DB2 Enterprise Control Center for TME 10 V1.2

Monitoring Collection and Task Library Reference

SC26-9085-01

DB2 Enterprise Control Center for TME 10 V1.2

Monitoring Collection and Task Library Reference

SC26-9085-01

Note!
Before using this information and the product it supports, please read the general information under “Notices” on page xiii.

Second Edition (December 1997)

This edition replaces and makes obsolete the previous edition, SC26-9085-00. The technical changes for this edition are indicated by a vertical bar (|) to the left of a change.

This edition applies to V1.2 of DB2 Enterprise Control Center for TME 10, 5648-A46, and to any subsequent releases until otherwise indicated in new editions or technical newsletters. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

A form for readers’ comments is provided at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation, W92/H3
PO Box 49023
San Jose, CA
U.S.A.

Contents

Notices . . . xiii
Trademarks . . . xiii

About This Book . . . xv
Who Should Read This Book . . . xv
Prerequisite Books . . . xv
Terminology . . . xv
Typographic Conventions . . . xvi
Using This Book Online . . . xvii
Supported Versions of DB2 . . . xvii

Chapter 1. DB2 ECC Monitoring Collections . . . 1
Using the DB2 Monitoring Sources . . . 2
Adding a Monitor Using the GUI . . . 2
Adding a Monitor Using the CLI . . . 3
Using Monitoring Sources That Have Defaults . . . 3
Using Monitoring Sources That Return Numeric Values . . . 4
DB2 Agents and Applications Monitoring Collection . . . 4
conn: # of times agents are stolen . . . 5
conn: Application work load . . . 6
conn: Maximum associated agents . . . 7
conn: Total system CPU time used by agents . . . 7
conn: Total user CPU time used by agents . . . 8
db: Maximum agents associated with applications . . . 8
db: Maximum coordinating agents . . . 9
dbms: # of idle agents . . . 9
dbms: # of times agents are stolen . . . 9
dbms: Agent creation ratio . . . 9
dbms: Agents registered . . . 10
dbms: Agents waiting for a token . . . 10
dbms: Maximum agents registered . . . 11
dbms: Maximum agents waiting . . . 11
dbms: Percent agents waiting . . . 11
dbms: Percent total connections executing . . . 12
DB2 Basic Monitoring Collection . . . 13
DB2 ECC monitoring communication agent status . . . 15
DB2 instance status . . . 16
DB2 SNMP agent status . . . 16
User-defined OS command (numeric result) . . . 17
User-defined OS command (string result) . . . 17
conn: Percent locklist space used by application . . . 17
db: # of applications connected currently . . . 18
db: # of applications executing . . . 18
db: # of binds/precomps attempted . . . 19
db: # of catalog cache heap full . . . 19
db: # of catalog cache inserts . . . 19
db: # of catalog cache lookups . . . 20
db: # of catalog cache overflows . . . 20
db: # of commit statements attempted . . . 21
db: # of connects since 1st db connect . . . 21
db: # of deadlocks detected . . . 22
db: # of internal automatic rebinds . . . 22
db: # of internal commits . . . 22

db: # of internal rollbacks . . . 23
db: # of internal rows deleted . . . 23
db: # of internal rows inserted . . . 23
db: # of internal rows updated . . . 24
db: # of lock timeouts . . . 24
db: # of log pages read . . . 24
db: # of log pages written . . . 25
db: # of package cache inserts . . . 25
db: # of package cache lookups . . . 25
db: # of rollback statements attempted . . . 26
db: # of rows deleted . . . 26
db: # of rows inserted . . . 26
db: # of rows selected . . . 26
db: # of rows updated . . . 27
db: # of secondary logs allocated currently . . . 27
db: Avg lock escalations per connection . . . 27
db: Internal rollbk due to deadlock . . . 28
db: Last backup timestamp . . . 28
db: Log I/O . . . 29
db: Max # of concurrent connections . . . 29
db: Max database heap allocated . . . 29
db: Max primary log space used . . . 30
db: Max secondary logs space used . . . 30
db: Max total log space used . . . 30
db: Percent connections used . . . 31
db: Percent internal rollbacks due to internal deadlocks . . . 31
db: Percent locklist space used by database . . . 32
db: Percent package cache hit ratio . . . 32
db: Percent used in primary log . . . 33
db: Percent used in secondary log . . . 34
db: Primary log space allocated . . . 35
dbms: # of idle agents . . . 35
dbms: # of local connections . . . 35
dbms: # of local connections executing . . . 36
dbms: # of local databases with current connections . . . 36
dbms: # of piped sorts accepted . . . 37
dbms: # of piped sorts requested . . . 37
dbms: # of remote connections . . . 37
dbms: # of remote connections executing . . . 38
dbms: Committed private memory . . . 38
dbms: Last reset timestamp . . . 38
dbms: Percent piped sort hit ratio . . . 39
dbms: Start database manager timestamp . . . 40
dbpg: Data redistribution status . . . 40
dbpg: Data redistribution status for a nodegroup . . . 41
dbpg: DB2 local nodes status . . . 41
dbpg: FCM daemons status . . . 42
dbpg: Partitioned database data redistribution status . . . 42
dbpg: Percentage differential in row distribution of a table . . . 42
tbsp: Percent space used in DMS table space . . . 43
tbsp: Space used in SMS table space . . . 44
tbsp: Table space status . . . 45
DB2 Buffer Pool I/O Monitoring Collection . . . 47
conn: # of direct reads (4 KB pages) . . . 52
conn: # of direct writes (4 KB pages) . . . 52
conn: Average # of sectors read per direct read . . . 53

conn: Average # of sectors written per direct write 54
conn: Buffer pool time waited for prefetch (ms) 54
conn: Data page writes (4 KB pages) 54
conn: Index page reads (4 KB pages) 55
conn: Index page writes (4 KB pages) 55
conn: Logical data reads (4 KB pages) 56
conn: Percent buffer pool hit ratio, data+index 57
conn: Percent buffer pool hit ratio, index 57
conn: Percent package cache hit ratio 58
conn: Pool read time (s) 59
conn: Pool write time (s) 59
conn: Total pool I/O time (s) 60
db: # of cleaners due to dirty threshold 60
db: # of database files closed 61
db: # of direct read requests 61
db: # of direct reads from database 62
db: # of direct write requests 62
db: # of direct writes to database 62
db: # of log space cleaners invoked 63
db: # of page cleans in an interval 63
db: # of pages for each prefetch request 64
db: # of pool reads 64
db: # of pool writes 64
db: # of prefetch requests in an interval 65
db: # of synchronous I/O 65
db: # of synchronous index reads (4 KB pages) 65
db: # of synchronous index writes (4 KB pages) 66
db: # of synchronous reads 66
db: # of synchronous writes 66
db: # of victim page cleaners invoked 67
db: Average # of async reads per pool read 67
db: Average # of async writes per pool write 68
db: Average # of pool writes per pool read 68
db: Average # of sectors read per direct read 69
db: Average # of sectors written per direct write 69
db: Average direct read time (ms) 69
db: Average direct write time (ms) 70
db: Average pool I/O time (ms) 70
db: Average pool read time (ms) 70
db: Average pool write time (ms) 71
db: Average synchronous data read time (ms) 71
db: Average synchronous data write time (ms) 71
db: Average synchronous I/O (ms) 72
db: Avg # of pages for each cleaner 72
db: Buffer pool async data reads (4 KB pages) 72
db: Buffer pool async data writes 73
db: Buffer pool async index reads (4 KB pages) 73
db: Buffer pool async index writes 73
db: Buffer pool async read time (ms) 74
db: Buffer pool async write time (ms) 74
db: Buffer pool data logical reads 74
db: Buffer pool data pages copied from extended storage (4 KB pages) . . 75
db: Buffer pool data pages copied to extended storage (4 KB pages) . . . 75
db: Buffer pool data physical reads 75
db: Buffer pool data writes 76
db: Buffer pool index logical reads 76

db: Buffer pool index pages copied from extended storage (4 KB pages) . . 76
db: Buffer pool index pages copied to extended storage (4 KB pages) . . . 77
db: Buffer pool index physical reads 77
db: Buffer pool index writes 77
db: Buffer pool I/Os per second 78
db: Buffer pool time waited for prefetch (ms). 78
db: Direct read time (ms) 78
db: Direct write time (ms). 79
db: Extended storage read/write ratio 79
db: Percent buffer pool hit ratio, data+index 80
db: Percent buffer pool hit ratio, index 80
db: Percent catalog cache hit ratio 81
db: Synchronous read time (ms) 82
db: Synchronous write time (ms) 82
db: Total direct I/O time (ms) 82
db: Total pool physical I/O (ms) 83
db: Total pool physical read (ms) 83
db: Total pool physical write (ms) 83
db: Total synchronous I/O time (ms) 84
dbp: # of FCM nodes 84
dbp: Connection status 84
dbp: FCM buffers currently free 85
dbp: FCM connection entries currently free 85
dbp: FCM request blocks currently free 85
dbp: Minimum FCM buffers free 86
dbp: Minimum FCM connection entries free 86
dbp: Minimum FCM message anchors free 86
dbp: Minimum FCM request blocks free 87
dbp: Percent FCM buffers currently used 87
dbp: Percent FCM connection entries currently used. 88
dbp: Percent FCM request blocks currently used 88
dbp: Percent maximum FCM buffers used 89
dbp: Percent maximum FCM connection entries used 89
dbp: Percent maximum FCM message anchors used 90
dbp: Percent maximum FCM request blocks used. 91
dbp: Total buffers received 91
dbp: Total buffers sent 92
tbsp: # of direct read requests 92
tbsp: # of direct reads from database 93
tbsp: # of direct write requests 93
tbsp: # of direct writes to database 94
tbsp: # of pool reads 94
tbsp: # of pool writes 94
tbsp: # of synchronous index reads (4 KB pages) 95
tbsp: # of synchronous index writes (4 KB pages). 95
tbsp: # of synchronous reads 96
tbsp: # of synchronous writes 96
tbsp: Average # of sectors read per direct read. 97
tbsp: Average # of sectors written per direct write 97
tbsp: Average direct read time (ms) 97
tbsp: Average direct write time (ms) 98
tbsp: Average pool I/O time (ms) 98
tbsp: Average pool read time (ms) 98
tbsp: Average pool write time (ms) 99
tbsp: Average synchronous data read time (ms) 99
tbsp: Average synchronous data write time (ms) 100

tbsp: Average synchronous I/O time (ms) . . . 100
tbsp: Buffer pool async data reads . . . 100
tbsp: Buffer pool async data writes . . . 101
tbsp: Buffer pool async index reads (4 KB pages) . . . 101
tbsp: Buffer pool async index writes . . . 102
tbsp: Buffer pool async read time (ms) . . . 102
tbsp: Buffer pool async write time (ms) . . . 103
tbsp: Buffer pool data logical reads . . . 103
tbsp: Buffer pool data pages copied from extended storage (4 KB pages) . . . 104
tbsp: Buffer pool data pages copied to extended storage (4 KB pages) . . . 104
tbsp: Buffer pool data physical reads . . . 104
tbsp: Buffer pool data writes . . . 105
tbsp: Buffer pool I/Os per second . . . 105
tbsp: Buffer pool index logical reads . . . 106
tbsp: Buffer pool index pages copied from extended storage (4 KB pages) . . . 106
tbsp: Buffer pool index pages copied to extended storage (4 KB pages) . . . 106
tbsp: Buffer pool index physical reads . . . 107
tbsp: Buffer pool index writes . . . 107
tbsp: Direct read time . . . 108
tbsp: Direct write time (ms) . . . 108
tbsp: Extended storage read/write ratio . . . 108
tbsp: File closed . . . 109
tbsp: Percent buffer pool hit ratio, data+index . . . 109
tbsp: Percent buffer pool hit ratio, index . . . 110
tbsp: Percent prefetch satisfied . . . 110
tbsp: Percent total I/O . . . 111
tbsp: Prefetch requests in an interval . . . 111
tbsp: Synchronous read time (ms) . . . 112
tbsp: Synchronous write time (ms) . . . 112
tbsp: Total direct I/O time (ms) . . . 112
tbsp: Total pool physical I/O time (ms) . . . 113
tbsp: Total pool physical read time (ms) . . . 113
tbsp: Total pool physical write time (ms) . . . 113
tbsp: Total synchronous I/O . . . 114
tbsp: Total synchronous I/O time (ms) . . . 114
DB2 Configuration Monitoring Collection . . . 115
db: app_ctl_heap_sz . . . 116
db: applheapsz . . . 117
db: avg_appls . . . 117
db: buffpage . . . 117
db: catalogcache_sz . . . 118
db: chngpgs_thresh . . . 118
db: dbheap . . . 119
db: locklist . . . 119
db: logbufsz . . . 120
db: logprimary . . . 121
db: maxappls . . . 121
db: maxlocks . . . 122
db: mincommit . . . 123
db: newlogpath . . . 123
db: num_iocleaners . . . 124
db: num_ioservers . . . 125
db: pckcachesz . . . 126
db: Percent space used in database heap . . . 126
db: restore_pending . . . 127
db: seqdetect . . . 127

db: sortheap . . . 128
dbms: agentpri . . . 128
dbms: aslheapsz . . . 129
dbms: fcm_num_anchors . . . 129
dbms: fcm_num_buffers . . . 129
dbms: fcm_num_connect . . . 130
dbms: fcm_num_rqb . . . 130
dbms: max_agents . . . 131
dbms: max_coordagents . . . 131
dbms: maxcagents . . . 132
dbms: Maximum idle agents in the agent pool . . . 133
dbms: Minimum committed private memory . . . 133
dbms: mon_heap_sz . . . 133
dbms: num_poolagents . . . 134
dbms: Percent private memory used . . . 134
dbms: query_heap_sz . . . 135
dbms: rqrioblk . . . 135
dbms: sheapthres . . . 136
dbms: sqlstmtsz . . . 137
DB2 Data Replication Relational Monitoring Collection . . . 137
DataJoiner status . . . 138
Data replication Apply status . . . 139
SNA link status . . . 139
TCP/IP communication link status . . . 139
db: # of Apply lagging subscriptions . . . 140
db: Apply failed subscriptions . . . 140
db: Apply requires full refresh . . . 140
db: Apply subscriptions lag time . . . 141
db: Apply subscription status . . . 142
db: Capture errors . . . 142
db: Capture lag . . . 143
db: Capture pruning . . . 143
DB2 Locks and Deadlocks Monitoring Collection . . . 143
conn: Deadlocks found during last interval . . . 144
conn: Lock escalations during last interval . . . 145
conn: Lock wait time (s) . . . 146
conn: Lock waits . . . 146
conn: Locks held . . . 147
conn: UOW lock wait time (s) . . . 147
db: Applications in lock wait . . . 148
db: Average lock wait time (s) . . . 148
db: Average locks held per application . . . 148
db: Deadlocks found during last interval . . . 149
db: Lock escalations during last interval . . . 149
db: Lock list in use (bytes) . . . 150
db: Lock timeouts during last interval . . . 150
db: Lock waits . . . 150
db: Lock waits in an interval . . . 151
db: Locks held . . . 151
db: Percent application in lock wait . . . 151
db: Percent deadlock rollbacks . . . 152
db: Time waited on locks . . . 152
db: Total exclusive lock escalations . . . 153
DB2 Query Monitoring Collection . . . 153
db: # of database table spaces . . . 154
db: # of event monitors . . . 154

db: # of invalid triggers . . . 155
db: # of system table spaces . . . 155
db: # of tables . . . 156
db: # of table spaces . . . 156
db: # of table spaces with LONG data . . . 156
db: # of triggers . . . 156
db: # of user indexes . . . 157
db: # of views . . . 157
db: User-defined SQL (numeric result) . . . 157
db: User-defined SQL (string result) . . . 158
table: # of rows . . . 158
tbsp: # of tables . . . 158
user: # of tables . . . 159
DB2 SNMP Monitoring Collection . . . 159
DB2 database state . . . 161
DB2 server accumulated connections . . . 161
DB2 server current connections . . . 162
DB2 server finished transactions . . . 162
DB2 server status . . . 163
DB2 server version . . . 163
Maximum number of agents allowed to register at the same time . . . 164
Maximum number of agents registered at the same time . . . 164
Total Logical bytes read by DB2 server for all active databases . . . 165
Total logical pages read by DB2 server for all active databases . . . 165
Total physical bytes read by DB2 server . . . 166
Total physical bytes written by DB2 server for all active databases . . . 166
Total physical pages written by DB2 server for all active databases . . . 166
Total rows selected returned by DB2 server for all active databases . . . 167
Total SQL statements handled by DB2 server for all active databases . . . 167
Total SQL statements received by DB2 server for all active databases . . . 168
DB2 Sort Work Monitoring Collection . . . 168
conn: Application sorts . . . 169
conn: Sort time (s) . . . 170
conn: Statement sorts . . . 170
db: Active sorts . . . 171
db: Average sort time (ms) . . . 171
db: Percent sort overflowed . . . 172
db: Sort heap allocated (4 KB pages) . . . 172
db: Sort overflows . . . 173
db: Total sort time (ms) . . . 173
db: Total sorts . . . 173
dbms: Percent piped sorts rejected . . . 174
dbms: Percent sort heap allocated . . . 174
dbms: Post threshold sorts . . . 175
dbms: Sort heap allocated (4 KB pages) . . . 175
dbms: Total piped sorts rejected . . . 175
DB2 SQL Statement Activity Monitoring Collection . . . 176
conn: Application section inserts . . . 177
conn: Application section lookups . . . 178
conn: Binds/precompiles attempted . . . 179
conn: Commits . . . 179
conn: DDL SQL statements . . . 180
conn: Dynamic SQL statements . . . 180
conn: Explicit commits . . . 181
conn: Explicit rollbacks . . . 182
conn: Failed SQL statements . . . 182

conn: Open block cursors . . . 183
conn: Open cursors . . . 183
conn: Percent DDL SQL . . . 184
conn: Percent UID SQL . . . 184
conn: Rollbacks . . . 185
conn: Rows deleted . . . 186
conn: Rows inserted . . . 186
conn: Rows selected . . . 187
conn: Rows updated . . . 187
conn: Select SQL statements . . . 188
conn: SQL requests since last commit . . . 188
conn: Static SQL statements . . . 189
conn: Update/Insert/Delete SQL . . . 189
db: Committed statements per second . . . 190
db: DDL SQL statements . . . 190
db: Dynamic SQL statements . . . 190
db: Failed SQL statements . . . 190
db: Percent DDL SQL . . . 191
db: Percent failed SQL. . . . 191
db: Percent select SQL . . . 192
db: Percent UID SQL . . . 192
db: Rollbacks per second. . . . 192
db: Select SQL statements . . . 193
db: SQL statements per second . . . 193
db: Static SQL statements . . . 193
db: Update/Insert/Delete SQL . . . 193
DB2 Table Monitoring Collection . . . 194
db: Number of all invalid packages . . . 194
db: Number of invalid SYSTEM packages . . . 195
db: Number of invalid user packages . . . 196
table: Overflow accesses . . . 196
table: Reorg needed . . . 197
table: Rows read per second . . . 199
table: Rows written per second . . . 199

Chapter 2. DB2 ECC Task Libraries . . . 201
Running DB2 ECC Tasks . . . 201
Running DB2 ECC Tasks Directly. . . . 201
Running DB2 ECC Tasks from Monitors . . . 202
Monitor and Administration Tasks . . . 203
ECC_Broadcast_Message . . . 203
ECC_Reset_Counters . . . 204
ECC_Send_Notice . . . 204
ECC_Set_Up_Monitoring_Agent . . . 205
ECC_Start_DB2SNMP_Agent . . . 206
ECC_Start_Monitoring_Agent . . . 207
ECC_Stop_DB2SNMP_Agent . . . 207
ECC_Stop_Monitoring_Agent . . . 207
DB2 Database Tasks . . . 208
ECC_Activate_Database . . . 210
ECC_Alter_Bufferpool . . . 211
ECC_Backup_Database . . . 212
ECC_Create_Admin_Server. . . . 214
ECC_Create_Explain_Tables . . . 214
ECC_Create_Sample_Database . . . 215
ECC_Deactivate_Database . . . 216

ECC_Drop_Admin_Server . . . 217
ECC_Drop_Sample_Database . . . 217
ECC_Force_All_Applications . . . 217
ECC_Force_Applications . . . 218
ECC_Get_Admin_Configuration . . . 219
ECC_Get_Admin_Server_Instance . . . 219
ECC_Get_Database_Configuration . . . 220
ECC_Get_Database_Manager_Configuration . . . 221
ECC_Invoke_Stored_Procedure . . . 221
ECC_List_Applications . . . 222
ECC_List_Backup_Recovery_File . . . 223
ECC_List_DCS_Applications . . . 224
ECC_List_Node_Directory . . . 225
ECC_Prune_Recovery_History_File . . . 226
ECC_Quiesce_Tablespaces . . . 227
ECC_Rebind_All_Packages . . . 228
ECC_Rebind_Package . . . 229
ECC_Reorganize_Table . . . 230
ECC_Reorgchk . . . 231
ECC_Reset_Admin_Configuration . . . 232
ECC_Reset_Database_Configuration . . . 232
ECC_Reset_Database_Manager_Configuration . . . 233
ECC_Restart_Database . . . 234
ECC_Run_Statistics . . . 235
ECC_Set_Admin_Server_ID . . . 236
ECC_Start_Admin_Server . . . 237
ECC_Start_DB2 . . . 237
ECC_Start_DB2_NT_Security . . . 239
ECC_Stop_Admin_Server . . . 239
ECC_Stop_DB2 . . . 240
ECC_Stop_DB2_NT_Security . . . 241
ECC_Update_Admin_Configuration . . . 241
ECC_Update_Database_Configuration . . . 242
ECC_Update_Database_Manager_Configuration . . . 243
Data Replication Tasks . . . 244
ECC_Apply_Start . . . 245
ECC_Apply_Stop . . . 247
ECC_Capture_Get_Log_Seq . . . 249
ECC_Capture_Prune . . . 249
ECC_Capture_Reinit . . . 251
ECC_Capture_Resume . . . 252
ECC_Capture_Start . . . 253
ECC_Capture_Stop . . . 255
ECC_Capture_Suspend . . . 256

Appendix. DB2 Snapshot Application ID Formats . . . 259

Terms and Abbreviations . . . 261

Bibliography . . . 267
The DB2 Enterprise Control Center for TME 10 V1.2 Library . . . 267
Related IBM Publications . . . 267
DB2 for common servers Publications . . . 267
DB2 Universal Database Publications . . . 267
DDCS and DB2 Connect Publications . . . 267
DRDA Publications . . . 267

IBM Replication Publications 267
ITSO (Redbook) Publications 267
Tivoli Publications 267
Tivoli/Sentry and Distributed Monitoring Publications 267
TME 10 Versions 3.1 and 3.2 Publications 268

Index 269

Readers' Comments — We'd Like to Hear from You 279

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling (1) the exchange of information between independently created programs and other programs (including this one) and (2) the mutual use of the information that has been exchanged, should contact:

IBM Corporation
555 Bailey Avenue, W92/H3
PO Box 49023
San Jose, CA 95161–9023
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX
DATABASE 2
DataJoiner
DataPropagator
DB2
IBM
MVS/ESA
OS/2

The following terms are trademarks of Tivoli Systems, Inc., an IBM Company, in the United States or other countries or both:

Tivoli
Tivoli/Enterprise Console
Tivoli Management Environment
Tivoli Management Framework
Tivoli Management Platform
Tivoli/Sentry
TME
TME 10

Microsoft, Windows, Windows NT, and the Windows 95 logo are registered trademarks of Microsoft Corporation.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

About This Book

This book provides descriptions of the DB2 ECC monitoring collections and task library.

