy  08:45:47 E
/DB DEVELOPMENT CHANGE CONTROL
EDB-SUPPORT  mm/dd/yy  08:45:47 E
/DB SUPPORT CHANGE CONTROL
```
Report No. 24 Entity/User Change Log Summary

This report contains one line for each contiguous string of updates made to an entity under a given User. More than one line under an entity signifies multiple User updates for the reported time interval.

<table>
<thead>
<tr>
<th>TYPE</th>
<th>NS</th>
<th>DESTINATION</th>
<th>ACTION</th>
<th>CBS</th>
<th>ABC</th>
<th>1</th>
<th>EDB-</th>
<th>ACTION</th>
<th>CBS</th>
<th>ACTION</th>
<th>ACTION</th>
<th>ACTION</th>
<th>ACTION</th>
<th>ACTION</th>
<th>ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>ADSCSELB</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>DCADMIN</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>DCADMIN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>ADSCSELB</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>DBADMIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>ADSCSELB</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>DBADMIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>LOAD-MODULE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>100 AUTOUSER</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>EDB-SYST</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>LOAD-MODULE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>DBADMIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>DIALOG</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>100 AUTOUSER</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>EDB-SYST</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>LOAD-MODULE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>100 AUTOUSER</td>
<td>6</td>
<td>AUTOHELP</td>
<td>AUTOHELP</td>
<td>AUTOHELP</td>
<td>AUTOHELP</td>
<td>AUTOHELP</td>
<td>AUTOHELP</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>LOAD-MODULE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>AUTOUSER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>LOAD-MODULE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>AUTOUSER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>LOAD-MODULE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>AUTOUSER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>LOAD-MODULE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>AUTOUSER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>DIALOG</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>100 AUTOUSER</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>EDB-SYST</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>DIALOG</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>EDB-SYST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>DIALOG</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>DBADMIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>DIALOG</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>DBADMIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>DIALOG</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>mm/dd/yy</td>
<td>DBADMIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

16-Jan-2018 872/898
Report No. 25 Entity Preauthorization Detail Report

This report displays each entity preauthorized to entities and CCIDs, by entity.

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ENTITY</th>
<th>VERSION</th>
<th>AUTHED</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUT TO NAME</td>
<td>ADSCSELB</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>N CCID MODULE</td>
<td>EDB-SYSADMIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HELP RECORD</td>
<td>CUSTOMER</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>N CCID FILE</td>
<td>EDB-SYSADMIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N CCID FILE</td>
<td>CUSTOMER</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>N CCID FILE</td>
<td>EDB-SYSADMIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N CCID FILE</td>
<td>EDB-SYSADMIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N CCID ELEMENT</td>
<td>EDB-SYSADMIN</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>N CCID ELEMENT</td>
<td>EDB-SYSADMIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 ELEMENT</td>
<td>EDB-SYSADMIN</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>2 ELEMENT</td>
<td>EDB-SYSADMIN</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>N CCID MESSAGE</td>
<td>DCG601086</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>N CCID MESSAGE</td>
<td>DCG601086</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>N CCID PROGRAM</td>
<td>EMPINQ</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>N CCID PROGRAM</td>
<td>EMPINQ</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>N CCID TASK</td>
<td>EMPINQ</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>N CCID TASK</td>
<td>EMPINQ</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>N CCID LOAD-MODULE</td>
<td>EMPMAP</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>N CCID LOAD-MODULE</td>
<td>EMPMAP</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>N CCID RECORD</td>
<td>EDB-SYSADMIN</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>N CCID RECORD</td>
<td>EDB-SYSADMIN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PUT YOUR COMPANY NAME HERE**
Report No. 26 Entity Preauthorization by User Report

This report displays each entity preauthorized to entities and CCIDs, by user.

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ENTITY</th>
<th>VERSION</th>
<th>AUTHED</th>
<th>OUT</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODULE</td>
<td>NDVRPT01</td>
<td>1</td>
<td>Y</td>
<td>Y</td>
<td>SIGNOUT</td>
</tr>
<tr>
<td>CULPRIT REPORT 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODULE</td>
<td>NDVRPT02</td>
<td>1</td>
<td>Y</td>
<td>Y</td>
<td>SIGNOUT</td>
</tr>
<tr>
<td>CULPRIT REPORT 02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODULE</td>
<td>NDVRPT03</td>
<td>1</td>
<td>Y</td>