Who Should Read This Book

This book is written for system and database administrators who manage DB2 databases in a Tivoli Management Environment 10 (TME 10) environment.

Readers of this book should have knowledge of:

- The Windows NT, AIX, or Sun Solaris operating system
- The Tivoli Management Environment
- The DB2 relational database management system

Readers should also have knowledge of the setup of their current Tivoli Management Environment and DB2 installations.

Prerequisite Books

The *Tivoli/Sentry User's Guide* explains how to set up and use the Tivoli/Sentry application. You must be familiar with Tivoli/Sentry before you can install a monitoring collection.

The *DB2 Enterprise Control Center for TME 10 User's Guide* explains how to install, set up, and use the DB2 Enterprise Control Center for TME 10 application. You must install this application before you can use the monitoring collections and task libraries described in this book.

Terminology

Table 1 provides a brief list of common DB2 Enterprise Control Center for TME 10 (DB2 ECC) terms and shortened names used in this book.

You can find definitions of DB2 ECC terms in the text of this book. When the book introduces new terms, it italicizes and defines them. You can also find definitions in "Terms and Abbreviations" on page 261, which contains definitions of DB2 ECC terms and technical terms that pertain to DB2 ECC and Tivoli.

Table 1. DB2 Enterprise Control Center Basic Terminology List

Term	Definition
AIX	Advanced Interactive Executive
CLI	Command line interface
DB2	IBM DATABASE 2
DB2 ECC	DB2 Enterprise Control Center
DB2 Enterprise-Extended Edition	IBM DB2 Universal Database Enterprise-Extended Edition, Version 5

Table 1. DB2 Enterprise Control Center Basic Terminology List (continued)

Term	Definition
DB2 Enterprise-Extended Edition for AIX	IBM DB2 Universal Database Enterprise-Extended Edition for AIX, Version 5
DB2 for AIX	IBM DATABASE 2 for AIX, Version 2
DB2 for common servers	IBM DATABASE 2 for common servers, Version 2
DB2 for NT	IBM DATABASE 2 for Windows NT, Version 2
DB2 UDB	IBM DB2 Universal Database, Version 5
DB2 UDB for AIX	IBM DB2 Universal Database for AIX, Version 5
DB2 UDB for NT	IBM DB2 Universal Database for NT, Version 5
DB2 UDB for Solaris	IBM DB2 Universal Database for Solaris, Version 5
DPROPR	DataPropagator Relational
I/O	Input/output
NT	Microsoft Windows NT operating system
Sentry	Sentry or Distributed Monitoring
Solaris	Sun Microsystems Solaris operating system for UNIX workstations
TME	Tivoli Management Environment

Typographic Conventions

This book uses the following typographical conventions:

Bold

Used to indicate:

- AIX, NT, or Solaris commands and keywords
- Menus, push buttons, check boxes, list boxes, and other field names in the windows

Italics

Used to indicate:

- General emphasis
- The introduction of a new term
- Titles of books

`Monospace`

Used to indicate:

- Information that you are instructed to type at a command prompt or window
- Examples of specific data values
- Examples of text similar to what might be displayed by the system
- Examples of system messages

Using This Book Online

This book is available in PDF format on the installation media. You can display the PDF file, ECCM1200.PDF, with the Adobe Acrobat Reader 3.0.1 DB2 ECC provides the Adobe Acrobat Reader on the installation media in two formats: AIX and Windows NT. For instructions on how to install and use the Adobe Acrobat reader, see the *DB2 Enterprise Control Center for TME 10 User's Guide*.

The ECCM1200.PDF file can also be printed on any type of printer.

Supported Versions of DB2

Table 2 shows the versions and platforms of DB2 that DB2 ECC V1.2 supports.

Table 2. DB2 Versions and Platforms Supported by DB2 ECC V1.2

Version	Platform		
	AIX	**Solaris**	**Windows NT**
DB2 for common servers	✓		✓
DB2 Universal Database	✓	✓	✓
DB2 Enterprise-Extended Edition	✓		

Chapter 1. DB2 ECC Monitoring Collections

The DB2 ECC monitoring collections provide over 300 monitoring sources that help you use Tivoli/Sentry to manage distributed DB2 resources (instances and databases) effectively. Each collection is a group of monitoring sources that help you manage a different aspect of your DB2 installations. For example, the DB2 table monitoring collection contains monitoring sources that monitor table row activity. In general, the DB2 ECC monitoring collections are based on the various DB2 monitor groups (for information on the DB2 monitor groups, see the *DB2 System Monitor Guide and Reference* for the version of DB2 that you are using).

The eleven DB2 ECC monitoring collections are:

- DB2 agents and applications
- DB2 basic
- DB2 buffer pool I/O
- DB2 configuration
- DB2 data replication relational
- DB2 locks and deadlocks
- DB2 query
- DB2 SNMP
- DB2 sort work
- DB2 SQL statement activity
- DB2 table

This chapter describes each DB2 ECC monitoring collection and the monitoring sources in that collection. When a monitoring source name contains a prefix, the monitoring source returns information on a specific DB2 object (for example, a DB2 table space or a DB2 database). The prefix also indicates the type of *endpoint* for that monitoring source. An endpoint is a managed resource that is the target for monitoring. A *managed resource* is an entity that DB2 ECC manages and is represented by an icon on the TME 10 desktop. For more information about DB2 ECC endpoints and managed resources and monitoring them, see the *DB2 Enterprise Control Center for TME 10 User's Guide*.

These prefixes, the DB2 objects they refer to, and the type of endpoint for each prefix are listed in Table 3.

Table 3. Monitoring Sources Prefixes and Endpoints

Prefix	**DB2 Object**	***Monitoring Source Endpoints**					
		DB	DB Part.	DB Part. Grp.	Inst. (N)	Inst. (P)	Man. Node
conn:	Application	✓	✓				
db:	Database, nonpartitioned	✓	✓				
dbms:	Instance, nonpartitioned	✓	✓		✓		
dbp:	Database partition		✓				
dbpg:	Database partition group			✓			
tbsp:	Table space	✓	✓				

Table 3. Monitoring Sources Prefixes and Endpoints (continued)

Prefix	DB2 Object			*Monitoring Source Endpoints		
table:	Table	✓	✓			
user:	User	✓	✓			
None	Any					✓

*Monitoring Source Endpoints
DB= database, DB Part. = DB2 Partition, DB Part. Grp. = DB2 Partition Group, Inst. (N) = Instance (nonpartitioned database server), Inst. (P) = Instance (partitioned database server), Man. Node = Managed Node

A monitoring source without a prefix has a managed node endpoint.

You can use many of these monitoring sources to tune DB2 configuration parameters for the database manager and its databases. For detailed information on configuration parameters, see the *DB2 Administration Guide* for the version of DB2 that you are using.

Using the DB2 Monitoring Sources

You can use either the Tivoli graphical user interface (GUI) or the command line interface (CLI) to add or edit a monitoring source. To perform these actions, you need to know:

- The name of the profile in which the monitoring source is to reside. For information on profiles, see the *DB2 Enterprise Control Center for TME 10 User's Guide*.
- The monitoring collection name for the collection that contains the monitoring source that you want to use. For example, DB2_Stmt_Monitors is the name for the DB2 SQL statement activity monitoring collection.

  In this chapter, the name for each DB2 ECC monitoring collection is listed in the section that describes that collection. These names also appear in the list of collections in the Add Monitor to Tivoli/Sentry Profile window (see Figure 1 on page 3).
- The name of the monitoring source.

The following sections briefly explain how to use the monitoring sources in the DB2 ECC monitoring collections. For detailed information on using DB2 ECC monitoring sources, see the *DB2 Enterprise Control Center for TME 10 User's Guide*.

Adding a Monitor Using the GUI

In the Add Monitor to Tivoli/Sentry Profile window (shown in Figure 1 on page 3), the names of the DB2 ECC monitoring collections are listed in the **Monitoring Collections** box. If you select a collection name, the names of the monitoring sources in that collection are listed in the **Monitoring Sources** box.

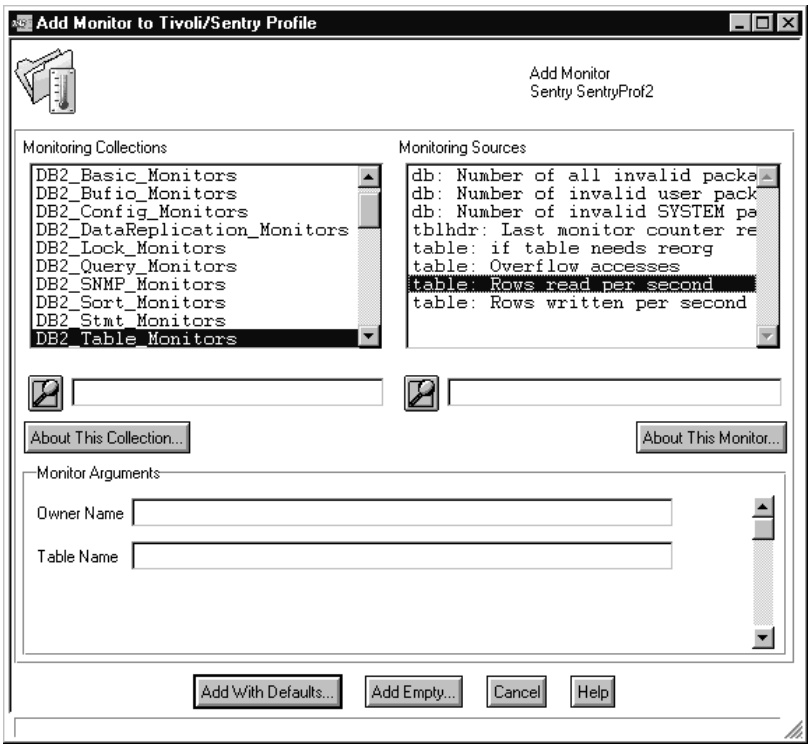

Figure 1. Add Monitor to Tivoli/Sentry Profile window

If a particular monitoring source requires one or more arguments, you must specify a value for each argument. You specify these values in the **Monitor Arguments** box of the Add Monitor to Tivoli/Sentry Profile window. In this chapter, the Argument section of a monitoring source description describes the arguments needed for that monitor, if any.

Adding a Monitor Using the CLI

You can add a monitor using the CLI with the Tivoli/Sentry **waddmon** command. To add a monitor with this command, use the CLI name for the monitor and specify values for any arguments the monitor requires.

For the syntax of the Tivoli/Sentry **waddmon** command, see the *Tivoli/Sentry User's Guide*.

Using Monitoring Sources That Have Defaults

Some DB2 ECC monitoring sources have default thresholds, actions, and monitoring schedules. In this book, these defaults are listed in the descriptions for those monitoring sources that have them. When adding a monitor for one of these monitoring sources, you can click on the **Add With Defaults** push button in the Add Monitor to Tivoli/Sentry Profile window to add the monitor with these default values.

For monitoring sources that have default thresholds, the following default actions are provided:

Send Tivoli notice
Send a notice to the DB2 Sentry notice group.

Change icon
Update the corresponding indicator collection.

None There is no default action for that threshold.

A threshold listed as N/A indicates there is no default threshold for that response level.

Some monitoring sources have only default monitoring schedules, and not default thresholds and actions. When you use these monitoring sources, you need to set the threshold values yourself, because the appropriate thresholds depend on your particular database environment (for example, some configuration parameters depend on whether the database is on AIX or NT). See the *DB2 Enterprise Control Center for TME 10 User's Guide* or the *Tivoli/Sentry User's Guide* for more information on adding monitors with default values.

Unless otherwise noted, the DB2 ECC monitoring sources that you can add with defaults have a default monitoring schedule to run the monitor every 60 minutes without any time restrictions.

Using Monitoring Sources That Return Numeric Values

If a DB2 ECC monitoring source that returns a numeric value does not receive valid data from DB2, the monitoring source returns one of two invalid data error numbers:

32767 If the monitoring source normally returns a positive number to represent valid data, it returns a value of 32767 for invalid data.

−32767
If the monitoring source normally returns 0 or a negative number to represent valid data, it returns a value of −32767 for invalid data.

For example, if the ″conn: Percent buffer pool hit ratio, data+index″ monitoring source does not receive a valid number from DB2 because no active DB2 application is running, the monitoring source returns 32767 to represent invalid data.

DB2 Agents and Applications Monitoring Collection

Collection name:
DB2_Agent_Monitors

The monitoring sources in this collection provide information about a database manager's agents and their related applications. An agent is a process or thread that carries out the requests made by a client application. Each connected application is served by exactly one coordinator agent and possibly a set of subordinator agents or subagents. Subagents are used for parallel SQL processing in partitioned databases and on SMP machines.

For each database transaction (unit of work) that occurs when the client is connected to a database, an agent requests permission from the database manager to process the transaction. The database manager grants permission by giving the agent a processing token. Only agents with a token are permitted to execute a unit of work against a database. The number of tokens available is controlled by the maximum number of concurrent agents (maxcagents) configuration parameter. If a token is not available, the agent waits until one is available.

DB2 Agents and Applications Monitoring Collection

For additional information on DB2 agents and applications, see the *DB2 System Monitor Guide and Reference* for the version of DB2 that you are using.

Table 4 lists the monitoring sources in this collection.

Table 4. Monitoring Sources in the DB2 Agents and Applications Collection

GUI Name	CLI Name	Returned Value Format
conn: # of times agents are stolen	connStolenAgents	Numeric
conn: Application work load	connApplWorkLoad	Numeric
conn: Maximum associated agents	connMaxAssoAgents	Numeric
conn: Total system CPU time used by agents	connSystemCPUTime	Numeric: seconds
conn: Total user CPU time used by agents	connUserCPUTime	Numeric: seconds
db: Maximum agents associated with applications	dbMaxAgentsAssoWithAppls	Numeric
db: Maximum coordinating agents	dbMaxCoordAgent	Numeric
dbms: # of idle agents	dbmsTotalIdleAgents	Numeric
dbms: # of times agents are stolen	dbmsStolenAgents	Numeric
dbms: Agent creation ratio	dbmsAgentCreationRatio	Numeric
dbms: Agents registered	dbmsAgentsReg	Ratio
dbms: Agents waiting for a token	dbmsAgentsWaiting	Numeric
dbms: Maximum agents registered	dbmsHwmAgentsReg	Numeric
dbms: Maximum agents waiting	dbmsHwmAgentsWaiting	Numeric
dbms: Percent agents waiting	dbmsPctAgentsWait	Percentage
dbms: Percent total connections executing	dbmsPctConnectionsExec	Percentage

conn: # of times agents are stolen

Description

Returns the number of times that agents are stolen from an application.

CLI **connStolenAgents** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

Use the returned value to adjust the num_poolagents configuration parameter (see “dbms: num_poolagents” on page 134). If the number of agents stolen from this application is high, the value of num_poolagents might be too low. When the agent pool size is too small, one application might fill the pool with associated subagents. When another application requires a new subagent and has no subagents in its associated agent pool, it steals subagents from the agent pools of other applications.

For information on related monitoring sources, see “conn: Application work load”.

conn: Application work load

Description

Returns the ratio of the maximum number of subagents associated with this application to the number of agents that are stolen from the application by DB2 to work on a different application.

CLI **connApplWorkLoad** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to evaluate the load that this application places on the system. An agent working for an application is associated with that application. After the agent completes the work for the application, it is placed in the agent pool as an idle agent, but it remains associated with the application. When the application requires an agent again, DB2 searches the agent pool for an agent already associated with the application and assigns work to the associated agent.

If another application needs an agent, DB2 attempts to satisfy the request in the following sequence:

1. Assigning an idle agent that is not associated with any application
2. Creating an agent if an idle agent is not available
3. Assigning an agent that is associated with another application

For example, if an agent cannot be created because the maximum number of agents (see “dbms: max_agents” on page 131) has been reached, DB2 assigns an idle agent associated with another application. This reassigned agent is called a stolen agent.

If the returned value is less than 1, you might want to redistribute the database to other nodes or increase the num_poolagents configuration parameter (see “dbms: num_poolagents” on page 134) to avoid the costs associated with frequently creating and terminating agents. If returned value is 32767 (no agents were stolen), it might suggest that the num_poolagents configuration is set too high, and that some of the agents in the pool are rarely used and are wasting system resources.

For information on related monitoring sources, see “conn: Maximum associated agents” and “conn: # of times agents are stolen” on page 5 .

conn: Maximum associated agents

Description

Returns the maximum number of subagents associated with this application.

CLI **connMaxAssoAgents** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

If the peak number of subagents is close to the num_poolagents configuration parameter (see “dbms: num_poolagents” on page 134), this might indicate that the workload for this DB2 node is too high.

conn: Total system CPU time used by agents

Description

Returns the total system CPU time used by this application’s database manager agents. System CPU time represents time spent in system calls.

CLI **connSystemCPUTime** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and

the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes
Use the returned value to identify applications or queries that consume large amounts of CPU.

conn: Total user CPU time used by agents

Description
Returns the total user CPU time used by this application's database manager agents. User CPU time represents time spent executing database manager code.

CLI **connUserCPUTime** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID
Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes
Use the returned value to identify applications or queries that consume large amounts of CPU.

db: Maximum agents associated with applications

Description
Returns the maximum number of agents associated with the applications connected to the database at one time.

CLI **dbMaxAgentsAssoWithAppls**

DB2 version
This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes
If the peak number of agents is close to the num_poolagents configuration parameter (see "dbms: num_poolagents" on page 134), this might indicate that the workload for this DB2 node is too high.

db: Maximum coordinating agents

Description
Returns the maximum number of coordinating agents connected to the database manager instance since the first connection was established.

CLI **dbMaxCoordAgent**

DB2 version
This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes
If the peak number of coordinating agents represents too high of a workload for this DB2 node, you can reduce the number that can concurrently execute a transaction by changing the maxcagents configuration parameter (see “dbms: maxcagents” on page 132).

dbms: # of idle agents

Description
Returns the number of agents in the agent pool that are unassigned to an application and idle within the database manager instance during the monitoring interval.

CLI **dbmsTotalIdleAgents**

DB2 version
This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes
Use this monitoring source to help set the num_poolagents configuration parameter (see “dbms: num_poolagents” on page 134). Having idle agents available to service requests for agents can improve performance.

dbms: # of times agents are stolen

Description
Returns the number of times that agents are stolen from all applications connected to the database within the database manager instance.

CLI **dbmsStolenAgents**

DB2 version
This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes
Agents are stolen when an idle agent associated with an application is reassigned to work on a different application.

dbms: Agent creation ratio

Description
Returns the number of times that agents are created when there are no more free agents available in the agent pools within the database manager instance.

CLI **dbmsAgentCreationRatio**

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

Use the returned value to adjust the num_poolagents configuration parameter (see “dbms: num_poolagents” on page 134). A low ratio suggests that the num_poolagents configuration parameter (see “dbms: num_poolagents” on page 134) is set too high, and that some of the agents in the pool are rarely used and are wasting system resources. A high ratio could indicate that the overall workload for this node is too high. You can adjust the workload by lowering the maximum number of coordination agents specified by the maxcagents configuration parameter (see “dbms: maxcagents” on page 132), or by redistributing data among the nodes.

For more information, see the *DB2 System Monitor Guide and Reference* and the *DB2 Administration Guide* for the version of DB2 that you are using.

Default monitoring schedule

Every 60 minutes without any time restrictions

dbms: Agents registered

Description

Returns the number of agents registered in the database manager instance.

CLI **dbmsAgentsReg**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The number of registered agents is used to calculate the value for the “dbms: Percent agents waiting” monitoring source (page 11).

The number of registered agents is always greater than or equal to the number of local databases with current connections (see “dbms: # of local databases with current connections” on page 36). Each application has a dedicated agent to process database requests within the database manager.

dbms: Agents waiting for a token

Description

Returns the number of agents that are waiting for a token so that they can execute a transaction in the database manager.

CLI **dbmsAgentsWaiting**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The number of agents waiting for a token is used to calculate the value for the “dbms: Percent agents waiting” monitoring source (page 11).

Each application has a dedicated agent to process database requests within the database manager. Each agent must get a token before it can execute a transaction. If all available tokens are being used, an agent waits until one is available.

dbms: Maximum agents registered

Description

Returns the highest number of agents that were registered at the same time by the database manager since the DB2 instance started.

CLI **dbmsHwmAgentsReg**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you evaluate your setting of the maxagents configuration parameter (see “dbms: max_agents” on page 131). The value of maxagents is the maximum number of database manager agents available at any given time to accept application requests. If the returned value from this monitoring source is near the value for maxagents, you might want to increase the value for maxagents.

dbms: Maximum agents waiting

Description

Returns the maximum number of agents that waited for a token at the same time since the database manager started.

CLI **dbmsHwmAgentsWaiting**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you evaluate your setting of the maxcagents configuration parameter. The value of maxcagents is the maximum number of database manager agents that can be concurrently executing a database manager transaction. By adjusting this parameter, you can control the throughput on the system during periods of high simultaneous application activity.

If the maxcagents parameter is set to its default value (−1), no agents should wait for a token and the value returned by this monitoring source should be zero.

dbms: Percent agents waiting

Description

Returns the percentage of registered agents that are waiting to execute a transaction.

CLI **dbmsPctAgentsWait**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

If the percentage is high, you can improve concurrency in the database manager by increasing the maxcagents configuration parameter.

Default thresholds and actions

Table 5. Default Thresholds and Actions for dbms: Percent agents waiting

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 90%	Send Tivoli notice Change icon
warning	Greater than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule

Every 60 minutes without any time restrictions.

dbms: Percent total connections executing

Description

Returns the percentage of the maximum number of applications allowed that are connected to a database and processing a unit of work (within the database manager instance) during the monitoring interval.

CLI **dbmsPctConnectionsExec**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

If the percentage is close to 100%, you might need to increase the maximum number of coordinating agents specified by “dbms: maxcagents” on page 132 or rebalance your workload.

Default thresholds and actions

Table 6. Default Thresholds and Actions for dbms: Percent of total connections executing

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 90%	Send Tivoli notice Change icon
warning	Greater than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

DB2 Basic Monitoring Collection

Collection name:
DB2_Basic_Monitors

The monitoring sources in this collection provide general information and statistics about:

- Status of the database manager and the DB2 ECC monitoring system
- Catalog caching
- Database activity
- Database connections
- Deadlocks and lock escalations
- Instance piped sorts
- Logging
- Package caching
- Row operations in databases

Some database system monitoring sources provide information about the *Fast Communication Manager* (FCM). In a partitioned database environment, most communication between database partitions is handled by the FCM, which provides internodal communications support. For more information on the FCM, see the *DB2 Administration Guide* for the version of DB2 that you are using.

Table 7 lists the monitoring sources in this collection.

Table 7. Monitoring Sources in the DB2 Basic Monitoring Collection

GUI Name	CLI Name	Returned Value Format
DB2 ECC monitoring communication agent status	DBTwoMonCommStatus	String: ″up″ or ″down″
DB2 instance status	DBTwoStatus	String: ″up″ or ″down″
DB2 SNMP agent status	DBTwoSnmpStatus	String: ″up″ or ″down″
User-defined OS command (numeric result)	DBTwoSystemCmd	Numeric
User-defined OS command (string result)	DBTwoSystemCmdS	String
conn: Percent locklist space used by application	connPctLocklistUsed	Percentage
db: # of applications connected currently	dbApplsCurCons	Numeric
db: # of applications executing	dbApplsInDbTwo	Numeric
db: # of binds/precomps attempted	dbBindsPrecompiles	Numeric
db: # of catalog cache heap full	dbCatCacheHeapFull	Numeric
db: # of catalog cache inserts	dbCatCacheInserts	Numeric
db: # of catalog cache lookups	dbCatCacheLookups	Numeric
db: # of catalog cache overflows	dbCatCacheOverflows	Numeric
db: # of commit statements attempted	dbCommitSqlStmts	Numeric
db: # of connects since 1st db connect	dbTotalCons	Numeric
db: # of deadlocks detected	dbDeadlocks	Numeric

DB2 Basic Monitoring Collection

Table 7. Monitoring Sources in the DB2 Basic Monitoring Collection (continued)

GUI Name	CLI Name	Returned Value Format
db: # of internal automatic rebinds	dbIntAutoRebinds	Numeric
db: # of internal commits	dbIntCommits	Numeric
db: # of internal rollbacks	dbIntRollbacks	Numeric
db: # of internal rows deleted	dbIntRowsDeleted	Numeric
db: # of internal rows inserted	dbIntRowsInserted	Numeric
db: # of internal rows updated	dbIntRowsUpdated	Numeric
db: # of lock timeouts	dbLockTimeouts	Numeric
db: # of log pages read	dbLogReads	Numeric: 4 KB pages
db: # of log pages written	dbLogWrites	Numeric: 4 KB pages
db: # of package cache inserts	dbPkgCacheInserts	Numeric
db: # of package cache lookups	dbPkgCacheLookups	Numeric
db: # of rollback statements attempted	dbRollbackSqlStmts	Numeric
db: # of rows deleted	dbRowsDeleted	Numeric
db: # of rows inserted	dbRowsInserted	Numeric
db: # of rows selected	dbRowsSelected	Numeric
db: # of rows updated	dbRowsUpdated	Numeric
db: # of secondary logs allocated currently	dbSecLogsAllocated	Numeric
db: Avg lock escalations per connection	dbAvgLockEscalPerConn	Numeric
db: Internal rollbk due to deadlock	dbIntDeadlockRollbacks	Numeric
db: Last backup timestamp	dbLastBackup	String: local timestamp
db: Log I/O	dbLogIo	Numeric
db: Max # of concurrent connections	dbConnectionsTop	Numeric
db: Max database heap allocated	dbDbHeapTop	Numeric: 4 KB pages
db: Max primary log space used	dbMaxPriLogSpaceUsed	Numeric: bytes
db: Max secondary logs space used	dbSecLogUsedTop	Numeric: bytes
db: Max total log space used	dbTotLogUsedTop	Numeric: bytes
db: Percent connections used	dbPctConnUsed	Percentage
db: Percent internal rollbacks due to internal deadlocks	dbPctIntRbPerIntDeadlk	Percentage
db: Percent locklist space used by database	dbPctLocklistUsed	Percentage
db: Percent package cache hit ratio	dbPkgHitRatio	Percentage
db: Percent used in primary log	dbPctUsedInPriLog	Percentage
db: Percent used in secondary log	dbPctUsedInSecLog	Percentage
db: Primary log space allocated	dbPriLogSpaceAlloc	Numeric: bytes
dbms: # of idle agents	dbmsIdleAgents	Numeric
dbms: # of local connections	dbmsLocalCons	Numeric
dbms: # of local connections executing	dbmsLocalConsInExec	Numeric

Table 7. Monitoring Sources in the DB2 Basic Monitoring Collection (continued)

GUI Name	CLI Name	Returned Value Format
dbms: # of local databases with current connections	dbmsConLocalDbases	Numeric
dbms: # of piped sorts accepted	dbmsPipedSortsAccepted	Numeric
dbms: # of piped sorts requested	dbmsPipedSortsRequested	Numeric
dbms: # of remote connections	dbmsRemConsIn	Numeric
dbms: # of remote connections executing	dbmsRemConsInExec	Numeric
dbms: Committed private memory	dbmsCommPrivateMem	Numeric
dbms: Last reset timestamp	dbmsLastReset	String: local timestamp
dbms: Percent piped sort hit ratio	dbmsPipedSortHitRatio	Percentage
dbms: Start database manager timestamp	dbmsDbTwostartTime	String: local timestamp
dbpg: Data redistribution status	dbpgNodegroupsRedistStatus	Numeric
dbpg: Data redistribution status for a nodegroup	dbpgNodegroupRedistStatus	Numeric
dbpg: DB2 local nodes status	dbpgNodeStatus	Numeric
dbpg: FCM daemons status	dbpgFcmStatus	Numeric
dbpg: Partitioned database data redistribution status	dbpgRedistStatus	Numeric
dbpg: Percentage differential in row distribution of a table	dbpgPercentDiffRowDistribution	Percentage
tbsp: Percent space used in DMS table space	tbspPctSpaceUsedDMS	Percentage
tbsp: Space used in SMS table space	tbspSpaceUsedSMS	Numeric: bytes
tbsp: Table space status	tbspStatus	Numeric

DB2 ECC monitoring communication agent status

Description

Checks whether the DB2 ECC monitoring communication agent is up or down.

CLI **DBTwoMonCommStatus**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is a string (″up″ or ″down″) that indicates whether the DB2 ECC monitoring communication agent is currently up or down. The DB2 ECC monitoring communication agent coordinates the communications between Sentry and DB2.

Default thresholds and actions

Table 8. Default Thresholds and Actions for DB2 ECC monitoring communication agent status

Response Level	Trigger When	Default Actions
critical	Never	None

DB2 Basic Monitoring Collection

Table 8. Default Thresholds and Actions for DB2 ECC monitoring communication agent status (continued)

Response Level	Trigger When	Default Actions
severe	Never	None
warning	Matches down	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule
Every 60 minutes without any time restrictions.

DB2 instance status

Description
Checks whether the DB2 instance is up or down.

CLI **DBTwoStatus** *DB2_instance_name*

Arguments

DB2_instance_name
Specify the name of the DB2 instance whose status you want.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is a string (″up″ or ″down″) that indicates whether the DB2 instance is currently up or down.

Default thresholds and actions

Table 9. Default Thresholds and Actions for DB2 instance status

Response Level	Trigger When	Default Actions
critical	Matches down	Send Tivoli notice Change icon
severe	Never	None
warning	Never	None
normal	N/A	None
always	N/A	None

Default monitoring schedule
Every 60 minutes without any time restrictions.

DB2 SNMP agent status

Description
Checks whether the DB2 SNMP agent is up or down.

CLI **DBTwoSnmpStatus**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is a string (″up″ or ″down″) that indicates whether the DB2 SNMP agent is currently up or down. The DB2 SNMP agent is a resident program that generates alerts to an SNMP manager and supplies information about the status of the DB2 server and databases (see “DB2 SNMP Monitoring Collection” on page 159 for more information).

User-defined OS command (numeric result)

Description

Runs a user-defined operating system command at the specified destination. You enter an operating system command and the path to where you want the command processed.

CLI **DBTwoSystemCmd** *OS_command*

Arguments

OS_command
Specify the operating system command or a shell script, with complete execution path. Use / as the path separator.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The user-defined command is expected to return a numeric value.

User-defined OS command (string result)

Description

Runs a user-defined operating system command at the specified destination. You enter an operating system command and the path to where you want the command processed.

CLI **DBTwoSystemCmdS** *OS_command*

Arguments

OS_command
Specify the operating system command or a shell script, with complete execution path. Use / as the path separator.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The user-defined command is expected to return a string value.

conn: Percent locklist space used by application

Description

Returns the percentage of space used in the lock list by the application.

CLI **connPctLocklistUsed** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID
Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine if this application is holding too much of the lock list. If an application holds too much of the lock list, other applications might experience lock escalations.

db: # of applications connected currently

Description
Returns the number of applications that are currently connected to the database.

CLI **dbApplsCurCons**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to evaluate the level of activity within the database and the amount of system resource being used.

If the returned value is always the same as the maxappls configuration parameter (meaning activity is high and you are at the maximum number of applications allowed), you might want to increase the value of maxappls to allow more applications to connect (if maxappls is less than the value of the maxagents configuration parameter).

If the returned value is always less than maxappls, application activity might be limited by the number of available database manager agents. In this case, you might be able to improve concurrency by increasing the value of the maxagents configuration parameter. You might also need to adjust the maxlocks or locklist configuration parameters.

db: # of applications executing

Description
Returns the number of currently-connected applications for which the database manager is currently processing a request.

CLI **dbApplsInDbTwo**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value indicates the number of database manager tokens that

are being used by applications connected to the database. See “dbms: Agents waiting for a token” on page 10 for monitoring the number of applications waiting for a token from the database manager.

db: # of binds/precomps attempted

Description

Returns the number of binds and precompiles that have been attempted in the database.

CLI **dbBindsPrecompiles**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

A high number of binds and precompiles indicates a high level of activity within the database.

This value does not include the number of internal automatic rebinds performed by the database, but it does include binds that occur as a result of the **REBIND PACKAGE** command.

db: # of catalog cache heap full

Description

Returns the number of times that an insert into the catalog cache failed due to a heap-full condition in the database heap.

CLI **dbCatCacheHeapFull**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

If the returned value is not zero, increase the database heap size or reduce the catalog cache size. You can increase the database heap size by increasing the dbheap configuration parameter. You can decrease the catalog cache size by decreasing the catalogcache_sz configuration parameter. For more information on these parameters, see “db: catalogcache_sz” on page 118, “db: dbheap” on page 119, or the *DB2 Administration Guide* for the version of DB2 that you are using .

The catalog cache draws its storage dynamically from the database heap. Even if the cache storage has not reached its limit, inserts into the catalog cache might fail due to a lack of space in the database heap.

Default monitoring schedule

Every 60 minutes without any time restrictions.

db: # of catalog cache inserts

Description

Returns the number of times that the system tried to insert table descriptor information into the catalog cache.

CLI **dbCatCacheInserts**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

If the returned value is low compared to the number of catalog cache lookups, the catalog cache is performing well. If this value is high compared to the number of catalog cache lookups, you might need to increase the catalog cache size (catalogcache_sz) configuration parameter (see “db: catalogcache_sz” on page 118 for more information).

A large value is to be expected immediately following the first connection to the database. In addition, heavy execution of DDL SQL statements involving a table, view, or alias can also increase this value.

This value also includes the number of times an insert failed due to catalog cache overflow and heap full conditions (see “db: # of catalog cache heap full” on page 19).

db: # of catalog cache lookups

Description

Returns the number of times that the catalog cache was referenced to obtain table descriptor information.

CLI **dbCatCacheLookups**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The catalog cache is referenced whenever a table, view, or alias name is processed during the compilation of an SQL statement. The returned value includes both successful and unsuccessful accesses to the catalog cache.

db: # of catalog cache overflows

Description

Returns the number of times that an insert into the catalog cache failed because the catalog cache was full.

CLI **dbCatCacheOverflows**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

If this value is large, the catalog cache might be too small for the workload. Enlarging the catalog cache can improve its performance (see “db: catalogcache_sz” on page 118 for more information). In addition, if the workload includes either:

- Transactions that compile many SQL statements that reference many tables, views, and aliases in a single unit of work
- Binding of packages that contain many SQL statements that reference many tables, views and aliases

you can improve the performance of the catalog cache by compiling fewer SQL statements in a single transaction (by issuing more commits or rollbacks) or by splitting packages so that they include fewer SQL statements.

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: # of commit statements attempted

Description
Returns the total number of SQL COMMIT statements that were attempted.

CLI **dbCommitSqlStmts**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine how the number of COMMIT statements changes over a period of time. A small rate of change can indicate that applications are not doing frequent commits. Infrequent commits can lead to problems with logging and data concurrency.

The returned value is also used in calculating the returned value for the “db: Committed statements per second” monitoring source (page 190).

db: # of connects since 1st db connect

Description
Returns the number of connections made to a database since the first database connection.

CLI **dbTotalCons**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to calculate the frequency at which applications connect to the database (over the monitoring period). If this frequency is low, you might want to perform one of the following actions:

- Use the ECC_Activate_Database task to activate the database before any applications connect to the database. Activating the database starts up all necessary database services, so that the database is available for connection and use by any application. See “ECC_Activate_Database” on page 210 for more information.
- Set up the first connecting application to issue the **ACTIVATE DATABASE** command before connecting to the database. For more information, see the **ACTIVATE DATABASE** command in the *DB2 Command Reference* for the version of DB2 that you are using.

Activating the database before connecting applications can avoid the extra overhead that the first connection sometimes incurs (for example, initial buffer pool allocation). Subsequent connections are then processed faster.

db: # of deadlocks detected

Description
Returns the total number of deadlocks that have occurred in the database.

CLI **dbDeadlocks**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
A high returned value indicates that applications are experiencing contention problems. These problems could be caused by the following situations:

- Lock escalations are occurring for the database.
- An application might be locking tables explicitly when system-generated row locks might be sufficient.
- An application might be using an inappropriate isolation level when binding.
- Catalog tables are locked for repeatable read.
- Applications are getting the same locks in different orders, resulting in deadlock.

Try to resolve the problem by determining in which applications (or application processes) the deadlocks are occurring. Then try to modify the application to better execute concurrently.

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: # of internal automatic rebinds

Description
Returns the number of automatic rebinds or recompiles that were attempted in the database.

CLI **dbIntAutoRebinds**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine the level of database activity. Automatic rebinds are the internal binds that the system performs when a package is invalidated. They can have a significant impact on performance and should be minimized where possible.

db: # of internal commits

Description
Returns the total number of commits initiated internally by the database.

CLI **dbIntCommits**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to gain insight into internal activity within the database. The returned value is also used in calculating the returned value for the “db: Committed statements per second” monitoring source (page 190).

db: # of internal rollbacks

Description

Returns the total number of rollbacks initiated internally by the database.

CLI **dbIntRollbacks**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to gain insight into internal activity within the database. The returned value is also used in calculating the returned value for the “db: Rollbacks per second” monitoring source (page 192).

db: # of internal rows deleted

Description

Returns the number of rows deleted from the database as a result of internal activity.

CLI **dbIntRowsDeleted**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to gain insight into internal activity within the database. If this activity is high, you might want to evaluate your table design to determine if the referential constraints or triggers that you defined on your database are necessary.

Internal row deletions can be a result of:

- Enforcement of an ON CASCADE DELETE referential constraint
- A trigger being fired

db: # of internal rows inserted

Description

Returns the number of rows inserted into a database as a result of internal activity caused by triggers.

CLI **dbIntRowsInserted**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to gain insight into internal activity within the database. If this activity is high, you might want to evaluate your design to determine if you can alter it to reduce this activity.

db: # of internal rows updated

Description
Returns the number of rows updated in the database as a result of internal activity.

CLI **dbIntRowsUpdated**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to gain insight into internal activity within the database. If this activity is high, you might want to evaluate your table design to determine if the referential constraints that you defined are necessary.

Internal update activity can be a result of:

- Enforcement of a referential constraint with the ON DELETE SET NULL rule
- A trigger being fired

db: # of lock timeouts

Description
Returns the number of times that a request to lock an object timed out instead of being granted for the database.

CLI **dbLockTimeouts**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to help you evaluate the setting for the locktimeout configuration parameter. If the returned value is high when compared to normal operating levels, you might have an application that is holding locks for long durations. In this case, use the monitoring sources described in “DB2 Locks and Deadlocks Monitoring Collection” on page 143 to determine whether an application has a problem.

If the returned value is low, the locktimeout parameter might be set too high. In this case, your applications might wait excessively to obtain a lock.

If the value of the locktimeout parameter equals −1, lock timeout detection is turned off. For online transaction processing (OLTP) environments, start with locktimeout = 30 seconds. For query-only environments, start with locktimeout greater than 30 seconds.

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: # of log pages read

Description
Returns the number of log pages read from disk by the logger.

CLI **dbLogReads**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value with values from an operating system monitoring source to quantify the amount of I/O on a device that is attributable to activity from the database. See “db: Log I/O” on page 29 for more information on log I/O.

db: # of log pages written

Description
Returns the number of log pages written to disk by the logger.

CLI **dbLogWrites**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value with values from an operating system monitoring source to quantify the amount of I/O on a device that is attributable to activity from the database. See “db: Log I/O” on page 29 for more information on log I/O.

db: # of package cache inserts

Description
Returns the number of times that a section had to be loaded into the package cache because an application found that the section it wanted was not in the package cache.

CLI **dbPkgCacheInserts**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
This value is used with the number of package cache lookups to calculate the package cache hit ratio (see “db: Percent package cache hit ratio” on page 32).

db: # of package cache lookups

Description
Returns the number of times that an application looked for a section in the package cache.

CLI **dbPkgCacheLookups**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
This value is used with the number of package cache lookups to calculate the package cache hit ratio (see “db: Percent package cache hit ratio” on page 32).

db: # of rollback statements attempted

Description
Returns the total number of SQL ROLLBACK statements that were attempted for the database.

CLI **dbRollbackSqlStmts**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine the amount of activity in the database and the amount of conflict between applications on the database. In general, you should try to minimize the number of rollbacks, because higher rollback activity results in lower throughput for the database.

The returned value is also used in calculating the returned value for the “db: Rollbacks per second” monitoring source (page 192).

db: # of rows deleted

Description
Returns the number of row deletions attempted for the database.