<td>Y</td>
<td>SIGNOUT</td>
</tr>
<tr>
<td>CULPRIT REPORT 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Report No. 27 Entity Preauthorization by CCID Report

This report displays each entity preauthorized to entities and CCIDs, by CCID.

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ENTITY</th>
<th>VERSION</th>
<th>AUTHED</th>
<th>OUT</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODULE</td>
<td>NDVRPT04</td>
<td>1</td>
<td>Y</td>
<td>Y</td>
<td>SIGNOUT</td>
</tr>
<tr>
<td>CULPRIT REPORT 04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODULE</td>
<td>NDVRPT05</td>
<td>1</td>
<td>Y</td>
<td>Y</td>
<td>SIGNOUT</td>
</tr>
<tr>
<td>CULPRIT REPORT 05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How to Save Your Report

In this section This section shows you how to save your current report. Saved reports differ from other CA OLQ functions in that they:

- **Reflect a static picture of the data** in the database when the report is created.
- **Retain all of the editing and formatting features** you have built into your report, such as pictures, headers, and column spacing.
- **Retain any group calculations or calculated columns.**
- **Can be accessed by any other CA OLQ user,** given the proper security.

Once you have saved a report, you cannot change it by:

- Changing your selection of rows and columns
- Reaccessing the database to include more recent data

However, you can change it by modifying how columns or rows are displayed.

**Report retention period**

When you save your report, you specify how long you want to keep the report available online. The Saved Report screen has a **Retention period** field, in which you can specify how many days you want to keep the report file.

**Creating a Saved Report:**
Saved reports are contrasted with other CA OLQ functions in the following table.

<table>
<thead>
<tr>
<th>What You Want To Do</th>
<th>Which CA OLQ Function To Use</th>
<th>Formatting Retained?</th>
<th>Calculations Reexecuted?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save the set of commands used to build a report</td>
<td>Qfile</td>
<td>Retained</td>
<td>Reexecuted each time the report is built</td>
</tr>
<tr>
<td>The report created by a qfile reflects the data that is in the database when the qfile is executed. The contents of the report change as the data changes.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Save a copy of a report for later use</td>
<td>Saved report</td>
<td>Retained</td>
<td>Executed at report definition</td>
</tr>
<tr>
<td>The report reflects the data as it was when the report was executed. The data doesn't change.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create a table from your report</td>
<td>Saved table</td>
<td>Not retained</td>
<td>Retained as they are computed when the table is saved</td>
</tr>
<tr>
<td>The table can later be used to create other reports, or can be joined with other tables</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For more information, see the following topics:
- OLQ Report Key Terms (see page 877)
- Creating an OLQ Report (see page 877)
- Saving the Report (see page 877)
- Using a Saved Report (see page 879)
- Modifying a Saved Report (see page 880)
- Deleting Saved Reports (see page 882)
OLQ Report Key Terms

Some terms used to discuss saved reports are:

Current report

The report you're working on in an active CA OLQ session. If you retrieve a saved report, CA OLQ clears out the current report.

Retention period

The number of days your saved report file is kept in your directory. After the retention period has expired, the report file is deleted.

Saved report

A report file maintained in your user directory that contains a copy of a CA OLQ report.

Creating an OLQ Report

In this example, you create a report using the following steps:

1. Build a Salaries report containing employee names, social security numbers, and salary information.

2. Modify your report headers.

3. Format the numeric values in the report.

4. Sort the report rows by salary grade.

5. Group the report rows by salary grade and calculate the average salary in each group.

Saving the Report

In this example, you save the current report in a saved report file.

From the Display Report screen, proceed to the Saved Reports screen by typing save in the command line.

Specify Create. Name your report salaries and specify that you want to retain the report for 5 days.
Enter report name: **salaries**  
Report Name               Time             Date             Retention

No saved reports are available

1=HELP             3=QUIT             4=MESSAGE             6=MENU             PA2=REFRESH

Specify that you want to view an existing report.

--- Data Source for Report ---
   Choose tables             TABLE
   Choose subschema             SUBschema

--- Retrieval Activity ---
   Choose records from selected subschema            RCord
   Choose columns for report            COLumn
   Retrieve data to build report            RETrieve
   Alter database access strategy            LINkage

--- Processing Mode ---
   Execute or create a predefined routine            QFile
   Submit batch report request            BATch

1=HELP             2=GLOBAL HELP             3=QUIT             4=MESSAGE             8=FWD

CA OLQ lists the SALARIES report and its retention period.

--- Saved Reports ---

--- View existing or save current report ---
   SAVE
   x View existing or save current report

CA OLQ lists the SALARIES report and its retention period.

1=HELP             3=QUIT             4=MESSAGE             6=MENU             PA2=REFRESH
Using a Saved Report

In this example, you retrieve a saved report. Remember that when you retrieve a saved report, it overrides your current report.

Start on the Saved Reports screen. To get there, type `save` in the command line.

Choose **Select**. Select the SALARIES report.