CLI **dbRowsDeleted**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to gain insight into the current level of activity within the database.

This count does not include the attempts counted in the “db: # of internal rows deleted” monitoring source (page 23).

db: # of rows inserted

Description
Returns the number of row insertions attempted for the database.

CLI **dbRowsInserted**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to gain insight into the current level of activity within the database.

db: # of rows selected

Description
Returns the number of rows that were selected and returned to an application.

CLI **dbRowsSelected**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to gain insight into the current level of activity within the database.

This value does not include a count of rows read for actions such as COUNT(*) or joins.

db: # of rows updated

Description
Returns the number of row updates attempted for the database.

CLI **dbRowsUpdated**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to gain insight into the current level of activity within the database.

This count does not include the attempts counted in the “db: # of internal rows updated” monitoring source (page 24). However, rows that are updated by more than one update statement are counted for each update.

db: # of secondary logs allocated currently

Description
Returns the total number of secondary log files that are currently being used for the database.

CLI **dbSecLogsAllocated**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is used in calculating the returned value for the “db: Percent used in secondary log” monitoring source (page 34).

This value is zero if the database does not have any secondary log files (if none are defined or if log retention is enabled).

db: Avg lock escalations per connection

Description
Returns the average number of lock escalations per connection.

CLI **dbAvgLockEscalPerConn**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
A high returned value indicates that the lock list space that is available to

applications keeps filling up. This situation results in escalations and leads to excessive lock waits for your applications. Check if any of these three conditions exist:

- The size of the maximum storage for lock lists (locklist) configuration parameter is too small, causing the lock list to fill up.
- The maxlocks configuration parameter is too small. In this case, the value for locklist is fine, but an application keeps reaching its maximum percent of the lock list, causing lock escalations.
- Applications are getting more locks than they really need. In this case, examine your applications.

A low returned value might indicate that the value for locklist is too high and memory is being wasted in the lock list.

db: Internal rollbk due to deadlock

Description

Returns the total number of forced rollbacks initiated by the database manager due to a deadlock. A rollback is performed on the current unit of work in an application selected by the database manager to resolve the deadlock.

CLI **dbIntDeadlockRollbacks**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value equals the number of deadlocks that the database manager needed to break, and can be an indication of concurrency problems. Internal rollbacks due to deadlocks lower the throughput of the database.

This value is included in the value returned by the “db: # of internal rollbacks” monitoring source (page 23).

Default monitoring schedule

Every 60 minutes without any time restrictions.

db: Last backup timestamp

Description

Returns a string that contains the date and time when the last database backup was completed for the database.

CLI **dbLastBackup**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to identify a database that was not recently backed up, or to identify which database backup file is the most recent.

The string returned gives the local date and time. For example, if the database is located in California and was last backed up on Wednesday, April 30, 1997 at 10:52 PM during daylight saving time, this monitoring source returns the following string:

```
Wed Apr 30 22:52:38 1997 PDT
```

If the database was never backed up, the returned string is ″None″.

db: Log I/O

Description

Returns the total amount of log I/O. This amount is the sum of the number of log pages read and the number of log pages written within the monitoring interval.

CLI **dbLogIo**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine whether you should move the log to a different device. If this I/O is beyond the capabilities of the current device, you can determine if moving the log (by changing the newlogpath configuration parameter), will improve performance.

db: Max # of concurrent connections

Description

Returns the highest number of simultaneous connections (both local and remote) made to a database since the first application connected to the database.

CLI **dbConnectionsTop**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

A returned value that equals the value of the maxappls configuration parameter indicates that some database connections were probably rejected, because maxappls limits the number of database connections allowed.

For more information about the maxappls parameter, see “db: maxappls” on page 121 or the *DB2 Administration Guide* for the version of DB2 that you are using.

db: Max database heap allocated

Description

Returns the largest amount of database heap (in 4 KB pages) allocated and used by the database since the first application connected to the database.

CLI **dbDbHeapTop**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to evaluate your setting of the dbheap configuration parameter. A returned value that equals the value of dbheap, indicates that

an application has probably received an error indicating that there was not enough storage available. The dbheap parameter limits the amount of storage that can be allocated for database heap. For more information about the dbheap parameter, see “db: dbheap” on page 119 or the *DB2 Administration Guide* for the version of DB2 that you are using.

db: Max primary log space used

Description

Returns the maximum number of bytes used by the primary log.

CLI **dbMaxPriLogSpaceUsed**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculating the returned value for the “db: Percent used in primary log” monitoring source (page 33).

This value is valid only if circular logging is used.

db: Max secondary logs space used

Description

Returns the maximum number of bytes used by the secondary log.

CLI **dbSecLogUsedTop**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculating the returned value for the “db: Percent used in secondary log” monitoring source (page 34).

This value is zero if the database does not have any secondary log files (if none are defined or if log retention is enabled).

db: Max total log space used

Description

Returns the maximum amount (in bytes) of total log space used.

CLI **dbTotLogUsedTop**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculating the returned value for the “db: Max primary log space used” monitoring source (page 33).

This value includes space used in both primary and secondary log files and is valid only if circular logging is used.

db: Percent connections used

Description

Returns the percentage of the allowed number of connections that are concurrent connections.

CLI **dbPctConnUsed**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you tune the maxappls configuration parameter. The maxappls configuration parameter limits the number of concurrent applications that can be connected (both local and remote) to a database. If the returned value is high, you might need to adjust the value of maxappls or schedule some of your applications so that they do not run concurrently.

Default thresholds and actions

Table 10. Default Thresholds and Actions for db: Percent connections used

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 80%	Send Tivoli notice Change icon
warning	Greater than 60%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule

Every 60 minutes without any time restrictions.

db: Percent internal rollbacks due to internal deadlocks

Description

Returns the percentage of the total number of internal rollbacks that were due to deadlocks.

CLI **dbPctIntRbPerIntDeadlk**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to distinguish those rollbacks caused by internal deadlocks from rollbacks caused by other situations (for example, incomplete imports). The returned value is the percentage of internal rollbacks due to internal deadlocks since the first database connection or the last reset of the database monitor counters.

Default thresholds and actions

Table 11. Default Thresholds and Actions for db: Percent internal rollbacks due to internal deadlocks

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 80%	Send Tivoli notice Change icon
warning	Greater than 70%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: Percent locklist space used by database

Description
Returns the percentage of space used in the database's lock list.

CLI **dbPctLocklistUsed**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine how much of the lock list space is free for new locks to be requested.

Default thresholds and actions

Table 12. Default Thresholds and Actions for db: Percent locklist space used by database

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 85%	Send Tivoli notice Change icon
warning	Greater than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: Percent package cache hit ratio

Description
Returns the package cache hit ratio (as a percent) for the last monitoring interval. The package cache hit ratio is the ratio of the difference between the package cache lookups and the package cache inserts to all package cache lookups.

CLI **dbPkgHitRatio**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This percentage indicates whether the package cache is being used effectively. If the returned value is high (greater than 80%), the package cache is performing well. A smaller percentage might indicate that the package cache should be increased. The size of the package cache is set by the pckcachesz configuration parameter (see “db: pckcachesz” on page 126 for more information).

To determine whether or not the execution of DDL statements is affecting the performance of the package cache, use the returned value from the “db: Percent DDL SQL” monitoring source (page 191). Execution of DDL statements can lower the percent package hit ratio (indicating poor performance), but in this case, increasing the size of the package cache will not necessarily improve overall performance. If DDL statements rarely occur, then performance might improve if you increase pckcachesz. If DDL statements frequently occur, then you might want to leave pckcachesz as it is and instead limit the use of DDL statements (for example, to certain time periods).

Use the “db: Dynamic SQL statements” and “db: Static SQL statements” monitoring sources (see pages 190 and 193) to provide information on the quantity and type of sections being cached. Caching of a section for static or dynamic SQL statements can improve performance, especially when the same statement is used multiple times by applications connected to the database.

Default thresholds and actions

Table 13. Default Thresholds and Actions for db: Percent package cache hit ratio

Response Level	Trigger When	Default Actions
critical	Less than 80%	Send Tivoli notice Change icon
severe	Less than 90%	Send Tivoli notice Change icon
warning	Less than 95%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule

Every 60 minutes without any time restrictions.

db: Percent used in primary log

Description

Returns the percentage of log space used by the primary log.

CLI **dbPctUsedInPriLog**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you evaluate the allocated amount of primary log space and tune the logbufsz, logfilsiz, and logprimary configuration parameters. See “db: logbufsz” on page 120 and “db: logprimary” on page 121 for more information.

The returned value is valid only if circular logging is used.

Default thresholds and actions

Table 14. Default Thresholds and Actions for db: Percent internal rollbacks due to internal deadlocks

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 85%	Send Tivoli notice Change icon
warning	Greater than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule

Every 60 minutes without any time restrictions.

db: Percent used in secondary log

Description

Returns the percentage of log space used by the secondary log.

CLI **dbPctUsedInSecLog**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to show the current dependency on secondary logs. Secondary logs are used when you have circular logging (log retention off) and the primary log files are full.

If the returned value is high, you might need one or more of the following things:

- Larger log files
- More primary log files
- More frequent COMMIT statements within your applications

For information on how to get larger log files and more primary log files, see the *DB2 Administration Guide* for the version of DB2 that you are using.

The returned value is zero if the database does not have any secondary log files (if none are defined or if log retention is enabled).

Default thresholds and actions

Table 15. Default Thresholds and Actions for db: Percent internal rollbacks due to internal deadlocks

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 85%	Send Tivoli notice Change icon
warning	Greater than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: Primary log space allocated

Description
Returns the number of bytes allocated to the primary log.

CLI **dbPriLogSpaceAlloc**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is used in calculating the returned value for the “db: Percent used in primary log” monitoring source (page 33).

The allocated amount (in bytes) of primary log space is determined by the following formula: logprimary × logfilsiz × 4096

dbms: # of idle agents

Description
Returns the number of agents in the agent pool that are currently unassigned to an application.

CLI **dbmsIdleAgents**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to help set the maximum number of idle agents (max_idleagents) configuration parameter. Having idle agents available to service requests for agents can improve performance.

dbms: # of local connections

Description
Returns the number of local applications that are currently connected to a local database within the DB2 instance.

CLI **dbmsLocalCons**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine the level of concurrent processing occurring in the DB2 instance. This value changes frequently, so you might need to sample it at specific intervals over an extended period of time to get a realistic view of system usage.

This number includes only applications that are initiated from the same node where the DB2 instance is located. The applications are connected but might not be executing a unit of work.

Use this value with the “dbms: # of remote connections” monitoring source (page 37) to help you evaluate the setting of the maxagents configuration parameter.

dbms: # of local connections executing

Description

Returns the number of local applications that are currently connected to a local database within the DB2 instance and that are currently processing a unit of work.

CLI **dbmsLocalConsInExec**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine the level of concurrent processing occurring in the DB2 instance. This value changes frequently, so you might need to sample it at specific intervals over an extended period of time to get a realistic view of system usage.

This number includes only applications that are initiated from the same node where the DB2 instance is located.

Use this value with the “dbms: # of remote connections executing” monitoring source (page 38) to help you evaluate the setting of the maxcagents configuration parameter.

dbms: # of local databases with current connections

Description

Returns the number of local databases that have applications connected.

CLI **dbmsConLocalDbases**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to get an indication of how many database information records you can expect when gathering monitoring values at the database level.

dbms: # of piped sorts accepted

Description
Returns the number of piped sorts that were accepted.

CLI **dbmsPipedSortsAccepted**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to help you improve sort performance. Every active sort on the system allocates memory, which can result in sorting taking up too much of the available system memory. Piped sorts help reduce disk I/O, so increasing the number of accepted piped sorts can improve performance.

When the returned value is low compared to the number of piped sorts requested, you can improve performance by adjusting either the sort list heap (sortheap) or sort heap threshold (sheapthres) configuration parameter, or by adjusting both. See “dbms: Percent piped sort hit ratio” on page 39 for the implications of adjusting these parameters.

dbms: # of piped sorts requested

Description
Returns the number of piped sorts that were requested.

CLI **dbmsPipedSortsRequested**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to help you improve sort performance. Every active sort on the system allocates memory, which can result in sorting taking up too much of the available system memory. Piped sorts help reduce disk I/O, so increasing the number of accepted piped sorts can improve performance. See “dbms: Percent piped sort hit ratio” on page 39 for a discussion of increasing the number of accepted piped sorts to improve performance.

dbms: # of remote connections

Description
Returns the number of current connections that remote clients initiated to the DB2 instance.

CLI **dbmsRemConsIn**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value shows the level of activity between this DB2 instance and other DB2 instances.

Use this value with the “dbms: # of local connections” monitoring source (page 35) to help you evaluate the setting of the maxagents configuration parameter.

dbms: # of remote connections executing

Description

Returns the number of remote applications that are currently connected to a database and are currently processing a unit of work within the DB2 instance.

CLI **dbmsRemConsInExec**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine the level of concurrent processing occurring in the DB2 instance. This value will change frequently, so you might need to sample it at specific intervals over an extended period of time to get a realistic view of system usage.

Use this value with the “dbms: # of local connections executing” monitoring source (page 36) to help you evaluate the setting of the maxcagents configuration parameter.

dbms: Committed private memory

Description

Returns the amount of private memory that the DB2 instance has currently committed.

CLI **dbmsCommPrivateMem**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help set the min_priv_mem configuration parameter. This parameter specifies the number of pages that the database server process will reserve as private virtual memory when a DB2 instance is started. If the returned value is close to the current min_priv_mem value, you might want to increase the value for min_priv_mem to ensure that you have enough private memory available.

dbms: Last reset timestamp

Description

Returns a string that contains the date and time that the monitoring source counters were reset for the database manager.

CLI **dbmsLastReset**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The monitoring source counters are reset using the ECC_Reset_Counters

task. Use the returned string to identify when the database system monitor counters were last reset. This timestamp determines the scope of information that's based on total counts (for example, the number of registered agents).

The string returned gives the local date and time. For example, if the database manager is located in California and the monitor counters were last reset on Wednesday, April 30, 1997 at 10:52 PM during daylight saving time, this monitoring source returns the following string:

```
Wed Apr 30 22:52:38 1997 PDT
```

If the counters were never reset, the returned string is ″None″.

See "ECC_Reset_Counters" on page 204 for more information on how to reset the counters.

dbms: Percent piped sort hit ratio

Description

Returns the piped sort hit ratio (as a percent) for the last monitoring interval. The piped sort hit ratio is the ratio of piped sorts accepted to piped sorts requested.

CLI **dbmsPipedSortHitRatio**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you improve sort performance. Every active sort on the system allocates memory, which can result in sorting taking up too much of the available system memory. Piped sorts help reduce disk I/O, so increasing the number of accepted piped sorts can improve performance.

When the returned value is low, you can improve performance by adjusting either the sort heap size (sortheap) or sort heap threshold (sheapthres) configuration parameter, or by adjusting both. If piped sorts are being rejected (resulting in a low percentage piped sort hit ratio), consider decreasing your sort heap or increasing your sort heap threshold. A piped sort is not accepted if the sort heap threshold will be exceeded when the sort heap is allocated for that piped sort.

Be aware that adjusting sortheap or sheapthres might have the following consequences:

- Increasing sheapthres might result in more memory allocated for sorting, causing the paging of memory to disk and slowing overall system performance.
- Decreasing sortheap might result in the need for extra sort merge phases that could slow down sorts.

Default thresholds and actions

Table 16. Default Thresholds and Actions for dbms: Percent piped sort hit ratio

Response Level	Trigger When	Default Actions
critical	Less than 80%	Send Tivoli notice Change icon
severe	Less than 90%	Send Tivoli notice Change icon
warning	Less than 95%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule
Every 60 minutes without any time restrictions.

dbms: Start database manager timestamp

Description
Returns a string that contains the date and time that the database manager was started.

CLI **dbmsDbTwostartTime**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned string with the value for the current time to calculate the elapsed time since the database manager was started.

The string returned gives the local date and time. For example, if the database manager is located in California and it was started on Wednesday, April 30, 1997 at 10:52 PM during daylight saving time, this monitoring source returns the following string:

```
Wed Apr 30 22:52:38 1997 PDT
```

If the database manager is not started, the returned string is ″None″.

dbpg: Data redistribution status

Description
Returns a positive value if row redistribution is in progress in one or more nodegroups within any of the partitions in the database partition group during the monitoring interval.

CLI **dbpgNodegroupsRedistStatus**

DB2 version
This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes
If any of the partitions in the database partition group is participating in more than one nodegroup, all the nodegroups in which the partition is a member will be inspected for data redistribution status. The monitoring source also returns a positive value if a failure occurs during data redistribution on any of the nodegroups and user intervention is required to recover redistribution errors.

To determine which nodegroups are currently redistributing data (or require user intervention), issue the following query:

```
SELECT NGNAME, REBALANCE_PMAP_ID FROM SYSCAT.NODEGROUPS WHERE
REBALANCE_PMAP_ID <> -1
```

dbpg: Data redistribution status for a nodegroup

Description

Returns a positive value if row redistribution is in progress in the specified nodegroup in the partitioned database during the monitoring interval.

CLI **dbpgNodegroupRedistStatus** *DB2_node_group_name*

Arguments

DB2_node_group_name

Specify the name of the DB2 node group that you want to monitor for data redistribution.

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

A positive value returned by this monitoring source indicates that the specified nodegroup in the partitioned database is redistributing data. Use the Data Redistribution utility to redistribute data across database partitions. If a failure occurs during redistribution of data, some tables might be redistributed while others are not. The monitoring source also returns a positive value if a failure occurs during data redistribution and user intervention is required to recover from redistribution errors.

For more information on redistributing data across database partitions, see the *DB2 Administration Guide* for the version of DB2 that you are using.

dbpg: DB2 local nodes status

Description

Returns zero if all of the nodes defined in the partition group are active. Otherwise, the value returned represents the number of nodes which are not active.

CLI **dbpgNodeStatus**

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

Depending on the actual partitions defined in the database partition group, it is not necessarily true that all the nodes defined in the parallel environment will be examined. To ensure that all the nodes in the partitioned environment are examined, define a partition group containing at least one database partition from each of the nodes in the partitioned environment.

dbpg: FCM daemons status

Description

Returns zero if all the FCM daemons on all the nodes in which any partition in the partition group is defined are active. Otherwise, the value returned represents the number of nodes without an active FCM daemon.

CLI **dbpgFcmStatus**

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

For each partition in a partition group, the node in which the partition is defined is examined to determine if an FCM daemon is servicing that partition. This monitoring source returns zero if all the FCM daemons that are examined are active. Otherwise, the value returned represents the number of nodes without an active FCM daemon.

Depending on the actual partitions defined in the database partition group, it is not necessarily true that all the nodes defined in the parallel environment will be examined. To ensure that all the nodes in the partitioned environment are examined, define a partition group containing at least one database partition from each of the nodes in the partitioned environment.

dbpg: Partitioned database data redistribution status

Description

Returns a positive value if row redistribution is in progress in any nodegroup in the partitioned database during the monitoring interval.

CLI **dbpgRedistStatus**

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

A positive value returned by this monitoring source indicates the number of nodegroups in the partitioned database that are actively redistributing data. Use the Data Redistribution utility to redistribute data across database partitions. If a failure occurs during redistribution, some tables might be redistributed while others are not. The monitoring source also returns a positive value if a failure occurs during data redistribution and user intervention is required to recover from redistribution errors.

To determine which nodegroups are redistributing data (or require intervention from redistribution errors) issue the following query:

```
SELECT NGNAME, REBALANCE_PMAP_ID FROM SYSCAT.NODEGROUPS WHERE
REBALANCE_PMAP_ID <> -1
```

dbpg: Percentage differential in row distribution of a table

Description

Returns the difference in percentage between the highest and the lowest rows distributed across the partitions in the selected partition group during the monitoring interval.

CLI **dbpgPercentDiffRowDistribution** *Owner_name Table_name Partition_key_column*

Arguments

Owner_name
Specify the owner of the table.

Table_name
Specify the name of the table.

Partition_key_column
Specify the name of one of the columns as defined in the table's partition key.

DB2 version
This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes
The partitions defined in the partition group can be a subset of the partitions across which the table rows are distributed. However, by defining the complete set of partitions to the partition group, the result returned by this monitoring source will present a more accurate picture of the distribution of the rows in the table.

In computing the percentage of rows being distributed to a partition in the partition group, the total number of rows distributed to all the partitions defined in the partition group is used as the base. This number might be smaller than the total number of the rows in the table.

For example, assume the rows of a table are distributed across four separate nodes (N1, N2, N3 and N4) and the partitions defined in the partition group are just N1, N2 and N3. Table 17 shows the actual number of rows distributed to each node with the computed percentage of rows distributed.

Table 17. Table rows distributed across four nodes

Nodes	N1	N2	N3	N4
Rows Distributed	30	10	140	20
Percentage of Rows Distributed	17%	6%	77%	N/A

Using the example in Table 17, the monitoring source returns a value of 71 (which represents the difference between 77% and 6%). This value indicates that the distribution of the table rows across the partitions in the partition group is highly skewed.

tbsp: Percent space used in DMS table space

Description
Returns the percentage of space used in the Database Managed Space (DMS) table space.

CLI **tbspPctSpaceUsedDMS** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 DMS table space.

DB2 Basic Monitoring Collection

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine whether the table space needs more space.

Default thresholds and actions

Table 18. Default Thresholds and Actions for tbsp: Percent space used in DMS table space

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 85%	Send Tivoli notice Change icon
warning	Greater than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule

Every 60 minutes without any time restrictions.

tbsp: Space used in SMS table space

Description

Returns the number of bytes allocated to the System Managed Space (SMS) table space.

CLI **tbspSpaceUsedSMS** *DB2_table_space_name*

Arguments

DB2_table_space_name

Specify the name of the DB2 SMS table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine whether the table space needs more space.

The returned value can be composed of the hexadecimal sum of various table space states. For example, the returned value could be:

```
0x000c = 0x0004 + 0x008
```

You can use the db2tbst tool to reverse the calculated returned value.

The sum of the table space status might contain one or more of the following values:

Value	Table space state
0	Normal
1	Quiesced: SHARE

2 Quiesced: UPDATE

4 Quiesced: EXCLUSIVE

8 Load pending

10 Delete pending

20 Backup pending

40 Roll forward in progress

80 Roll forward pending

100 Restore pending

200 Disable pending

400 Reorganization in progress

800 Backup in progress

1000 Storage must be defined

2000 Restore in progress

2000000
Storage may be defined

4000000
Storage definition is in 'final' state

8000000
Storage definition was changed prior to roll forward

10000000
DMS rebalancer is active

20000000
Table space deletion in progress

40000000
Table space creation in progress

Default monitoring schedule
Every 24 hours without any time restrictions.

tbsp: Table space status

Description
Returns a numeric value that indicates the status of the table space.

CLI **tbspStatus** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine the state of the table space. The returned value is one of the following values:

Value	Table space state

0x0 Normal

0x1 Quiesced: SHARE

0x2 Quiesced: UPDATE

0x4 Quiesced: EXCLUSIVE

0x8 Load pending

0x10 Delete pending

0x20 Backup pending

0x40 Roll forward in progress

0x80 Roll forward pending

0x100 Restore pending

0x200 Disable pending

0x400 Reorganization in progress

0x800 Backup in progress

0x1000
Storage must be defined

0x2000
Restore in progress

0x2000000
Storage may be defined

0x4000000
Storage definition is in 'final' state

0x8000000
Storage definition was changed prior to roll forward

0x10000000
DMS rebalancer is active

0x20000000
Table space deletion in progress

0x40000000
Table space creation in progress

Default thresholds and actions

Table 19. Default Thresholds and Actions for tbsp: Table space status

Response Level	Trigger When	Default Actions
critical	N/A	None
severe	N/A	None
warning	Greater than 0	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule
Every 60 minutes without any time restrictions.

For more information about table space states, see the *DB2 Administration Guide* for the version of DB2 that you are using.

DB2 Buffer Pool I/O Monitoring Collection

Collection name:
DB2_Bufio_Monitors

The monitoring sources in this collection provide statistics about the database buffer pool. The buffer pool is an area of storage into which database pages (that contain table rows or index entries) are read and changed. The purpose of the buffer pool is to improve database system performance. Data can be accessed much faster from memory than from a disk. Therefore, the fewer times the database manager needs to read from or write to a disk, the better the performance.

The configuration of the buffer pool is the most important area that you can tune, because most of the data manipulation for applications connected to the database takes place there.

Monitoring sources with a prefix of dbp can only be run on partition endpoints. They cannot run on partition group endpoints.

Table 20 lists the monitoring sources in this collection.

Table 20. Monitoring Sources in the DB2 Buffer Pool I/O Monitoring Collection

GUI Name	CLI Name	Returned Value Format
conn: # of direct reads (4 KB pages)	connDirectReads	Numeric: 4 KB pages
conn: # of direct writes (4 KB pages)	connDirectWrites	Numeric: 4 KB pages
conn: Average # of sectors read per direct read	connAvgSectorsReadPerDirectRead	Numeric
conn: Average # of sectors written per direct write	connAvgSectorsWritePerDirectWrite	Numeric
conn: Buffer pool time waited for prefetch (ms)	connTimeWaitedforPreFetch	Numeric: milliseconds
conn: Data page writes (4 KB pages)	connPoolDataWrites	Numeric: 4 KB pages
conn: Index page reads (4 KB pages)	connPoolIndexPReads	Numeric: 4 KB pages
conn: Index page writes (4 KB pages)	connPoolIndexWrites	Numeric: 4 KB pages
conn: Logical data reads (4 KB pages)	connPoolDataLReads	Numeric: 4 KB pages
conn: Percent buffer pool hit ratio, data+index	connPoolHitRatio	Percentage
conn: Percent buffer pool hit ratio, index	connIndxPlHitRatio	Percentage
conn: Percent package cache hit ratio	connPkgCacheHitRatio	Percentage
conn: Pool read time (s)	connPoolReadTime	Numeric: seconds
conn: Pool write time (s)	connPoolWriteTime	Numeric: seconds
conn: Total pool I/O time (s)	connTotalPoolIoTime	Numeric: seconds
db: # of cleaners due to dirty threshold	dbPlDrtyPgThrshCln	Numeric

DB2 Buffer Pool I/O Monitoring Collection

Table 20. Monitoring Sources in the DB2 Buffer Pool I/O Monitoring Collection (continued)

GUI Name	CLI Name	Returned Value Format
db: # of database files closed	dbFilesClosed	Numeric
db: # of direct read requests	dbDirectReadReqs	Numeric
db: # of direct reads from database	dbDirectReads	Numeric
db: # of direct write requests	dbDirectWriteReqs	Numeric
db: # of direct writes to database	dbDirectWrites	Numeric
db: # of log space cleaners invoked	dbPoolLsnGapClns	Numeric
db: # of page cleans in an interval	dbPageCleans	Numeric
db: # of pages for each prefetch request	dbPagesPerPrefetch	Numeric: 4 KB pages
db: # of pool reads	dbTotalPoolReads	Numeric
db: # of pool writes	dbTotalPoolWrites	Numeric
db: # of prefetch requests in an interval	dbPoolAsyncDatRdRq	Numeric
db: # of synchronous I/O	dbTotalSyncIos	Numeric
db: # of synchronous index reads (4 KB pages)	dbSyncIndexReads	Numeric: 4 KB pages
db: # of synchronous index writes (4 KB pages)	dbSyncIndexWrites	Numeric: 4 KB pages
db: # of synchronous reads	dbSyncDataReads	Numeric
db: # of synchronous writes	dbSyncDataWrites	Numeric
db: # of victim page cleaners invoked	dbPlDrtyPgStealCln	Numeric
db: Average # of async reads per pool read	dbAvgAsyncReadsPerTotReads	Numeric
db: Average # of async writes per pool write	dbAvgAsyncWritesPerTotWrites	Numeric
db: Average # of pool writes per pool read	dbAvgPlWritesPerPlReads	Numeric
db: Average # of sectors read per direct read	dbAvgSectorsReadPerDirectRead	Numeric
db: Average # of sectors written per direct write	dbAvgSectorsWritePerDirectWrite	Numeric
db: Average direct read time (ms)	dbAvgDirectReadTime	Numeric: milliseconds
db: Average direct write time (ms)	dbAvgDirectWriteTime	Numeric: milliseconds
db: Average pool I/O time (ms)	dbAvgPoolIoTime	Numeric: milliseconds
db: Average pool read time (ms)	dbAvgPoolReadTime	Numeric: milliseconds
db: Average pool write time (ms)	dbAvgPoolWriteTime	Numeric: milliseconds
db: Average synchronous data read time (ms)	dbAvgSyncReadTime	Numeric: milliseconds
db: Average synchronous data write time (ms)	dbAvgSyncWriteTime	Numeric: milliseconds
db: Average synchronous I/O (ms)	dbAvgSyncIoTime	Numeric: milliseconds
db: Avg # of pages for each cleaner	dbPagesPerClean	Numeric: 4 KB pages
db: Buffer pool async data reads (4 KB pages)	dbPoolAsyncDataReads	Numeric: 4 KB pages
db: Buffer pool async data writes	dbPoolAsyncDataWrites	Numeric

Table 20. Monitoring Sources in the DB2 Buffer Pool I/O Monitoring Collection (continued)

GUI Name	CLI Name	Returned Value Format
db: Buffer pool async index reads (4 KB pages)	dbPoolAsyncIndexReads	Numeric: 4 KB pages
db: Buffer pool async index writes	dbPoolAsyncIndexWrites	Numeric
db: Buffer pool async read time (ms)	dbPoolAsyncReadTime	Numeric: milliseconds
db: Buffer pool async write time (ms)	dbPoolAsyncWriteTime	Numeric: milliseconds
db: Buffer pool data logical reads	dbPoolDataLReads	Numeric
db: Buffer pool data pages copied from extended storage (4 KB pages)	dbDataPagesCopiedfromExtendedStorage	Numeric: 4 KB pages
db: Buffer pool data pages copied to extended storage (4 KB pages)	dbDataPagesCopiedtoExtendedStorage	Numeric: 4 KB pages
db: Buffer pool data physical reads	dbPoolDataPReads	Numeric
db: Buffer pool data writes	dbPoolDataWrites	Numeric
db: Buffer pool index logical reads	dbPoolIndexLReads	Numeric
db: Buffer pool index pages copied from extended storage (4 KB pages)	dbIndexPagesCopiedfromExtendedStorage	Numeric: 4 KB pages
db: Buffer pool index pages copied to extended storage (4 KB pages)	dbIndexPagesCopiedtoExtendedStorage	Numeric: 4 KB pages
db: Buffer pool index physical reads	dbPoolIndexPReads	Numeric
db: Buffer pool index writes	dbPoolIndexWrites	Numeric
db: Buffer pool I/Os per second	dbPoolIoRate	Numeric
db: Buffer pool time waited for prefetch (ms)	dbTimeWaitedforPreFetch	Numeric: milliseconds
db: Direct read time (ms)	dbDirectReadTime	Numeric: milliseconds
db: Direct write time (ms)	dbDirectWriteTime	Numeric: milliseconds
db: Extended storage read / write ratio	dbExtendedStorageReadWriteRatio	Percentage
db: Percent buffer pool hit ratio, data+index	dbPoolHitRatio	Percentage
db: Percent buffer pool hit ratio, index	dbIndxPlHitRatio	Percentage
db: Percent catalog cache hit ratio	dbCatCacheHitRatio	Percentage
db: Synchronous read time (ms)	dbSyncReadTime	Numeric: milliseconds
db: Synchronous write time (ms)	dbSyncWriteTime	Numeric: milliseconds
db: Total direct I/O time (ms)	dbTotalDirectIoTime	Numeric: milliseconds
db: Total pool physical I/O (ms)	dbTotalPoolIoTime	Numeric: milliseconds
db: Total pool physical read (ms)	dbTotalReadTime	Numeric: milliseconds
db: Total pool physical write (ms)	dbTotalWriteTime	Numeric: milliseconds
db: Total synchronous I/O time (ms)	dbTotalSyncIoTime	Numeric: milliseconds
dbp: # of FCM nodes	dbpTotFcmNodes	Numeric
dbp: Connection status	dbpConnStatus	Numeric
dbp: FCM buffers currently free	dbpFreeFcmBuf	Numeric
dbp: FCM connection entries currently free	dbpFcmConnFree	Numeric

DB2 Buffer Pool I/O Monitoring Collection

Table 20. Monitoring Sources in the DB2 Buffer Pool I/O Monitoring Collection (continued)

GUI Name	CLI Name	Returned Value Format
dbp: FCM request blocks currently free	dbpFcmReqBlkFree	Numeric
dbp: Minimum FCM buffers free	dbpMinFcmBufFree	Numeric
dbp: Minimum FCM connection entries free	dbpMinFcmConnEntries	Numeric
dbp: Minimum FCM message anchors free	dbpMinFcmMsgAnchors	Numeric
dbp: Minimum FCM request blocks free	dbpMinFcmReqBlks	Numeric
dbp: Percent FCM buffers currently used	dbpPctFcmBufUsed	Percentage
dbp: Percent FCM connection entries currently used	dbpPctFcmConnEntriesUsed	Percentage
dbp: Percent FCM request blocks currently used	dbpPctFcmReqBlkUsedPercentage	
dbp: Percent maximum FCM buffers used	dbpPctMaxFcmBufUsed	Percentage
dbp: Percent maximum FCM connection entries used	dbpPctMaxFcmConnEntriesUsed	Percentage
dbp: Percent maximum FCM message anchors used	dbpPctMaxFcmMsgAnchorsUsed	Percentage
dbp: Percent maximum FCM request blocks used	dbpPctMaxFcmReqBlksUsed	Percentage
dbp: Total buffers received	dbpTotBufReceived	Numeric
dbp: Total buffers sent	dbpTotBufSent	Numeric
tbsp: # of direct read requests	tbspDirectReadReqs	Numeric
tbsp: # of direct reads from database	tbspDirectReads	Numeric
tbsp: # of direct write requests	tbspDirectWriteReqs	Numeric
tbsp: # of direct writes to database	tbspDirectWrites	Numeric
tbsp: # of pool reads	tbspTotalPoolReads	Numeric
tbsp: # of pool writes	tbspTotalPoolWrites	Numeric
tbsp: # of synchronous index reads (4 KB pages)	tbspSyncIndexReads	Numeric: 4 KB pages
tbsp: # of synchronous index writes (4 KB pages)	tbspSyncIndexWrites	Numeric: 4 KB pages
tbsp: # of synchronous reads	tbspSyncDataReads	Numeric
tbsp: # of synchronous writes	tbspSyncDataWrites	Numeric
tbsp: Average # of sectors read per direct read	tbspAvgSectorsReadPerDirectRead	Numeric
tbsp: Average # of sectors written per direct write	tbspAvgSectorsWritePerDirectWrite	Numeric
tbsp: Average direct read time (ms)	tbspAvgDirectReadTime	Numeric: milliseconds
tbsp: Average direct write time (ms)	tbspAvgDirectWriteTime	Numeric: milliseconds
tbsp: Average pool I/O time (ms)	tbspAvgPoolIoTime	Numeric: milliseconds
tbsp: Average pool read time (ms)	tbspAvgPoolReadTime	Numeric: milliseconds

Table 20. Monitoring Sources in the DB2 Buffer Pool I/O Monitoring Collection (continued)

GUI Name	CLI Name	Returned Value Format
tbsp: Average pool write time (ms)	tbspAvgPoolWriteTime	Numeric: milliseconds
tbsp: Average synchronous data read time (ms)	tbspAvgSyncReadTime	Numeric: milliseconds
tbsp: Average synchronous data write time (ms)	tbspAvgSyncWriteTime	Numeric: milliseconds
tbsp: Average synchronous I/O time (ms)	tbspAvgSyncIoTime	Numeric: milliseconds
tbsp: Buffer pool async data reads	tbspPoolAsyncDataReads	Numeric
tbsp: Buffer pool async data writes	tbspPoolAsyncDataWrites	Numeric
tbsp: Buffer pool async index reads (4 KB pages)	tbspPoolAsyncIndexReads	Numeric: 4 KB pages
tbsp: Buffer pool async index writes	tbspPoolAsyncIndexWrites	Numeric
tbsp: Buffer pool async read time (ms)	tbspPoolAsyncReadTime	Numeric: milliseconds
tbsp: Buffer pool async write time (ms)	tbspPoolAsyncWriteTime	Numeric: milliseconds
tbsp: Buffer pool data logical reads	tbspPoolDataLReads	Numeric
tbsp: Buffer pool data pages copied from extended storage (4 KB pages)	tbspDataPagesCopiedfromExtendedStorage	Numeric: 4 KB pages
tbsp: Buffer pool data pages copied to extended storage (4 KB pages)	tbspDataPagesCopiedtoExtendedStorage	Numeric: 4 KB pages
tbsp: Buffer pool data physical reads	tbspPoolDataPReads	Numeric
tbsp: Buffer pool data writes	tbspPoolDataWrites	Numeric
tbsp: Buffer pool I/Os per second	tbspPoolIoRate	Numeric
tbsp: Buffer pool index logical reads	tbspPoolIndexLReads	Numeric
tbsp: Buffer pool index pages copied from extended storage (4 KB pages)	tbspIndexPagesCopiedfromExtendedSorage	Numeric: 4 KB pages
tbsp: Buffer pool index pages copied to extended storage (4 KB pages)	tbspIndexPagesCopiedtoExtendedStorage	Numeric: 4 KB pages
tbsp: Buffer pool index physical reads	tbspPoolIndexPReads	Numeric
tbsp: Buffer pool index writes	tbspPoolIndexWrites	Numeric
tbsp: Direct read time	tbspDirectReadTime	Numeric: milliseconds
tbsp: Direct write time (ms)	tbspDirectWriteTime	Numeric: milliseconds
tbsp: Extended storage read/write ratio	tbspExtendedStorageReadWriteRatio	Percentage
tbsp: File closed	tbspFilesClosed	Numeric
tbsp: Percent buffer pool hit ratio, data+index	tbspPoolHitRatio	Percentage
tbsp: Percent buffer pool hit ratio, index	tbspIndxPlHitRatio	Percentage
tbsp: Percent prefetch satisfied	tbspPctAsyDatRdRqS	Percentage
tbsp: Percent total I/O	tbspPctTotalIo	Percentage
tbsp: Prefetch requests in an interval	tbspPlAsyncDatRdRq	Numeric
tbsp: Synchronous read time (ms)	tbspSyncReadTime	Numeric: milliseconds
tbsp: Synchronous write time (ms)	tbspSyncWriteTime	Numeric: milliseconds

DB2 Buffer Pool I/O Monitoring Collection

Table 20. Monitoring Sources in the DB2 Buffer Pool I/O Monitoring Collection (continued)

GUI Name	CLI Name	Returned Value Format
tbsp: Total direct I/O time (ms)	tbspTotalDirectIoTime	Numeric: milliseconds
tbsp: Total pool physical I/O time (ms)	tbspTotalPoolIoTime	Numeric: milliseconds
tbsp: Total pool physical read time (ms)	tbspTotalReadTime	Numeric: milliseconds
tbsp: Total pool physical write time (ms)	tbspTotalWriteTime	Numeric: milliseconds
tbsp: Total synchronous I/O	tbspTotalSyncIos	Numeric
tbsp: Total synchronous I/O time (ms)	tbspTotalSyncIoTime	Numeric: milliseconds

conn: # of direct reads (4 KB pages)

Description

Returns the number of 4 KB pages that the application read from disk without using the buffer pool.

CLI **connDirectReads** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This returned value is used in calculating the returned value for the “conn: Average # of sectors read per direct read” monitoring source (page 53).

If you are using system monitoring sources to track I/O for a particular device, you can use the returned value from this monitor to distinguish database application I/O from nondatabase application I/O.

Direct reads are typically used when:

- Reading LONG VARCHAR columns
- Reading large object (LOB) columns
- Performing a backup

conn: # of direct writes (4 KB pages)

Description

Returns the number of 4 KB pages that the application wrote to disk without using the buffer pool.

CLI **connDirectWrites** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID
Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
This returned value is used in calculating the returned value for the "conn: Average # of sectors written per direct write" monitoring source (page 54).

If you are using system monitoring sources to track I/O for a particular device, you can use the returned value from this monitor to distinguish database application I/O from nondatabase application I/O.

Direct writes are typically used when:

- Writing LONG VARCHAR columns
- Writing large object (LOB) columns
- Performing a restore
- Performing a load

conn: Average # of sectors read per direct read

Description
Returns the average number of 512-byte sectors that are read in a direct read by this application.

CLI **connAvgSectorsReadPerDirectRead** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID
Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Direct reads result in poor performance because the data is physically read from disk each time, instead of using the buffer pool. If you are using system monitoring sources to track I/O for the device, this value helps you distinguish database I/O from nondatabase I/O.

conn: Average # of sectors written per direct write

Description

Returns the average number of 512-byte sectors that are written in a direct write by this application.

CLI **connAvgSectorsWritePerDirectWrite** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Direct writes result in poor performance because the data is physically written from disk each time, instead of using the buffer pool. If you are using system monitoring sources to track I/O for the device, this value helps you distinguish database I/O from nondatabase I/O.

conn: Buffer pool time waited for prefetch (ms)

Description

Returns the time (in milliseconds) that an application spent waiting for an I/O server (prefetcher) to finish loading pages into the buffer pool.

CLI **connTimeWaitedforPreFetch**

DB2 version

This monitoring source is supported for DB2 UDB and DB2 Enterprise-Extended Edition.

Usage notes

Use the returned value to measure and tune the number of I/O servers and I/O server sizes.

conn: Data page writes (4 KB pages)

Description

Returns the number of 4 KB buffer pool data pages written to disk by the application.

CLI **connPoolDataWrites** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

A buffer pool data page is written to disk for these reasons:

- To free a page in the buffer pool so that another data page can be read in
- To flush the buffer pool

The returned value includes data pages written to disk by both page cleaners and database agents.

conn: Index page reads (4 KB pages)

Description

Returns the number of physical read requests (in 4 KB pages) made by the application to get index pages into the buffer pool.

CLI **connPoolIndexPReads** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used to calculate the buffer pool index page hit ratio for the application (see “conn: Percent buffer pool hit ratio, index” on page 57).

conn: Index page writes (4 KB pages)

Description

Returns the number of 4 KB buffer pool index pages written to disk by the application.

CLI **connPoolIndexWrites** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID
Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
A buffer pool index page is written to disk for these reasons:

- To free a page in the buffer pool so that another index page can be read in
- To flush the buffer pool

The returned value includes index pages written to disk by both page cleaners and database agents.

conn: Logical data reads (4 KB pages)

Description
Returns the number of requests (in 4 KB pages) made by the application to read data pages using the buffer pool.