```
CA OLQ Release nn.n
->
120000 Select report function, report name, and press the ENTER key

Saved reports for user: DOC1
Function: c Select _ Create _ Replace
 _ Delete _ Delete all
Enter report name: c SALARIES
Report Name Retention period: 1
```

CA OLQ displays the SALARIES report. Note that all of your formatting and grouping features have been retained in the report definition.

```
- CA OLQ Release nn.n
->
125000 Press the ENTER key to go to the next page of the report.

EMPLOYEE/EMPICION REPORT
   mm/dd/yy

   NAME      SOCIAL SECURITY NUMBER  SALARY GRADE  SALARY
   FITZHUGH  11-234-5678       11     13000.00
   TERNER    04-567-2222       11     13000.00
   JOHNSON   01-134-7878       11     13500.00
     AVE FOR 11:  13166.66
   NICEMAN   03-345-6110       12     14000.00
   GARDNER   02-233-4444       12     14000.00
   KRAAMER   02-378-6666       12     14000.00
   KING      06-784-5516       12     14500.00
     AVE FOR 11:  13166.66
```

CA OLQ - 19.0
Modifying a Saved Report

In this example, you:

- Retrieve the SALARIES report
- Change the report to display only those rows containing summary computations
- Replace the report file

Start on the Saved Reports screen. To get there, type save in the command line.

Choose Select. Select the SALARIES report.

Specify picture on the command line. CA OLQ displays the SALARIES report.
Add a leading dollar sign and commas to your computation by selecting the $ and , options next to the listing of the computation syntax.

```
137000 Specify pictures and press the ENTER key

EMPLOYEE
X EMP-LAST-NAME-0415 1 X(15)
X SS-NUMBER-0415 2 _ _ _ 99-999-9999
EMPPOSITION
X SALARY-GRADE-0420 3 _ _ _ 99
X SALARY-AMOUNT-0420 4 _ _ _ -ZZZZZ9.99

COMPUTE FIELDS:
X SALARY-AMOUNT-0420-AVE-3=AVE(SALARY-AMOUNT-0420) GROUP BY SALARY-GRADE-0420 LEVEL 1
```

Compute:

```
1=HELP 3=QUIT 4=MESSAGE 5=DISPLAY 6=MENU 10=EDIT 11=GROUP BY

Your summary computation is listed with a leading dollar sign and separating commas.

```
AVE FOR 11: $13,166.66
```

Select Replace. Select the SALARIES saved report.

```
120000 Select report function, report name, and press the ENTER key
DOC1
Function: Select Create x Replace
_ Delete _ Delete all
```
Deleting Saved Reports

Contents
- Deleting All Saved Reports (see page 883)

In this example, you delete the SALARIES saved report. Start on the Saved Reports screen. To get there, type `save` in the command line.

Select **Delete**. Select the SALARIES saved report.

CA OLQ returns you to the Menu screen and issues a message indicating that your report has been saved.

**CA OLQ** returns you to the Menu screen and issues a message indicating that your report has been saved.

```
1=HELP     3=QUIT     4=MESSAGE     6=MENU     PA2=REFRESH
```

Deleting Saved Reports

In this example, you delete the SALARIES saved report. Start on the Saved Reports screen. To get there, type `save` in the command line.

Select **Delete**. Select the SALARIES saved report.

```
1=HELP     3=QUIT     4=MESSAGE     8=FWD
```
Deleting All Saved Reports

The **Delete all** function enables you to delete all of the reports that were created under a given user ID. To delete all of the reports that belong to you:

1. Go to the Saved Reports screen by typing `save` in the command line
2. Specify your user ID (for example, JTE) in the **Saved reports for user** field
3. Select **Delete all**
4. Press [Enter]
Modifying CA IDMS Reports

CA IDMS reports can be modified, either temporarily or permanently, to suit site-specific needs. Some common modifications users make are:

- Reformatting a report (for example, sorting the data differently, changing column sequence, reducing the number of characters per line)
- Changing report content (for example, adding or deleting columns, rewording headings, and modifying process operations)
- Selecting specific information for a report (for example, reporting on all occurrences after a certain date)

To modify report output, proceed with the following steps:

1. Obtain a sample of the report to be modified.
2. Obtain a listing of the report module.
3. Plan the modifications.
   - For modifying D-, C-, and AREPORTs, see the Data Dictionary Network Diagram and CA IDMS Dictionary Structure Reference section to determine which records to select and which elements in those records are needed.
4. Code the modifications.
5. Obtain a sample of the modified report.

The rest of this section explains how to perform Steps 2 and 4.