CLI **connPoolDataLReads** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID
Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value includes requests for data that is:

- Already in the buffer pool
- Read into the buffer pool to fulfill the request

The returned value is used to calculate the percent buffer pool hit ratio (data and index reads) for the application. See "conn: Percent buffer pool hit ratio, data+index" on page 57 for more information.

conn: Percent buffer pool hit ratio, data+index

Description

Returns the overall buffer pool hit ratio (as a percentage) for the application during the monitoring interval. This hit ratio includes both index and data page activity.

CLI **connPoolHitRatio** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The overall buffer pool hit ratio is the ratio of the number of accesses to pages (data and index) that are already in the buffer pool to the total number of read requests for pages (data and index) that use the buffer pool. The total number of read requests for pages includes both pages that must be loaded into the buffer pool before the request can be fulfilled and pages that are already in the buffer pool.

If the hit ratio is low (for example, a ratio of zero indicates that pages needed to be read from disk for every request), increasing the number of buffer pool pages might improve performance. The buffer pool size is set by the buffpage configuration parameter. For information on the buffpage configuration parameter, see "db: buffpage" on page 117 or the *DB2 Administration Guide* for the version of DB2 that you are using.

Typically, you should analyze buffer pool usage at the database level because the buffer pool size is configured at the database level. However, if the buffer pool hit ratio is low for an application, it can indicate the reason why that application has poorer performance when compared to other applications connected to the same database.

conn: Percent buffer pool hit ratio, index

Description

Returns the application's index page hit ratio (as a percentage) for the buffer pool during the monitoring interval.

CLI **connIndxPlHitRatio** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The index page hit ratio for the buffer pool is the ratio of the number of accesses to index pages that are already in the buffer pool to the total number of read requests for index pages that use the buffer pool. The total number of read requests for index pages includes both those index pages that must be loaded into the buffer pool before the request can be fulfilled and those index pages that are already in the buffer pool.

If the hit ratio is low, increasing the number of buffer pool pages might improve performance. For information on increasing the buffer pool size using the buffpage configuration parameter, see the *DB2 Administration Guide* for the version of DB2 that you are using.

Typically, you should analyze buffer pool usage at the database level because the buffer pool size is configured at the database level. However, if the buffer pool hit ratio is low for an application, it can indicate the reason why that application has poorer performance when compared to other applications connected to the same database.

conn: Percent package cache hit ratio

Description

Returns the application’s package cache hit ratio (as a percentage) during the last interval. The package cache hit ratio is the ratio of the difference between the package cache lookups and the package cache inserts, to all package cache lookups.

CLI **connPkgCacheHitRatio** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This percentage tells you whether the package cache is being used effectively by this application. If the hit ratio is high (greater than 80%), the package cache is performing well. A smaller percentage might indicate that the package cache should be increased. However, it might not be

worthwhile to increase the size of the package cache for an application that executes only once a day. The size of the package cache is set by the pckcachesz configuration parameter (see “db: pckcachesz” on page 126 for more information).

You can use the returned value from the “conn: Percent DDL SQL” monitoring source (page 184) to determine whether or not the execution of DDL statements is impacting the performance of the package cache. Execution of DDL statements can lower the percent package hit ratio (indicating poor performance), but in this case, increasing the size of the package cache will not necessarily improve overall performance. If DDL statements rarely occur, then performance might improve if you increase pckcachesz. If DDL statements frequently occur, then you might want to limit the use of DDL statements (for example, to certain time periods).

Use the “conn: Dynamic SQL statements” and “conn: Static SQL statements” monitoring sources (see pages 180 and 189) to provide information on the quantity and type of sections being cached for this application. Caching of a section for static or dynamic SQL statements can improve performance, especially when the same statement is used multiple times by the application.

conn: Pool read time (s)

Description

Returns the total time (in seconds) that the application spent processing read requests that caused data or index pages to be physically read from the disk to the buffer pool.

CLI **connPoolReadTime** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value indicates how much time the application spends in I/O operations for reading pages from disk and into the buffer pool.

conn: Pool write time (s)

Description

Returns the total time (in seconds) that an application spent physically writing data or index pages from the buffer pool to disk.

CLI **connPoolWriteTime** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID
Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value indicates how much time the application spends in I/O operations for writing pages from the buffer pool to disk.

conn: Total pool I/O time (s)

Description
Returns the total time (in seconds) that an application spent performing buffer pool I/O operations (reading or writing pages).

CLI **connTotalPoolIoTime** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID
Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is an indication of how much time the application spends in I/O operations using the buffer pool.

db: # of cleaners due to dirty threshold

Description
Returns the number of times that a page cleaner was invoked because the buffer pool reached the dirty page threshold criterion for the database.

CLI **dbPlDrtyPgThrshCln**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the return value to help you tune the value for the chngpgs_thresh

configuration parameter. This parameter sets a limit on the number of changed pages that can reside in the buffer pool. When the number of changed pages exceeds this value, the page cleaners are triggered to write out pages.

If the returned value is high, your value for chngpgs_thresh might be too low. In this case, pages might be written out too early, requiring them to be read back in if requested again. If the returned value is low, your value for chngpgs_thresh might be too high. In this case, too many pages might accumulate in the buffer pool, resulting in pages written out synchronously and performance reduction. For more information, see the *DB2 Administration Guide* for the version of DB2 that you are using.

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: # of database files closed

Description
Returns the total number of database files that are closed.

CLI **dbFilesClosed**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to help you determine the best value for the maxfilop configuration parameter.

The database manager opens files for reading into and writing out of the buffer pool. The maximum number of database files that are open by an application at any time is controlled by the maxfilop parameter. If the maximum is reached, one file is closed before a new file is opened, which can slow down performance.

For more information about the maxfilop parameter, see the *DB2 Administration Guide* for the version of DB2 that you are using.

db: # of direct read requests

Description
Returns the number of requests to perform a direct read from disk of one or more sectors of data for the database.

CLI **dbDirectReadReqs**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is used in calculating the returned value for the “db: Average # of sectors read per direct read” monitoring source (page 69).

Direct reads are performed in units, the smallest being a 512-byte sector. They are used when:

- Reading LONG VARCHAR columns
- Reading LOB columns
- Performing a backup

db: # of direct reads from database

Description

Returns the number of read operations for the database that did not use the database buffer pool.

CLI **dbDirectReads**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculating the returned value for the “db: Average # of sectors read per direct read” monitoring source (page 69).

Direct reads are performed in units, the smallest being a 512-byte sector. They are used when:

- Reading LONG VARCHAR columns
- Reading LOB columns
- Performing a backup

db: # of direct write requests

Description

Returns the number of requests to perform a direct write of one or more sectors of data for the database.

CLI **dbDirectWriteReqs**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculating the returned value for the “db: Average # of sectors written per direct write” monitoring source (page 69).

Direct writes are performed in units, the smallest being a 512-byte sector. They are used when:

- Writing LONG VARCHAR columns
- Writing LOB columns
- Performing a restore
- Performing a load

db: # of direct writes to database

Description

Returns the number of write operations for the database that do not use the buffer pool.

CLI **dbDirectWrites**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculating the returned value for the “db: Average # of sectors written per direct write” monitoring source (page 69).

Direct writes are performed in units, the smallest being a 512-byte sector. They are used when:

- Writing LONG VARCHAR columns
- Writing LOB columns
- Performing a restore
- Performing a load

db: # of log space cleaners invoked

Description

Returns the number of times a page cleaner was invoked because the used logging space reached a predefined criterion for the database.

CLI **dbPoolLsnGapClns**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help determine whether you have enough space for logging, and whether you need more log files or larger log files.

Page cleaners are invoked if the size of the log that would need to be read during a database recovery would exceed the value given by the following formula: logprimary × logfilsiz × softmax, where:

- The logprimary configuration parameter sets the number of primary log files.
- The logfilsiz configuration parameter sets the size of the log files.
- The softmax configuration parameter determines the percentage of the log to be filled before the database manager takes a soft checkpoint.

For more information on these parameters, see the *DB2 Administration Guide* for the version of DB2 that you are using.

Knowing how often log space cleaners are invoked can help you to adjust the values for logfilsiz and softmax. While lowering softmax can potentially reduce database recovery time, it can also increase system overhead because soft checkpoints are taken more frequently. When using this monitoring source, it is recommended that you check the percent increase (% increase) during the monitoring interval.

Default monitoring schedule

Every 60 minutes without any time restrictions.

db: # of page cleans in an interval

Description

Returns the number of times a page cleaner was invoked for the database (for any reason) during the monitoring interval.

CLI **dbPageCleans**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine how often pages are written to disk by

the database's page cleaners. If this value increases over time, you might want to increase the number of page cleaners that you have defined. The number of page cleaners is determined by the num_iocleaners configuration parameter (see "db: num_iocleaners" on page 124).

db: # of pages for each prefetch request

Description

Returns the number of data pages read per prefetch request for the database during the monitoring interval.

CLI **dbPagesPerPrefetch**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine the amount of asynchronous I/O done in each interaction with the prefetcher. See "db: num_ioservers" on page 125 for more information about prefetchers.

An excessively low returned value indicates that you need more I/O servers. The more I/O servers that you have, the better your query performance.

db: # of pool reads

Description

Returns the total number of physical read requests to get data or index pages into the buffer pool for the database.

CLI **dbTotalPoolReads**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculating the returned values for other monitoring sources; for example, the overall buffer pool hit ratio, which is returned by the "db: Percent buffer pool hit ratio, data+index" monitoring source (page 80).

db: # of pool writes

Description

Returns the total number of times that buffer pool data or index pages were physically written to disk for the database.

CLI **dbTotalPoolWrites**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculating the returned values for the "db: Average # of pool writes per pool read" monitoring source (page 68) and the "db: Average # of async writes per pool write" monitoring source (page 68).

db: # of prefetch requests in an interval

Description
Returns the number of asynchronous read requests for the database during the monitoring interval.

CLI **dbPoolAsyncDatRdRq**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is used in calculations for the "db: # of pages for each prefetch request" monitoring source (page 64). This value can help you determine the amount of asynchronous I/O done in each interaction with the prefetcher. For more information on prefetchers, see "db: num_ioservers" on page 125 .

db: # of synchronous I/O

Description
Returns the total number of synchronous reads and writes for both data and index pages.

CLI **dbTotalSyncIos**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to gain insight into how well the prefetchers and page cleaners are working and help you tune the num_ioservers and num_iocleaners configuration parameters. For more information on these parameters, see "db: num_ioservers" on page 125, "db: num_iocleaners" on page 124 , or the *DB2 Administration Guide* for the version of DB2 that you are using.

Because synchronous I/O operations are performed by database agents, a high number of them can slow performance.

db: # of synchronous index reads (4 KB pages)

Description
Returns the number of 4 KB physical index pages that were read synchronously for the database.

CLI **dbSyncIndexReads**

DB2 version
This monitoring source is supported for DB2 UDB and DB2 Enterprise-Extended Edition.

Usage notes
By comparing the ratio of asynchronous reads (see "db: Buffer pool async index reads (4 KB pages)" on page 73) to synchronous reads, you can determine how well the prefetchers are working. This ratio can be helpful when you are tuning the num_ioservers configuration parameters (see "db: num_ioservers" on page 125).

For more information, see the *DB2 Administration Guide* for the version of DB2 that you are using.

db: # of synchronous index writes (4 KB pages)

Description

Returns the number of 4 KB physical index page write requests that were performed synchronously for the database.

CLI **dbSyncIndexWrites**

DB2 version

This monitoring source is supported for DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

By comparing the ratio of asynchronous writes (see “db: Buffer pool async index writes” on page 73) to synchronous writes, you can determine how well the buffer pool page cleaners are performing. This ratio can be helpful when you are tuning the num_iocleaners configuration parameter (see “db: num_iocleaners” on page 124).

For more information, see the *DB2 Administration Guide* for the version of DB2 that you are using.

db: # of synchronous reads

Description

Returns the number of times physical data pages were read synchronously into the buffer pool within the database.

CLI **dbSyncDataReads**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Compare the returned value to the value returned by the “db: Buffer pool async data reads (4 KB pages)” monitoring source (page 72) to gain insight into how well the prefetchers are working and to help you tune the num_ioservers configuration parameter. For more information on this parameter and prefetchers, see “db: num_ioservers” on page 125 or the *DB2 Administration Guide* for the version of DB2 that you are using.

Because synchronous reads are performed by database agents, a high number of them can slow performance.

db: # of synchronous writes

Description

Returns the number of times that data pages were written synchronously from the buffer pool to disk for the database.

CLI **dbSyncDataWrites**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Compare the returned value with the value returned by the "db: Buffer pool async data writes" monitoring source (page 73) to gain insight into how well the page cleaners are working and to help you tune the num_iocleaners configuration parameter. For more information on this parameter and page cleaners, see "db: num_iocleaners" on page 124 or the *DB2 Administration Guide* for the version of DB2 that you are using.

Because synchronous writes are performed by database agents, a high number of them can slow performance.

db: # of victim page cleaners invoked

Description

Returns the number of times a page cleaner was invoked because a synchronous write was needed to make space for new pages in the buffer pool.

CLI **dbPlDrtyPgStealCln**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

When the number of changed (dirty) pages in the buffer pool reaches the value for the chngpgs_thresh configuration parameter, *victim* page cleaning is invoked to asynchronously write these pages to disk. Victim pages are pages that have been selected for an action to be performed on them, for example, pages that will be written to disk.

If the returned value is low, you might have too many page cleaners defined and they might be writing out changed pages that would be changed again later. Aggressive cleaning defeats one purpose of the buffer pool, which is to defer writing to disk until the last possible moment.

If the returned value is high, you might have too few page cleaners defined. Too few page cleaners can increase the time for recovering the database after failures.

The number of page cleaners defined is determined by the value of the num_iocleaners configuration parameter (see "db: num_iocleaners" on page 124 for more information).

Default monitoring schedule

Every 60 minutes without any time restrictions.

db: Average # of async reads per pool read

Description

Returns the ratio of buffer pool asynchronous data reads to the total number of pool reads for the database.

CLI **dbAvgAsyncReadsPerTotReads**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you gain insight into how well the

prefetchers are working and to help you tune the num_ioservers configuration parameter (see “db: num_ioservers” on page 125). If the returned value is low, there might not be enough I/O servers to prefetch data into the buffer, causing the database manager agents to spend extra time on physical reads. Increase the number of I/O servers by increasing the value of num_ioservers configuration parameter. If too many servers are allocated, it will not hurt system performance because the extra I/O servers are not used.

For more information on this parameter and prefetchers, see “db: num_ioservers” on page 125 or the *DB2 Administration Guide* for the version of DB2 that you are using.

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: Average # of async writes per pool write

Description
Returns the ratio of buffer pool asynchronous writes (data and index) to the total number of pool writes for the database.

CLI **dbAvgAsyncWritesPerTotWrites**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to help you gain insight into how well the page cleaners are working and to help you tune the num_iocleaners configuration parameter (see “db: num_iocleaners” on page 124). If the returned value is low, increase the number of I/O cleaners by increasing the value of num_iocleaners. If the returned value is high, you might save system resources by decreasing the number of I/O cleaners (by decreasing the value of num_iocleaners).

For more information on the num_iocleaners parameter, see the *DB2 Administration Guide* for the version of DB2 that you are using.

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: Average # of pool writes per pool read

Description
Returns the ratio of total pool writes to pool reads for the database.

CLI **dbAvgPlWritesPerPlReads**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
If the returned value is greater than 1, you might be able to improve performance by increasing the available buffer pool space. A returned value greater than 1 indicates that at least one write to disk had to occur (to either free a page in the buffer pool, or to flush the buffer pool) before a page could be read into the buffer pool.

You can increase the available buffer pool space by freeing the space more often (by increasing the chngpgs_thresh or num_iocleaners configuration parameters) or by increasing the total space for the buffer pool (by increasing the buffpage configuration parameter).

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: Average # of sectors read per direct read

Description
Returns the average number of 512-byte sectors that are read by a direct read for the database.

CLI **dbAvgSectorsReadPerDirectRead**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Direct reads do not use the buffer pool, and so result in poor performance because the data is physically read from disk each time. If you are using system monitoring sources to track I/O for the device, this value helps you distinguish database I/O from nondatabase I/O.

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: Average # of sectors written per direct write

Description
Returns the average number of 512-byte sectors that are written by a direct write for the database.

CLI **dbAvgSectorsWritePerDirectWrite**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Direct writes do not use the buffer pool, and so result in poor performance because the data is physically written from disk each time. If you are using system monitoring sources to track I/O for the device, this value helps you distinguish database I/O from nondatabase I/O.

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: Average direct read time (ms)

Description
Returns the average time (in milliseconds) spent for performing direct reads for the database.

CLI **dbAvgDirectReadTime**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
A high average time might indicate the existence of an I/O conflict.

db: Average direct write time (ms)

Description
Returns the average time (in milliseconds) for performing direct writes for the database.

CLI **dbAvgDirectWriteTime**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
A high average time might indicate the existence of an I/O conflict.

db: Average pool I/O time (ms)

Description
Returns the average time (in milliseconds) for performing buffer pool I/O operations (reading or writing) for the database.

CLI **dbAvgPoolIoTime**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
A high average time might indicate the existence of an I/O conflict. In this case, you might need to move data to a different device.

The returned value includes the time spent in asynchronous I/O operations (which are performed by prefetchers and page cleaners).

db: Average pool read time (ms)

Description
Returns the average time (in milliseconds) for processing read requests that caused data or index pages to be physically read from disk to buffer pool for the database.

CLI **dbAvgPoolReadTime**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
A high average time might indicate the existence of an I/O conflict. In this case, you might need to move data to a different device.

The returned value includes the time spent in asynchronous read operations that are performed by prefetchers.

db: Average pool write time (ms)

Description

Returns the average time (in milliseconds) for processing write requests that caused data or index pages to be physically written from buffer pool to disk for the database.

CLI **dbAvgPoolWriteTime**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

A high average time might indicate the existence of an I/O conflict. In this case, you might need to move data to a different device.

The returned value includes the time spent in asynchronous write operations that are performed by page cleaners.

db: Average synchronous data read time (ms)

Description

Returns the average time (in milliseconds) for synchronous data reads for the database.

CLI **dbAvgSyncReadTime**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to analyze the I/O work being performed for the database. Synchronous read operations for a database are performed by database manager agents. Asynchronous reads are performed by prefetchers, which read data pages from disk into the buffer pool in anticipation of their use.

If the returned value is high compared to the average time spent in asynchronous data reads, you might want to increase the amount of prefetching by increasing the num_ioservers configuration parameter or turning on the seqdetect configuration parameter. For more information, see “db: num_ioservers” on page 125, “db: seqdetect” on page 127, or the *DB2 Administration Guide* for the version of DB2 that you are using.

db: Average synchronous data write time (ms)

Description

Returns the average time (in milliseconds) for synchronous data writes for the database.

CLI **dbAvgSyncWriteTime**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to analyze the I/O work being performed for the database. Synchronous write operations for a database are performed by

database manager agents. Asynchronous writes are performed by page cleaners, which write out changed pages to disk and free up space in the buffer pool.

If the returned value is high compared to the average time spent in asynchronous data writes, you might want to increase the number of page cleaners by using the num_iocleaners configuration parameter. For more information, see “db: num_iocleaners” on page 124 or the *DB2 Administration Guide* for the version of DB2 that you are using.

db: Average synchronous I/O (ms)

Description

Returns the average time (in milliseconds) for synchronous I/O operations for the database.

CLI **dbAvgSyncIoTime**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to analyze the I/O work being performed for the database. Synchronous I/O operations for a database are performed by database manager agents. Asynchronous I/O operations are performed by prefetchers (reads) and page-cleaners (writes). In general, asynchronous I/O helps your applications run faster.

db: Avg # of pages for each cleaner

Description

Returns the average number of pages written per page cleaner invoked for the database during the monitoring interval.

CLI **dbPagesPerClean**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine how many pages are handled by this database’s page cleaners. If this value increases over time, you might want to increase the number of page cleaners that you have defined. The number of page cleaners is determined by the num_iocleaners configuration parameter.

See “db: num_iocleaners” on page 124 for more information about the effect of page cleaners on database performance.

db: Buffer pool async data reads (4 KB pages)

Description

Returns the number of 4 KB data pages read asynchronously into the buffer pool for the database.

CLI **dbPoolAsyncDataReads**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is used in calculating the returned value for the “db: Average # of async reads per pool read” monitoring source (page 67).

db: Buffer pool async data writes

Description
Returns the number of times that a buffer pool data page was written asynchronously to disk for the database.

CLI **dbPoolAsyncDataWrites**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is used in calculating the returned value for the “db: Average # of async writes per pool write” monitoring source (page 68).

db: Buffer pool async index reads (4 KB pages)

Description
Returns the number of 4 KB index pages read asynchronously into the buffer pool by a prefetcher for the database.

CLI **dbPoolAsyncIndexReads**

DB2 version
This monitoring source is supported for DB2 UDB and DB2 Enterprise-Extended Edition.

Usage notes
By comparing the ratio of asynchronous to synchronous reads (see “db: # of synchronous index reads (4 KB pages)” on page 65), you can determine how well the prefetchers are working. This ratio can be helpful when you are tuning the num_ioservers configuration parameters (see “db: num_ioservers” on page 125). Asynchronous reads are performed by database manager prefetchers.

For more information, see the *DB2 Administration Guide* for the version of DB2 that you are using.

db: Buffer pool async index writes

Description
Returns the number of times that a buffer pool index page was written asynchronously to disk for the database.

CLI **dbPoolAsyncIndexWrites**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculating the returned value for the “db: Average # of async writes per pool write” monitoring source (page 68).

By comparing the ratio of asynchronous to synchronous writes (see “db: # of synchronous index writes (4 KB pages)” on page 66), you can determine how well the buffer pool page cleaners are performing. This ratio can be helpful when you are tuning the num_iocleaners configuration parameter (see “db: num_iocleaners” on page 124).

For more information, see the *DB2 Administration Guide* for the version of DB2 that you are using.

db: Buffer pool async read time (ms)

Description

Returns the total time (in milliseconds) that database manager prefetchers spent reading data or index pages into the buffer pool for the database.

CLI **dbPoolAsyncReadTime**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Compare the returned value to the value returned by the “db: Synchronous read time (ms)” monitoring source (page 82) to understand where I/O time is being spent.

db: Buffer pool async write time (ms)

Description

Returns the total time (in milliseconds) that database manager page cleaners spent writing data or index pages from the buffer pool to disk for the database.

CLI **dbPoolAsyncWriteTime**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Compare the returned value to the value returned by the “db: Synchronous write time (ms)” monitoring source (page 82) to understand where I/O time is being spent.

db: Buffer pool data logical reads

Description

Returns the number of logical read requests for data pages that went through the buffer pool within the database.

CLI **dbPoolDataLReads**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used to calculate the value for the “db: Percent buffer pool hit ratio, data+index” monitoring source (page 80).

The returned value includes requests for data that is:

- Already in the buffer pool.
- Read from disk into the buffer pool to fulfill the request. (The count for these is given by the “db: Buffer pool data physical reads” monitoring source.

db: Buffer pool data pages copied from extended storage (4 KB pages)

Description

Returns the number of 4 KB buffer pool data pages copied from extended storage within the database.

CLI **dbDataPagesCopiedfromExtendedStorage**

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

Required pages are copied from extended storage to the buffer pool if they are not in the buffer pool but are in extended storage. This copying might incur the cost of connecting to the shared memory segment but saves the cost of a disk read.

db: Buffer pool data pages copied to extended storage (4 KB pages)

Description

Returns the number of 4 KB buffer pool data pages copied to extended storage within the database.

CLI **dbDataPagesCopiedtoExtendedStorage**

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

Pages are copied from the buffer pool to extended storage when they are selected as victim pages. This copying is required to make space for new pages in the buffer pool.

db: Buffer pool data physical reads

Description

Returns the number of read requests that required I/O to get data pages into the buffer pool within the database.

CLI **dbPoolDataPReads**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used to calculate the return values for the “db:

Percent buffer pool hit ratio, data+index" and the "db: # of synchronous reads" monitoring sources (see pages 80 and 66).

db: Buffer pool data writes

Description

Returns the number of times that a buffer pool data page was physically written to disk within the database.

CLI **dbPoolDataWrites**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

If the returned value is high compared to the value returned by the "db: Buffer pool data physical reads" monitoring source (page 75), you might be able to improve performance by increasing the available buffer pool space. You can increase available buffer pool space by:

- Freeing space more often by writing out changed pages. You can achieve this by increasing the chngpgs_thresh or num_iocleaners configuration parameter (see "db: num_iocleaners" on page 124).
- Increasing the total space for the buffer pool. You can achieve this by increasing the buffpage configuration parameter.

db: Buffer pool index logical reads

Description

Returns the number of logical read requests for index pages that went through the buffer pool within the database.

CLI **dbPoolIndexLReads**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculations for the "db: Percent buffer pool hit ratio, data+index" and the "db: Percent buffer pool hit ratio, index" monitoring sources (see pages 80 and 80).

The returned value includes requests for index pages that are:

- Already in the buffer pool.
- Read from disk into the buffer pool to fulfill the request (the count for these is given by the "db: Buffer pool index physical reads" monitoring source).

db: Buffer pool index pages copied from extended storage (4 KB pages)

Description

Returns the number of 4 KB buffer pool index pages copied from extended storage within the database.

CLI **dbIndexPagesCopiedfromExtendedStorage**

DB2 version
This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes
Required index pages are copied from extended storage to the buffer pool if they are not in the buffer pool but are in extended storage. This copying might incur the cost of connecting to the shared memory segment but saves the cost of a disk read.

db: Buffer pool index pages copied to extended storage (4 KB pages)

Description
Returns the number of 4 KB buffer pool index pages copied to extended storage within the database.

CLI **dbIndexPagesCopiedtoExtendedStorage**

DB2 version
This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes
Pages are copied from the buffer pool to extended storage when they are selected as victim pages. This copying is required to make space for new pages in the buffer pool.

db: Buffer pool index physical reads

Description
Returns the number of physical read requests to get index pages into the buffer pool within the database.

CLI **dbPoolIndexPReads**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is used in calculations for other buffer pool monitoring sources, for example, the “db: Percent buffer pool hit ratio, index” and “db: Percent buffer pool hit ratio, data+index” monitoring sources (see pages 80 and 80).

db: Buffer pool index writes

Description
Returns the number of times that a buffer pool index page was physically written to disk for the database.

CLI **dbPoolIndexWrites**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
If the returned value is high compared to the value returned by the “db:

Buffer pool index physical reads" monitoring source (page 77), you might be able to improve performance by increasing available buffer pool space. You can increase available buffer pool space by:

- Freeing space more often by writing out changed pages. You can achieve this by increasing the chngpgs_thresh or num_iocleaners configuration parameter (see "db: num_iocleaners" on page 124).
- Increasing the total space for the buffer pool. You can achieve this by increasing the buffpage configuration parameter.

db: Buffer pool I/Os per second

Description
Returns the rate (per second) for buffer pool I/O within the database. For this monitoring source, buffer pool I/O includes all physical data and index pages that go through the buffer pool when read or written.

CLI **dbPoolIoRate**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine how effective your data storage device is. A low value can indicate the presence of an I/O wait, which might indicate that you should move data to a different device.

db: Buffer pool time waited for prefetch (ms)

Description
Returns the time (in milliseconds) that all applications spent waiting for an I/O server (prefetcher) to finish loading pages into the buffer pool for the database during the monitoring interval.

CLI **dbTimeWaitedforPreFetch**

DB2 version
This monitoring source is supported for DB2 UDB and DB2 Enterprise-Extended Edition.

Usage notes
Use the returned value to measure and tune the number of I/O servers and I/O server sizes.

db: Direct read time (ms)

Description
Returns the time (in milliseconds) for performing the direct reads for the database.

CLI **dbDirectReadTime**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculations for the “db: Average direct read time (ms)” monitoring source (page 69). A high average time might indicate an I/O conflict.

db: Direct write time (ms)

Description

Returns the time (in milliseconds) for performing the direct writes for the database.

CLI **dbDirectWriteTime**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculations for the “db: Average direct write time (ms)” monitoring source (page 70). A high average time might indicate an I/O conflict.

db: Extended storage read/write ratio

Description

Returns the ratio (as a percentage) of pages (data+index) copied from extended storage to pages copied to extended storage within the database during the monitoring interval.

CLI **dbExtendedStorageReadWriteRatio**

DB2 version

This monitoring source is supported for DB2 UDB and DB2 Enterprise-Extended Edition.

Usage notes

When a page is transferred from extended storage to the buffer pool, you save a system I/O call. However, you still incur the cost of attaching to the extended memory segment, copying the page, and detaching from the segment.

Use the returned value to determine if you would benefit from using extended storage. The higher the ratio, the more likely you are to benefit. In general, extended storage is particularly useful if I/O activity is very high on your system.

For information on related monitoring sources, see “db: Buffer pool data pages copied from extended storage (4 KB pages)” on page 75, “db: Buffer pool data pages copied to extended storage (4 KB pages)” on page 75, “db: Buffer pool index pages copied from extended storage (4 KB pages)” on page 76 , and “db: Buffer pool index pages copied to extended storage (4 KB pages)” on page 77.

For more information on extended storage, see the *DB2 System Monitor Guide and Reference* for the version of DB2 that you are using.

db: Percent buffer pool hit ratio, data+index

Description

Returns the overall buffer pool hit ratio (as a percentage) for the database during the monitoring interval. This hit ratio includes both index and data page activity.

CLI **dbPoolHitRatio**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The overall buffer pool hit ratio indicates the percentage of page requests for which the database manager did not need to load a page from disk to service. That is, the page was already in the buffer pool. The greater the buffer pool hit ratio, the lower the frequency of disk I/O.

If the hit ratio is low (for example, a ratio of zero indicates that pages needed to be read in for every request), increasing the number of buffer pool pages might improve performance. The buffer pool size is set by the buffpage configuration parameter. For information on the buffpage configuration parameter, see “db: buffpage” on page 117 or the *DB2 Administration Guide* for the version of DB2 that you are using.

For a large database, increasing the buffer pool size might have a minimal effect on the buffer pool hit ratio. Such a database can have so large a number of data pages, that the statistical chance of a hit is not increased by increasing buffpage. However, even though the data might be too large to fit in the buffer pool, the entire index might fit. In this case, you could tune buffpage until the overall buffer pool hit ratio stops increasing, then tune buffpage until the buffer pool index hit ratio (returned by the “db: Percent buffer pool hit ratio, index” monitoring source on page 80) no longer increases.

Default thresholds and actions

Table 21. Default Thresholds and Actions for db: Percent buffer pool hit ratio, data+index

Response Level	Trigger When	Default Actions
critical	Less than 80%	Send Tivoli notice Change icon
severe	Less than 90%	Send Tivoli notice Change icon
warning	Less than 95%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule

Every 60 minutes without any time restrictions.

db: Percent buffer pool hit ratio, index

Description

Returns the database’s index page hit ratio (as a percentage) for the buffer pool during the monitoring interval.

CLI **dbIndxPlHitRatio**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The index page hit ratio for the buffer pool indicates the percentage of index page requests that the database manager did not need to load an index page from disk to service. That is, the index page was already in the buffer pool. The higher the returned value, the lower the frequency of disk I/O, and the faster the performance.

If the hit ratio is low, increasing the number of buffer pool pages might improve performance. For information on increasing the buffer pool size using the buffpage configuration parameter, see the *DB2 Administration Guide* for the version of DB2 that you are using.

Default thresholds and actions

Table 22. Default Thresholds and Actions for db: Percent buffer pool hit ratio, index

Response Level	Trigger When	Default Actions
critical	Less than 80%	Send Tivoli notice Change icon
severe	Less than 90%	Send Tivoli notice Change icon
warning	Less than 95%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule

Every 60 minutes without any time restrictions.

db: Percent catalog cache hit ratio

Description

Returns the catalog cache hit ratio (as a percentage) for the database during the monitoring interval.

CLI **dbCatCacheHitRatio**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The catalog cache hit ratio indicates how successful the catalog cache is. The returned ratio is the ratio of successful catalog cache lookups to the total number of catalog cache lookups. A low ratio indicates more failed lookups were made to the catalog cache while processing a table, view, or alias reference in an SQL statement.

If the returned value is low, you might want to increase the size of the catalog cache by increasing the catalogcache_sz configuration parameter. You can expect a small ratio following the first connection to the database. In addition, heavy use of DDL SQL statements can also decrease this ratio (see “db: Percent DDL SQL” on page 191 for a way to monitor DDL use).

Default thresholds and actions

Table 23. Default Thresholds and Actions for db: Percent catalog cache hit ratio

Response Level	Trigger When	Default Actions
critical	Less than 85%	Send Tivoli notice Change icon
severe	Less than 90%	Send Tivoli notice Change icon
warning	Less than 95%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: Synchronous read time (ms)

Description
Returns the time (in milliseconds) spent in synchronous reads for the database.

CLI **dbSyncReadTime**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Compare the returned value to the value returned by the “db: Buffer pool async read time (ms)” monitoring source (page 74) to understand where I/O time is being spent.

db: Synchronous write time (ms)

Description
Returns the time (in milliseconds) spent in synchronous writes to disk within the database.

CLI **dbSyncWriteTime**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Compare the returned value to the value returned by the “db: Buffer pool async write time (ms)” monitoring source (page 74) to understand where I/O time is being spent.

db: Total direct I/O time (ms)

Description
Returns the total time (in milliseconds) for direct reads and writes within the database.

CLI **dbTotalDirectIoTime**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value indicates the amount of time that the database spends in direct reads and writes. A high returned value might indicate the presence of an I/O conflict.

db: Total pool physical I/O (ms)

Description
Returns the total time (in milliseconds) for physical buffer pool I/O operations (including both synchronous and asynchronous reads and writes) within the database.

CLI **dbTotalPoolIoTime**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
A high returned value (as compared to the total number of physical buffer pool I/O operations) might indicate the presence of an I/O wait, which in turn might indicate that you should move data to a different device.

db: Total pool physical read (ms)

Description
Returns the total time (in milliseconds) for processing read requests that caused data or index pages to be physically read from disk to the buffer pool within the database.

CLI **dbTotalReadTime**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is used to calculate the average pool read time, which is returned by the “db: Average pool read time (ms)” monitoring source (page 70). This average can indicate the presence of an I/O wait, which in turn might indicate that you should move data to a different device.

db: Total pool physical write (ms)

Description
Returns the total time (in milliseconds) for buffer pool physical writes (including asynchronous writes).

CLI **dbTotalWriteTime**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is used to calculate the average pool write time, which is returned by the “db: Average pool write time (ms)” monitoring source

(page 71). This average can indicate the presence of an I/O wait, which in turn might indicate that you should move data to a different device.

db: Total synchronous I/O time (ms)

Description

Returns the total time (in milliseconds) for processing requests for synchronous reads or writes within the database.

CLI **dbTotalSyncIoTime**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is the sum of the returned values from the “db: Synchronous read time (ms)” monitoring source (page 82) and “db: Synchronous write time (ms)” monitoring source (page 82). This time is the amount of time that database agents spend doing synchronous reads and writes.

dbp: # of FCM nodes

Description

Returns the number of nodes in the current DB2 partitioned configuration.

CLI **dbpTotFcmNodes**

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

dbp: Connection status

Description

Returns the status of the communication connection between the database partition specified by the *DB2_node_number* variable and the database partition where this monitoring source executes.

CLI **dbpConnStatus** *DB2_node_number*

Arguments

DB2_node_number

Specify the number of the node for which you are checking the communication connection status. For node number specifications, see the db2nodes.cfg file.

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

Two nodes can be active, but the communication connection between them will remain inactive unless there is active communication between them.

The connection values are:

0 No current connection

1 Active connection

2 Congested connection

dbp: FCM buffers currently free

Description
Returns the number of FCM buffers that are free in the partitioned database server during the monitoring interval.

CLI **dbpFreeFcmBuf**

DB2 version
This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes
Use the returned value in conjunction with the dbms: fcm_num_buffers monitoring source to determine the current buffer pool utilization. Use this information to tune the fcm_num_buffers configuration parameter (see “dbms: fcm_num_buffers” on page 129).

dbp: FCM connection entries currently free

Description
Returns the number of connection entries that are free in the partitioned database server during the monitoring interval.

CLI **dbpFcmConnFree**

DB2 version
This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes
Use the returned value in conjunction with the fcm_num_connect configuration parameter (see “dbms: fcm_num_connect” on page 130) to determine the current connection entry utilization. You can use this information to tune the fcm_num_connect parameter.

dbp: FCM request blocks currently free

Description
Returns the number of request blocks that are free in the partitioned database server during the monitoring interval.

CLI **dbpFcmReqBlkFree**

DB2 version
This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes
Use the returned value in conjunction with the fcm_num_rqb configuration parameter (see “dbms: fcm_num_rqb” on page 130) to determine the current request block utilization. You can use this information to tune fcm_num_rqb.

dbp: Minimum FCM buffers free

Description
Returns the lowest number of free buffers reached during processing in the partitioned database server.

CLI **dbpMinFcmBufFree**

DB2 version
This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes
Use the returned value in conjunction with the dbms: fcm_num_buffers monitoring source to determine the maximum FCM buffer pool utilization. You can use this information to tune fcm_num_buffers.

dbp: Minimum FCM connection entries free

Description
Returns the lowest number of free connection entries reached during processing in the partitioned database server.

CLI **dbpMinFcmConnEntries**

DB2 version
This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes
Use the returned value in conjunction with the fcm_num_connect configuration parameter (see “dbms: fcm_num_connect” on page 130) to determine the maximum connection entry utilization. If the number of free connection entries reached during processing is low, you should increase the fcm_num_connect configuration parameter to ensure that operations do not run out of connection entries. If the number of free connection entries reached during processing is high, you can decrease the fcm_num_connect configuration parameter to conserve system resources.

dbp: Minimum FCM message anchors free

Description
Returns the lowest number of free message anchors reached during processing in the partitioned database server.

CLI **dbpMinFcmMsgAnchors**

DB2 version
This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes
Use the returned value in conjunction with the fcm_num_anchors configuration parameter (see “dbms: fcm_num_anchors” on page 129) to determine the current message anchor utilization. You can use this information to tune fcm_num_anchors.

dbp: Minimum FCM request blocks free

Description

Returns the lowest number of free request blocks reached during processing in the partitioned database server.

CLI **dbpFcmReqBlk**

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

Use the returned value in conjunction with the fcm_num_rqb configuration parameter (see “dbms: fcm_num_rqb” on page 130) to determine the maximum request block utilization. If the number of free request blocks reached during processing is low, you should increase the fcm_num_rqb configuration parameter to ensure that operations do not run out of request blocks. If the number of free request blocks reached during processing is high, you can decrease the fcm_num_rqb configuration parameter to conserve system resources.

dbp: Percent FCM buffers currently used

Description

Returns the percentage of FCM buffers that are used within the partitioned database server during the monitoring interval.

CLI **dbpPctFcmBufUsed**

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

If the percentage of FCM buffers used is high, you might want to adjust the fcm_num_buffers configuration parameter (see “dbms: fcm_num_buffers” on page 129).

For information on related monitoring sources, see “dbp: FCM buffers currently free” on page 85.

Default thresholds and actions

Table 24. Default Thresholds and Actions for dbp: Percent FCM buffers currently used

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 90%	Send Tivoli notice Change icon
warning	Greater than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

dbp: Percent FCM connection entries currently used

Description

Returns the percentage of FCM connection entries that are used within the partitioned database server during the monitoring interval.

CLI **dbpPctFcmConnEntriesUsed**

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

If the percentage of FCM connection entries used is high, you might want to increase the fcm_num_connect configuration parameter (see “dbms: fcm_num_connect” on page 130); if the percentage is low, you might want to decrease the parameter.

For information on related monitoring sources, see “dbp: FCM connection entries currently free” on page 85.

Default thresholds and actions

Table 25. Default Thresholds and Actions for dbp: Percent FCM connection entries currently used

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 90%	Send Tivoli notice Change icon
warning	Greater than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

dbp: Percent FCM request blocks currently used

Description

Returns the percentage of FCM request blocks used within the partitioned database server during the monitoring interval.

CLI **dbpPctFcmReqBlkUsed**

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

If the percentage of FCM request blocks used is high, you might want to adjust the fcm_num_rqb configuration parameter (see “dbms: fcm_num_rqb” on page 130).

For information on related monitoring sources, see “dbp: FCM request blocks currently free” on page 85.

Default thresholds and actions

Table 26. Default Thresholds and Actions for dbp: Percent FCM request blocks currently used

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 90%	Send Tivoli notice Change icon
warning	Greater than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

dbp: Percent maximum FCM buffers used

Description

Returns the percentage of maximum FCM buffers used during processing within the partitioned database server.

CLI **dbpPctMaxFcmBufUsed**

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

If the percentage of maximum FCM buffers used is high, you might want to increase the fcm_num_buffers configuration parameter (see “dbms: fcm_num_buffers” on page 129); if the percentage is low, you might want to decrease the parameter.

For information on related monitoring sources, see “dbp: Minimum FCM buffers free” on page 86.

Default thresholds and actions

Table 27. Default Thresholds and Actions for dbp: Percent maximum FCM buffers used

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 90%	Send Tivoli notice Change icon
warning	Greater than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

dbp: Percent maximum FCM connection entries used

Description

Returns the maximum percentage of FCM connection entries used during processing within the partitioned database server.

CLI **dbpPctMaxFcmConnEntriesUsed**

DB2 Buffer Pool I/O Monitoring Collection

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

If the percentage of maximum FCM connection entries used is high, you might want to increase the fcm_num_connect configuration parameter (see “dbms: fcm_num_connect” on page 130); if the percentage is low, you might want to decrease the parameter.

For information on related monitoring sources, see “dbp: FCM connection entries currently free” on page 85.

Default thresholds and actions

Table 28. Default Thresholds and Actions for dbp: Percent maximum FCM connection entries used

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 90%	Send Tivoli notice Change icon
warning	Greater than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

dbp: Percent maximum FCM message anchors used

Description

Returns the percentage of maximum FCM message anchors during processing within the partitioned database server.

CLI **dbpPctMaxFcmMsgAnchorsUsed**

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

If the percentage of maximum FCM message anchors used is high, you might want to increase the fcm_num_anchors configuration parameter (see “dbms: fcm_num_anchors” on page 129); if the percentage is low, you might want to decrease the parameter.

For information on related monitoring sources, see “dbp: Minimum FCM message anchors free” on page 86.