Obtaining Report Module Listings

Contents

- Types of Listings (see page 884)

Types of Listings

The user can obtain a printed listing, an online listing, or a listing output to punched cards or a disk file.

Printed Listing
To obtain a printed listing, use Method 1 or Method 2 as follows:

- **Method 1** -- Include PARAM=LIST with the report-specific control statements submitted to run the report. PARAM=LIST lists the CA Culprit for CA IDMS parameters in the Sequential and Input Parameter Listings associated with report output.

- **Method 2** -- Run DREPORT 076, Module Key Report, using the DICTNAME and DBNAME options on the DATABASE parameter to name the dictionary that stores the report modules, and specifying the name of the report module in the KEY parameter. In the following example, DREPORT 008 is requested; the report module is stored in the CULPDICTION dictionary:

  ```
  DATABASE DICTNAME=CULPDICTION DBNAME=CULPDICTION
  DREPORT=076
  KEY MOD-NAME-067 'DREPORT 008'
  ```

**Online Listing**

To obtain an online listing, sign on to the Integrated Data Dictionary (IDD) and submit the DISPLAY MODULE statement to the DDDL compiler, as shown in the following example:

```
DISPLAY MODULE 'DREPORT 008' AS SYNTAX.
```

**Output to Cards or Disk**

To output the report module to punched cards or a disk file, run DREPORTs 051 or 052, respectively. An example appears as follows. For more information on submitting these reports, see Special-Purpose Report Modules (see page 740).

```
DATABASE DICTNAME=CULPDICTION
DREPORT=051
KEY MOD-NAME-067 'DREPORT 008'
```

---

**Coding in CA Culprit for CA IDMS**

As all CA IDMS report modules are coded using CA Culprit for CA IDMS parameters, some knowledge of CA Culprit for CA IDMS is required to modify the reports. This is a brief introduction to CA Culprit for CA IDMS. For more information on CA Culprit for CA IDMS, see the following manuals:

- **CA Culprit for CA IDMS Reference section**

- **CA Culprit for CA IDMS Using section**

- **CA Culprit for CA IDMS Coding Guidelines** (see page 886)

- **Types of CA Culprit for CA IDMS Parameters** (see page 886)
  - Parameters Within Each Category (see page 886)

- **Report-Specific and Global Parameters** (see page 887)

- **Edit Parameters** (see page 887)
  - Functions of Each Column (see page 888)

- **Summary of Parameter Types** (see page 888)
CA Culprit for CA IDMS Coding Guidelines

CA Culprit for CA IDMS parameters are coded in an 80-column format. These general rules apply to all CA Culprit for CA IDMS parameters:

- All parameters must be coded in uppercase letters.
- Column 1 is always blank, except on a continuation line or a USE parameter. An asterisk (*) in column 1 designates a continuation line.
- CA Culprit for CA IDMS parameters must not extend past column 72.
- Comments are introduced by a dollar ($) sign.
- The USE parameter cannot appear in any run that includes REPORT=, =MACRO, or =COPY parameters.

Types of CA Culprit for CA IDMS Parameters

There are five major categories of CA Culprit for CA IDMS parameters:

- Input definition parameters
- Output definition parameters
- Process definition parameters
- Work field definition parameters
- CA IDMS/DB retrieval parameters

Parameters Within Each Category

Each category has a function and includes several types of parameters, as follows:

- **Input definition parameters** define the source of input for the run:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPU</td>
<td>Defines the physical characteristics of the input file or identifies the subschema for the database to be accessed. A REPORT= request automatically generates an INPUT parameter. Runs that call report modules with the USE parameter must supply an INPUT parameter.</td>
</tr>
<tr>
<td>T</td>
<td>Defines the location, length, and data type of fields that appear in the CA Culprit for CA IDMS code. REC parameters are generated automatically for CA IDMS report modules.</td>
</tr>
<tr>
<td>SELEC</td>
<td>Defines criteria by which records are selected for processing.</td>
</tr>
<tr>
<td>T or BYPA</td>
<td></td>
</tr>
<tr>
<td>SS</td>
<td></td>
</tr>
</tbody>
</table>

- **Output definition parameters** define the format of the report output:
OU  Defines the physical characteristics of the output; for example, the number of characters per TPU line. By default, output is a printed report with 132 characters per line and 55 lines per page.

SOR Defines the order in which records are output.

3  Identifies a title parameter. A title parameter outputs a title at the top of each page.

4  Identifies a heading parameter. Headings are output under the title line, if any, at the top of each page.

5  Identifies a detail line.

6  Identifies a total line.

- Process definition parameters define report-specific process operations, such as arithmetic and data comparisons:

7  Identifies a process parameter applied to input data.

8  Identifies a process parameter applied to data in CA Culprit for CA IDMS's output phase of processing.

- Work field definition parameters define work fields required for report processing. Identifiers for work field parameters are 0, 1, and 15.

- CA IDMS/DB retrieval parameters define methods to access CA IDMS/DB database records:

  PAT  Identifies record retrieval routes through the database.

  KEY  Identifies a key field by which to access database records directly, rather than by an area sweep.

### Report-Specific and Global Parameters

Output and process definition parameters are report-specific. A 2-digit report number, starting in column 2, must precede each parameter identifier. Work field definition parameters are either report-specific or global; GW must appear in columns 2 through 3 for a global work field.

### Edit Parameters

Parameters 4, 5, and 6 are collectively known as edit parameters. Edit parameters define where a field is output in a report. The following figure illustrates the fixed format portion of an edit parameter.
Functions of Each Column

The columns in the figure above function as follows:

- **Columns 2 and 3** identify the 2-digit report number.

- **Column 4** identifies the type of edit line: a heading, detail, or total line.

- **Column 5** identifies a particular definition of an edit line. Each type of edit line can have up to eight different definitions. For example, the code shown in the following figure specifies six heading lines for DREPORT 028.

- **Columns 6 through 9** identify the location of the output field on the edit line. The location can be *absolute*, with a value in the range 0001 through 0132 for a printed report, or *relative*, with a value in the range *000 to *999.

- **Column 10** identifies spacing associated with the edit lines. Possible values for printed output are:
  - Blank -- no blank lines
  - 0 -- one blank line
  - Hyphen -- two blank lines
  - 1 -- page eject

Summary of Parameter Types

The following table summarizes the CA Culprit for CA IDMS parameters and indicates what each can do in modifying CA IDMS reports.

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Function</th>
<th>Modifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>Defines input</td>
<td></td>
</tr>
<tr>
<td>Identifier</td>
<td>Function</td>
<td>Modifications</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
<td>---------------</td>
</tr>
<tr>
<td>REC</td>
<td>Defines fields</td>
<td>Modify the name, start position, length, or data type of a field</td>
</tr>
<tr>
<td>SELECT or BYPASS</td>
<td>Selects records</td>
<td>Select only records that meet certain criteria</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Defines output</td>
<td>Modify the report's line size or number of lines per page</td>
</tr>
<tr>
<td>SORT</td>
<td>Sorts</td>
<td>Modify the order in which the report presents information</td>
</tr>
<tr>
<td>3</td>
<td>Outputs a title</td>
<td>Add/delete/change the report title</td>
</tr>
<tr>
<td>4</td>
<td>Outputs a heading</td>
<td>Add/delete/change report headings</td>
</tr>
<tr>
<td>5</td>
<td>Outputs a detail line</td>
<td>Add/delete/change columns of information</td>
</tr>
<tr>
<td>6</td>
<td>Outputs a total line</td>
<td>For summary reports, add/delete/change columns of information</td>
</tr>
<tr>
<td>7</td>
<td>Defines input processing</td>
<td>Modify or add processing applied to input data</td>
</tr>
<tr>
<td>8</td>
<td>Defines total processing</td>
<td>For summary reports, modify or add processing applied to total-time data</td>
</tr>
<tr>
<td>0, 1, or 15</td>
<td>Defines work fields</td>
<td>Add if required for additional processing</td>
</tr>
<tr>
<td>PATH</td>
<td>Accesses database records</td>
<td>Modify or add to access different database records</td>
</tr>
<tr>
<td>KEY</td>
<td>Accesses specified database records</td>
<td>Add to access particular database records</td>
</tr>
</tbody>
</table>

The following are the CA Culprit for CA IDMS parameters that produced the sample report:

**Sample CA Culprit for CA IDMS source code and output:**

```
PATHU1 00AK-012, S-010, SS-026, SSR-032, RCDACT-059, PROG-051
280 FUNC-TBL.13 'CONNECT' 'OBTAIN'
  * 'DISCONNECT' 'FIND'
  * 'STORE' 'ERASE'
  * 'CURRENCY ACCEPTED' 'MODIFY'
  * 'GET' 'BIND'
  * 'FIND KEEP' 'OBTAIN KEEP'
  * ' ' $ FUNCTIONS
280 KEY-TBL.12 7 43 11 3 12 2 15 8 5 14 23 63 $ KEY TO FUNCTION TBL
28OUTPUT D LP=51
28SORT S-NAME-010 SS-NAM-026 SSR-NAM-032
* PROG-NAM-051 PROG-VER-051 FUNC-TBL.DISPC
283 DATA DICTIONARY REPORTER
28410001 'DREPORT 028'
28410053 'IDMS RECORD ACTIVITY REPORT'
28420001 'STAR-HD'
28420045 'STAR-HD'
28420089 'STAR-HD'
28430102 '---- D A T E ----'
28440001 'SCHEMA'
28440010 'VER'
28440016 'SUBSCHEMA'
28440027 'RECORD'
28440060 'PROGRAM'
28440069 'VER'
28440075 'USAGE'
```
Sample report:

REPORT NO. 28                  DATA DICTIONARY REPORTER
DREPORT 028                  IDMS RECORD ACTIVITY REPORT

**********************************************************************************
********************************************
---- DATE ----
SCHEMA   VER   SUBSCHEMA  RECORD               PROGRAM  VER   USAGE
TIMES  COMPILED  CREATED
**********************************************************************************
********************************************

<table>
<thead>
<tr>
<th>SCHEMA</th>
<th>VER</th>
<th>SUBSCHEMA</th>
<th>RECORD</th>
<th>TIMES</th>
<th>COMPILED</th>
<th>CREATED</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPSCMH</td>
<td>100</td>
<td>EMPSS01</td>
<td>COVERAGE</td>
<td></td>
<td></td>
<td>EMPINQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 mm/dd/yy</td>
<td>mm/dd/yy</td>
<td></td>
</tr>
<tr>
<td>EMPSCMH</td>
<td>100</td>
<td>EMPSS01</td>
<td>DENTAL-</td>
<td></td>
<td></td>
<td>EMPINQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 mm/dd/yy</td>
<td>mm/dd/yy</td>
<td></td>
</tr>
<tr>
<td>EMPSCMH</td>
<td>100</td>
<td>EMPSS01</td>
<td>DEPARTMENT</td>
<td></td>
<td></td>
<td>EMPINQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 mm/dd/yy</td>
<td>mm/dd/yy</td>
<td></td>
</tr>
<tr>
<td>EMPSCMH</td>
<td>100</td>
<td>EMPSS01</td>
<td>DEPARTMENT</td>
<td>D3</td>
<td></td>
<td>EMPINQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 mm/dd/yy</td>
<td>mm/dd/yy</td>
<td></td>
</tr>
<tr>
<td>EMPSCMH</td>
<td>100</td>
<td>EMPSS01</td>
<td>DEPARTMENT</td>
<td>D4</td>
<td></td>
<td>EMPINQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 mm/dd/yy</td>
<td>mm/dd/yy</td>
<td></td>
</tr>
<tr>
<td>EMPSCMH</td>
<td>100</td>
<td>EMPSS01</td>
<td>DEPARTMENT</td>
<td>D6</td>
<td></td>
<td>EMPINQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 mm/dd/yy</td>
<td>mm/dd/yy</td>
<td></td>
</tr>
<tr>
<td>EMPSCMH</td>
<td>100</td>
<td>EMPSS01</td>
<td>DEPARTMENT</td>
<td>D3</td>
<td></td>
<td>EMPINQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 mm/dd/yy</td>
<td>mm/dd/yy</td>
<td></td>
</tr>
<tr>
<td>EMPSCMH</td>
<td>100</td>
<td>EMPSS01</td>
<td>DEPARTMENT</td>
<td>D4</td>
<td></td>
<td>EMPINQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 mm/dd/yy</td>
<td>mm/dd/yy</td>
<td></td>
</tr>
<tr>
<td>EMPSCMH</td>
<td>100</td>
<td>EMPSS01</td>
<td>DEPARTMENT</td>
<td>D3</td>
<td></td>
<td>EMPINQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 mm/dd/yy</td>
<td>mm/dd/yy</td>
<td></td>
</tr>
<tr>
<td>EMPSCMH</td>
<td>100</td>
<td>EMPSS01</td>
<td>DEPARTMENT</td>
<td>D4</td>
<td></td>
<td>EMPINQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 mm/dd/yy</td>
<td>mm/dd/yy</td>
<td></td>
</tr>
<tr>
<td>EMPSCMH</td>
<td>100</td>
<td>EMPSS01</td>
<td>DEPARTMENT</td>
<td>D4</td>
<td></td>
<td>EMPINQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 mm/dd/yy</td>
<td>mm/dd/yy</td>
<td></td>
</tr>
</tbody>
</table>
Temporary Modifications

Contents
- Copying the Report Module (see page 891)
- Deleting Columns (see page 892)
- Changing Headings (see page 893)
- Selecting Specific Data (see page 893)
- Specifying a New Sort Sequence (see page 894)
- Moving Columns (see page 894)

Runtime Modifications

When a report module is modified at run time, the modifications apply only to the run for which they are specified. You can make the following types of runtime modifications:

- Delete or change the parameters that define the report modules with a CA Culprit for CA IDMS USE parameter.
- Add new CA Culprit for CA IDMS parameters after the REPORT= request parameter or in conjunction with the USE parameter.

Copying the Report Module

USE Parameter

To make temporary changes or deletions to the code of a report module, a USE parameter is required. A USE parameter replaces the REPORT= parameter in the user-supplied code. Additionally, the following parameters must be supplied:

- An INPUT parameter that defines the subschema of the dictionary to be accessed (D-, C-, and AREPORTs only); INPUT parameters are always required for J- and SREPORTs.
- A report module that performs initial processing and field definitions (D- and JREPORTs only)

USE Parameter Clauses

USE parameter clauses perform several functions. The three most useful clauses are:

- DROP, which drops parameters
- KEEP, which retains parameters and drops any that are not specified
- CHANGE, which modifies parameters

Example
The following code copies in DREPORT modules 000 and 028. The INPUT parameter allocates a 10,000-byte buffer and accesses information defined by subschema IDMSNWKA.

DATABASE DICTNAME=CULPDICT DBNAME=DOCUDICT
INPUT 10000 DB SS=IDMSNWKA
USE ‘DREPORT 000’
USE ‘DREPORT 028’

Deleting Columns

DROP Clause of the USE Parameter

The following code deletes the SCHEMA, VER, and SUBSCHEMA columns shown in the sample report above. Each character string begins with 5, which designates an edit parameter for detail lines. The 1 that follows designates the edit line number. The remaining four numbers designate the absolute column position for each field.

DROP '510001' AND $DROP SCHEMA
'510009' AND $DROP VER
'510016' $DROP SUBSCHEMA

The following figure shows DREPORT 038 modified by this code:
Changing Headings

**CHANGE clause of the USE parameter**

In the following code, the headings for the columns just deleted are modified as follows:

- The first CHANGE clause modifies the literals SCHEMA, VER, and SUBSCHEMA to indicate a specific schema and subschema. Note that double quotation marks enclose the literals, which include single quotation marks.

- The second CHANGE clause changes the edit line number for the headings from 4 to 3.

```sql
CHANGE "SCHEMA" TO "EMPSCHM VERSION 100" AND "SUBSCHEMA" TO "SUBSCHEMA EMPSS01"
CHANGE '440001' TO '430001' AND '440016' TO '430023'
DROP '440010' $DROP SCHEMA VERSION HEADING
```

The following figure shows DREPORT 028 as it appears after these changes in the code.

**Selecting Specific Data**

To agree with the headings modified above, the code must select those records associated with subschema EMPSS01 and version 100 of EMPSCHM. The process parameter shown below specifies sequence number 030; CA Culprit for CA IDMS positions this parameter between existing process parameters with sequence numbers 020 and 100.

```sql
287030 IF S-NAM-010 NE 'EMPSCHM' AND * S-SER-010 NE 100 AND * SS-NAM-026 NE 'EMPSCHM' DROP
```
Specifying a New Sort Sequence

The following code replaces the existing SORT parameter with a new SORT parameter. The new SORT parameter orders the report contents by verb usage within program name.

```
DROP SORT
-
28SORT PROG-NAME-051 PROG-VER-051 FUNC-TBL.DISP STAR-HD
```

The following figure shows DREPORT 028 with a new sort sequence:

![Report Output](image)

Moving Columns

Use **CHANGE Clause of the USE Parameter**

Since the report contents are sorted by program name, the program name should appear in the left-most column. The following code rearranges the columns and column headings so that they appear as in the following report output.

```
- 
CHANGE '0060' TO '0001' AND $MOVE PROGRAM TO COLUMN 1
```
Permanent Modifications

Modify Dictionary Module

To permanently modify an existing module in the data dictionary, use one of the methods shown below:

Method 1

Use the DDDL compiler online in command mode:

- Display the module to be modified:
  
  DISPLAY MODULE NAME IS 'JREPORT 003' AS SYNTAX.

- Change the module name in the ADD MODULE statement; for example, JREPORT 103.
Use full screen editing to modify the report.

Press ENTER to store the new module in the dictionary.

**Method 2**

Use the DDDL compiler online in menu mode, as shown in the following series of screens:

```
CA
IDD REL nn.n
*** MASTER SELECTION ***
TOP

DICTIONARY NAME:DOCUDICT

USER NAME:ln
PASSWORD:**********

USAGE MODE:X UPDATE _ RETRIEVAL
PFKEY SIMULATION:X OFF _ ON

ATTR = ATTRIBUTE <PF2>
CLAS = CLASS <PF4>
ELEM = ELEMENT <PF6>
FILE = FILE <PF8>
MODU = MODULE <PF10>
ENTL = USER DEFINED ENTITY LIST
MSGS = MESSAGE
DISP = DISPLAY ALL

MODU

DISPLAY

MODULE NAME:jreport 403

ADD

VERSION NUMBER:1

DELETE

LANGUAGE:

DESCRIPTION:

SRCE = MODULE SOURCE <PF9>
MSYS = WITHIN SYSTEM
REGN = USER REGISTRATION
CLAT = CLASS/ATTRIBUTES <PF2>
COM = COMMENTS <PF6>
HIST = HISTORY <PF8>
XREF = CROSS REFERENCE <PF10>

COPY

SAME AS MODULE NAME:

VERSION NUMBER:

LANGUAGE:

MODULE 'JREPORT 403' VERSION 1
```
COPY FROM MODULE NAME: jreport 003
VERSION NUMBER: _HIGHEST_ LOWEST

LANGUAGE:  
  x SOURCE TEXT
  - MODULES
  - SYSTEMS
  - ATTRIBUTES
  - USERS
  - ALL COMMENT TYPES
  - COMMENTS
  - DEFINITION
  - USER DEFINED COMMENT (COMMENT KEY)
  - USER DEFINED NEST (RELATIONAL KEY)

MODULE 'JREPORT 403' VERSION 1 DISPLAYED

DISPLAY
x MODIFY
  - ADD
  - DELETE

LANGUAGE:  
TEXT:  
DESCRIPTION:  

x SRCE = MODULE SOURCE  <PF9>  _ MODX = MODULE EXTENSION  <PF11>
  _ MSYS = WITHIN SYSTEM
  _ REGN = USER REGISTRATION  <PF2>  _ PUBL = PUBLIC ACCESS  <PF3>
  _ CLAT = CLASS/ATTRIBUTES  <PF4>  _ RKEY = RELATIONAL KEYS  <PF5>
  _ COMM = COMMENTS  <PF6>  _ COML = COMMENT KEY LIST  <PF7>
  _ HIST = HISTORY  <PF8>  _ COPY = SAME AS/COPY FROM  
  _ XREF = CROSS REFERENCE  <PF10>  _ HELP = HELP  <PF1>

MODULE 'JREPORT 403' VERSION 1

---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----
03$00**** 'J' REPORTS IDMS JOURNAL FILE mm/dd/yy ROUTINE-JRPT003
030 WK-ABRT-FLAG '12345678901' $ FLAG ABNORMALLY ENDED RUN UNITS
030 WK-CNTR 1 $ AUTOMATIC COUNTER
030OUTPUT D
03SORT PROGRAM-NAME TRANSACT-ID DDR-HD
03410001 'REPORT NO. 03'
03410051 DDR-HD
03410106 DATE
03410122 'PAGE'
03410127 PAGE
03420001 'JREPORT 003'
03420055 'PROGRAM I/O STATISTICS'
034200460 '------------- PAGES -------------'
0351*010 PROGRAM-NAME HH 'PROGRAM'
0351*020 TRANSACT-ID FM 'ZZZZ9999' HH 'TRANSACTION ID'
0351*030 WK-ABRT-FLAG
0351*040 READ FM 'ZZZZZZZZZ' HH 'READ'
0351*050 WRITTEN FM 'ZZZZZZZZZ' HH 'WRITTEN'
0351*060 PAGE-REQUESTS FM 'ZZZZZZZZZ' HH 'REQUESTED'
0351*070 '  
037010 IF TYPE NE ('ENDJ' 'ABRT') DROP
037015 IF TYPE EQ 'ABRT' 030
037020 MOVE ' ' TO WK-ABRT-FLAG
037025 TAKE
Method 3

Copy the report module to cards using DREPORT 051, as follows:

REPORT=051
KEY MOD-NAME-067 'JREPORT 003'

Add, delete, or replace cards as required, using the modified cards as input to the DDDL compiler.

Method 4

Copy the report to a disk file using DREPORT 052, as follows:

REPORT=52
KEY MOD-NAME-067 'JREPORT 003'

Edit the file as necessary, using the edited module as input to the DDDL compiler.

Retrieve Module Using REPORT= or USE Parameters

Once the modified module is stored in the data dictionary, it can be retrieved by specifying its module number on a REPORT= parameter or a USE parameter. A good rule of thumb is to base the name of the modified module on the existing module name; for example, DREPORT xyy, where x is a digit in the range 1 through 9 and yy represents the last two digits of the original module number.