Default thresholds and actions

Table 29. Default Thresholds and Actions for dbp: Percent maximum FCM message anchors used

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon

Table 29. Default Thresholds and Actions for dbp: Percent maximum FCM message anchors used (continued)

Response Level	Trigger When	Default Actions
severe	Less than 90%	Send Tivoli notice Change icon
warning	Less than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

dbp: Percent maximum FCM request blocks used

Description

Returns the percentage of maximum FCM request blocks used during processing within the partitioned database server.

CLI **dbpPctMaxFcmReqBlksUsed**

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

If the percentage of maximum FCM request blocks used is high, you might want to adjust the fcm_num_rqb configuration parameter (see “dbms: fcm_num_rqb” on page 130).

For information on related monitoring sources, see “dbp: Minimum FCM request blocks free” on page 87.

Default thresholds and actions

Table 30. Default Thresholds and Actions for dbp: Percent maximum FCM request blocks used

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 90%	Send Tivoli notice Change icon
warning	Greater than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

dbp: Total buffers received

Description

Returns the total number of FCM buffers received by the node where this monitoring source executes from the database node specified in the *DB2_node_number* variable.

CLI **dbpTotBufReceived** *DB2_node_number*

Arguments

DB2_node_number
Specify the number of the node that you are monitoring. For node number specifications, see the db2nodes.cfg file.

DB2 version
This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes
Use the returned value to measure the level of traffic between the node where this monitoring source executes and another node. If the total number of FCM buffers received from the other node is high, you might want to redistribute the database or move tables to reduce the internode traffic.

dbp: Total buffers sent

Description
Returns the total number of FCM buffers sent from the database node where this monitoring source executes to the node specified by the *DB2_node_number* variable.

CLI **dbpTotBufSent** *DB2_node_number*

Arguments

DB2_node_number
Specify the number of the node that you are monitoring. For node number specifications, see the db2nodes.cfg file.

DB2 version
This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes
Use the returned value to measure the level of traffic between the current node where this monitoring source executes and the node specified by *DB2_node_number*. If the total number of FCM buffers sent to the other node is high, you might want to redistribute the database or move tables to reduce the internode traffic.

tbsp: # of direct read requests

Description
Returns the number of requests to perform a direct read from disk of one or more sectors of data for the table space.

CLI **tbspDirectReadReqs** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculating the returned value for the "tbsp: Average # of sectors read per direct read" monitoring source (page 97).

Direct reads are performed in units, the smallest being a 512-byte sector. They are used when:

- Reading LONG VARCHAR columns
- Reading LOB columns
- Performing a backup

tbsp: # of direct reads from database

Description

Returns the number of read operations for the table space that did not use the buffer pool.

CLI **tbspDirectReads** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculating the returned value for the "tbsp: Average # of sectors read per direct read" monitoring source (page 97).

Direct reads are performed in units, the smallest being a 512-byte sector. They are used when:

- Reading LONG VARCHAR columns
- Reading LOB columns
- Performing a backup

tbsp: # of direct write requests

Description

Returns the number of requests to perform a direct write of one or more sectors of data for the table space.

CLI **tbspDirectWriteReqs**

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculating the returned value for the "tbsp: Average # of sectors written per direct write" monitoring source (page 97).

Direct writes are performed in units, the smallest being a 512-byte sector. They are used when:

- Writing LONG VARCHAR columns
- Writing LOB columns
- Performing a restore
- Performing a load

tbsp: # of direct writes to database

Description

Returns the number of write operations for the table space that did not use the buffer pool.

CLI **tbspDirectWrites** *DB2_table_space_name*

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculating the returned value for the "tbsp: Average # of sectors written per direct write" monitoring source (page 97).

Direct writes are performed in units, the smallest being a 512-byte sector. They are used when:

- Writing LONG VARCHAR columns
- Writing LOB columns
- Performing a restore
- Performing a load

tbsp: # of pool reads

Description

Returns the total number of physical read requests to get data or index pages into the buffer pool for the table space.

CLI **tbspTotalPoolReads** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used with the total pool physical read time to calculate the average pool read time. The average pool read time is returned by the "tbsp: Average pool read time (ms)" monitoring source (page 98). This average can indicate the presence of an I/O wait, which might indicate that you should move data to a different device.

tbsp: # of pool writes

Description

Returns the total number of times buffer pool data or index pages were physically written to disk for the table space.

CLI **tbspTotalPoolWrites** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
If the returned value is high compared to the value returned by the "tbsp: # of pool reads" monitoring source (page 94), you might be able to improve performance by increasing the available buffer pool space. You can increase available buffer pool space by:

- Freeing space more often by writing out changed pages. You can have space freed more often by increasing the chngpgs_thresh or num_iocleaners configuration parameters (see "db: num_iocleaners" on page 124).
- Increasing the total space for the buffer pool. You can increase the space for the buffer pool by increasing the buffpage configuration parameter.

tbsp: # of synchronous index reads (4 KB pages)

Description
Returns the number of 4 KB physical index pages that were read synchronously for the table space.

CLI **tbspSyncIndexReads**

DB2 version
This monitoring source is supported for DB2 UDB and DB2 Enterprise-Extended Edition.

Usage notes
By comparing the ratio of asynchronous reads (see "tbsp: Buffer pool async index writes" on page 102) to synchronous reads, you can determine how well the prefetchers are working. This ratio can be helpful when you are tuning the num_ioservers configuration parameters (see "db: num_ioservers" on page 125).

For more information, see the *DB2 Administration Guide* for the version of DB2 that you are using.

tbsp: # of synchronous index writes (4 KB pages)

Description
Returns the number of 4 KB physical index page write requests that were performed synchronously for the table space.

CLI **tbspSyncIndexWrites**

DB2 version
This monitoring source is supported for DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes
By comparing the ratio of asynchronous writes (see "tbsp: Buffer pool async index writes" on page 102) to synchronous writes, you can determine how

well the buffer pool page cleaners are performing. This ratio can be helpful when you are tuning the num_iocleaners configuration parameter (see “db: num_iocleaners” on page 124).

For more information, see the *DB2 Administration Guide* for the version of DB2 that you are using.

tbsp: # of synchronous reads

Description

Returns the number of times that physical data pages were read synchronously into the buffer pool for the table space.

CLI **tbspSyncDataReads** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Compare the returned value to the value returned by the “tbsp: Buffer pool async data reads” monitoring source (page 100) to gain insight into how well the prefetchers are working and to help you tune the num_ioservers configuration parameter. For more information on this parameter and prefetchers, see “db: num_ioservers” on page 125 or the *DB2 Administration Guide* for the version of DB2 that you are using.

Because synchronous reads are performed by database agents, a high number of them can slow performance.

tbsp: # of synchronous writes

Description

Returns the number of times that data pages were written synchronously from the buffer pool to disk for the table space.

CLI **tbspSyncDataWrites** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Compare the returned value with the value returned by the “tbsp: Buffer pool async data writes” monitoring source (page 101) to gain insight into how well the page cleaners are working and to help you tune the num_iocleaners configuration parameter. For more information on this parameter and page cleaners, see “db: num_iocleaners” on page 124 or the *DB2 Administration Guide* for the version of DB2 that you are using.

Because synchronous writes are performed by database agents, a high number of them can slow performance.

tbsp: Average # of sectors read per direct read

Description
Returns the average number of 512-byte sectors that are read by a direct read for this table space.

CLI **tbspAvgSectorsReadPerDirectRead**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Direct reads do not use the buffer pool, and so result in poor performance because the data is physically read from disk each time. If you are using system monitoring sources to track I/O for the device, this returned value helps you distinguish database I/O from nondatabase I/O.

tbsp: Average # of sectors written per direct write

Description
Returns the average number of 512-byte sectors that are written by a direct write for this table space.

CLI **tbspAvgSectorsWritePerDirectWrite**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Direct writes do not use the buffer pool, and so result in poor performance because the data is physically written from disk each time. If you are using system monitoring sources to track I/O for the device, this returned value helps you distinguish database I/O from nondatabase I/O.

tbsp: Average direct read time (ms)

Description
Returns the average time (in milliseconds) spent for performing direct reads for the table space.

CLI **tbspAvgDirectReadTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
A high average time might indicate the existence of an I/O conflict.

tbsp: Average direct write time (ms)

Description

Returns the average time (in milliseconds) for performing direct writes for the table space.

CLI **tbspAvgDirectWriteTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

A high average time might indicate the existence of an I/O conflict.

tbsp: Average pool I/O time (ms)

Description

Returns the average time (in milliseconds) for performing buffer pool I/O operations (reading or writing) for the table space.

CLI **tbspAvgPoolIoTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

A high average time might indicate the existence of an I/O conflict. In this case, you might need to move data to a different device.

The returned value includes the time spent in asynchronous I/O operations (which are performed by prefetchers and page cleaners).

tbsp: Average pool read time (ms)

Description

Returns the average time (in milliseconds) for processing read requests that caused data or index pages to be physically read from disk to buffer pool for the table space.

CLI **tbspAvgPoolReadTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

A high average time might indicate the existence of an I/O conflict. In this case, you might need to move data to a different device.

The returned value includes the time spent in asynchronous read operations that are performed by prefetchers.

tbsp: Average pool write time (ms)

Description

Returns the average time (in milliseconds) for processing write requests that caused data or index pages to be physically written from buffer pool to disk for the table space.

CLI **tbspAvgPoolWriteTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

A high average time might indicate the existence of an I/O conflict. In this case, you might need to move data to a different device.

The returned value includes the time spent in asynchronous write operations that are performed by page cleaners.

tbsp: Average synchronous data read time (ms)

Description

Returns the average time (in milliseconds) for synchronous data reads for the table space.

CLI **tbspAvgSyncReadTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to analyze the I/O work being performed for the table space. Synchronous read operations are performed by database manager agents. Asynchronous reads are performed by prefetchers, which read data pages from disk into the buffer pool in anticipation of their use.

If the returned value is high compared to the average time spent in asynchronous data reads, you might want to increase the amount of prefetching by increasing the num_ioservers configuration parameter or turn on the seqdetect configuration parameter. For more information, see “db: num_ioservers” on page 125, “db: seqdetect” on page 127, or the *DB2 Administration Guide* for the version of DB2 that you are using.

tbsp: Average synchronous data write time (ms)

Description

Returns the average time (in milliseconds) for synchronous data writes for the table space.

CLI **tbspAvgSyncWriteTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to analyze the I/O work being performed for the table space. Synchronous write operations are performed by database manager agents. Asynchronous writes are performed by page cleaners, which write out changed pages to disk and free up space in the buffer pool.

If the returned value is high compared to the average time spent in asynchronous data writes, you might want to increase the number of page cleaners by using the num_iocleaners configuration parameter. For more information, see “db: num_iocleaners” on page 124 or the *DB2 Administration Guide* for the version of DB2 that you are using.

tbsp: Average synchronous I/O time (ms)

Description

Returns the average time (in milliseconds) for synchronous I/O operations for the table space.

CLI **tbspAvgSyncIoTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to analyze the I/O work being performed for the table space. Synchronous I/O operations are performed by database manager agents. Asynchronous I/O operations are performed by prefetchers (reads) and page-cleaners (writes). In general, asynchronous I/O helps your applications run faster.

tbsp: Buffer pool async data reads

Description

Returns the number of 4 KB data pages read asynchronously into the buffer pool for the table space.

CLI **tbspPoolAsyncDataReads** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Compare the returned value with the value returned by the "tbsp: # of synchronous reads" monitoring source (page 96) to gain insight into how well the prefetchers are working and to help you tune the num_ioservers configuration parameter. For more information on this parameter and prefetchers, see "db: num_ioservers" on page 125 or the *DB2 Administration Guide* for the version of DB2 that you are using.

tbsp: Buffer pool async data writes

Description
Returns the number of times a buffer pool data page was written asynchronously to disk for the table space.

CLI **tbspPoolAsyncDataWrites** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Compare the returned value with the value returned by the "tbsp: # of synchronous writes" monitoring source (page 96) to gain insight into how well the page cleaners are working and to help you tune the num_iocleaners configuration parameter. For more information on this parameter and page cleaners, see "db: num_iocleaners" on page 124 or the *DB2 Administration Guide* for the version of DB2 that you are using.

tbsp: Buffer pool async index reads (4 KB pages)

Description
Returns the number of 4 KB index pages read asynchronously into the buffer pool by a prefetcher within the table space.

CLI **tbspPoolAsyncIndexReads** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version
This monitoring source is supported for DB2 UDB and DB2 Enterprise-Extended Edition.

Usage notes
By comparing the ratio of asynchronous to synchronous reads (see "tbsp: # of synchronous index reads (4 KB pages)" on page 95), you can determine how well the prefetchers are working. This ratio can be helpful when you

are tuning the num_ioservers configuration parameters (see “db: num_ioservers” on page 125). Asynchronous reads are performed by database manager prefetchers.

For more information, see the *DB2 Administration Guide* for the version of DB2 that you are using.

tbsp: Buffer pool async index writes

Description

Returns the number of times a buffer pool index page was written asynchronously to disk for the table space.

CLI **tbspPoolAsyncIndexWrites** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Subtract the returned value from the value returned by the “tbsp: Buffer pool index writes” monitoring source (page 107) to calculate the number of synchronous index writes. By comparing the number of asynchronous index writes to synchronous index writes, you can gain insight into how well the buffer pool page cleaners are performing so you can tune the num_iocleaners configuration parameter.

By comparing the ratio of asynchronous to synchronous writes (see “tbsp: # of synchronous index writes (4 KB pages)” on page 95), you can determine how well the buffer pool page cleaners are performing. This ratio can be helpful when you are tuning the num_iocleaners configuration parameter.

For more information on this parameter and page cleaners, see “db: num_iocleaners” on page 124 or the *DB2 Administration Guide* for the version of DB2 that you are using.

tbsp: Buffer pool async read time (ms)

Description

Returns the time (in milliseconds) that database manager prefetchers spent reading data into the buffer pool for the table space.

CLI **tbspPoolAsyncReadTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Compare the returned value to the value returned by the “tbsp: Synchronous read time (ms)” monitoring source (page 112) to understand where I/O time is being spent.

tbsp: Buffer pool async write time (ms)

Description

Returns the time (in milliseconds) that database manager page cleaners spent writing data or index pages from the buffer pool to disk for the table space.

CLI **tbspPoolAsyncWriteTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Compare the returned value to the value returned by the “tbsp: Synchronous write time (ms)” monitoring source (page 112) to understand where I/O time is being spent.

tbsp: Buffer pool data logical reads

Description

Returns the number of logical read requests for data pages that went through the buffer pool for the table space.

CLI **tbspPoolDataLReads** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used to calculate the return value for the “tbsp: Percent buffer pool hit ratio, data+index” monitoring source (page 109).

The returned value includes requests for data that is:

- Already in the buffer pool.
- Read from disk into the buffer pool to fulfill the request. (The count for these is given by the “tbsp: Buffer pool data physical reads” monitoring source.)

tbsp: Buffer pool data pages copied from extended storage (4 KB pages)

Description

Returns the number of 4 KB buffer pool data pages copied from extended storage within the table space.

CLI **tbspDataPagesCopiedfromExtendedStorage** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

Required pages are copied from extended storage to the buffer pool if they are not in the buffer pool but are in extended storage. This copying might incur the cost of connecting to the shared memory segment but saves the cost of a disk read.

tbsp: Buffer pool data pages copied to extended storage (4 KB pages)

Description

Returns the number of 4 KB buffer pool data pages copied to extended storage for the table space.

CLI **tbspDataPagesCopiedtoExtendedStorage** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

Pages are copied from the buffer pool to extended storage when they are selected as victim pages. This copying is required to make space for new pages in the buffer pool.

tbsp: Buffer pool data physical reads

Description

Returns the number of read requests requiring I/O to get data pages into the buffer pool for the table space.

CLI **tbspPoolDataPReads** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used to calculate the return values for the “tbsp: Percent buffer pool hit ratio, data+index” and the “tbsp: # of synchronous reads” monitoring sources (see pages 109 and 96).

tbsp: Buffer pool data writes

Description

Returns the number of times that a buffer pool data page was physically written to disk for the table space.

CLI **tbspPoolDataWrites** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

If the returned value is high compared to the value returned by the “tbsp: Buffer pool data physical reads” monitoring source (page 104), you might be able to improve performance by increasing the available buffer pool space. You can increase available buffer pool space by:

- Freeing space more often by writing out changed pages. You can have space freed more often by increasing the chngpgs_thresh or num_iocleaners configuration parameter (see “db: num_iocleaners” on page 124).
- Increasing the total space for the buffer pool. You can increase the buffer pool space by increasing the buffpage configuration parameter.

tbsp: Buffer pool I/Os per second

Description

Returns the rate (per second) for buffer pool I/O for the table space. For this monitoring source, buffer pool I/O includes all physical data and index pages that go through the buffer pool when read or written.

CLI **tbspPoolIoRate** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine how effective your data storage device is. A low value can indicate the presence of an I/O wait, which might indicate that you should move data to a different device.

tbsp: Buffer pool index logical reads

Description

Returns the number of logical read requests for index pages that went through the buffer pool for the table space.

CLI **tbspPoolIndexLReads** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculating the returned values for the “tbsp: Percent buffer pool hit ratio, index” and “tbsp: Percent buffer pool hit ratio, data+index” monitoring sources (see pages 110 and 109).

The returned value includes requests for index pages that are:

- Already in the buffer pool.
- Read from disk into the buffer pool to fulfill the request. (The count for these is given by the “tbsp: Buffer pool index physical reads” monitoring source.)

tbsp: Buffer pool index pages copied from extended storage (4 KB pages)

Description

Returns the number of 4 KB buffer pool index pages copied from extended storage for the table space.

CLI **tbspIndexPagesCopiedfromExtendedStorage** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

Required index pages are copied from extended storage to the buffer pool if they are not in the buffer pool but are in extended storage. This copying might incur the cost of connecting to the shared memory segment but saves the cost of a disk read.

tbsp: Buffer pool index pages copied to extended storage (4 KB pages)

Description

Returns the number of 4 KB buffer pool index pages copied to extended storage within the table space during the monitoring interval.

CLI **tbspIndexPagesCopiedtoExtendedStorage**

DB2 version

This monitoring source is supported for DB2 Enterprise-Extended Edition only.

Usage notes

Pages are copied from the buffer pool to extended storage when they are selected as victim pages. This copying is required to make space for new pages in the buffer pool.

tbsp: Buffer pool index physical reads

Description

Returns the number of physical read requests to get index pages into the buffer pool for the table space.

CLI **tbspPoolIndexPReads** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used to calculate the return values for other buffer pool monitoring sources, for example, the “tbsp: Percent buffer pool hit ratio, data+index” monitoring source (page 109).

tbsp: Buffer pool index writes

Description

Returns the number of times that a buffer pool index page was physically written to disk for the table space.

CLI **tbspPoolIndexWrites** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

If the returned value is high compared to the value returned by the “tbsp: Buffer pool index physical reads” monitoring source (page 107), you might be able to improve performance by increasing the available buffer pool space. You can increase available buffer pool space by:

- Freeing space more often by writing out changed pages. You can have space freed more often by increasing the chngpgs_thresh or num_iocleaners configuration parameters (see “db: num_iocleaners” on page 124).
- Increasing the total space for the buffer pool. You can increase the buffer pool space by increasing the buffpage configuration parameter.

tbsp: Direct read time

Description

Returns the time (in milliseconds) for performing the direct reads for the table space.

CLI **tbspDirectReadTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculations for the “tbsp: Average direct read time (ms)” monitoring source (page 97). A high average time might indicate an I/O conflict.

tbsp: Direct write time (ms)

Description

Returns the time (in milliseconds) for performing the direct writes for the table space.

CLI **tbspDirectWriteTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculations for the “tbsp: Average direct write time (ms)” monitoring source (page 98). A high average time might indicate an I/O conflict.

tbsp: Extended storage read/write ratio

Description

Returns the ratio (as a percent) of pages (data+index) copied from extended storage to pages coped to extended storage within the table space.

CLI **tbspExtendedStorageReadWriteRatio**

DB2 version

This monitoring source is supported for DB2 UDB and DB2 Enterprise-Extended Edition.

Usage notes

When a page is transferred from extended storage to the buffer pool, you save a system I/O call. However, you still incur the cost of attaching to the extended memory segment, copying the page, and detaching from the segment.

Use the returned value to determine if you would benefit from using extended storage. The higher the ratio, the more likely you are to benefit. In general, extended storage is particularly useful if I/O activity is very high on your system.

For information on related monitoring sources, see “tbsp: Buffer pool data pages copied from extended storage (4 KB pages)” on page 104, “tbsp: Buffer pool data pages copied to extended storage (4 KB pages)” on page 104 , “tbsp: Buffer pool index pages copied from extended storage (4 KB pages)” on page 106, and “tbsp: Buffer pool index pages copied to extended storage (4 KB pages)” on page 106.

For more information on extended storage, see the *DB2 System Monitor Guide and Reference* for the version of DB2 that you are using.

tbsp: File closed

Description

Returns the total number of files for the table space that are closed.

CLI **tbspFilesClosed** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you determine the best value for the maxfilop configuration parameter (for more information on this parameter, see the *DB2 Administration Guide* for the version of DB2 that you are using).

The database manager opens files for reading and writing into and out of the buffer pool. The maximum number of database files that are open by an application at any time is controlled by the maxfilop parameter. If the maximum is reached, one file must close before a new file opens, which can slow down performance.

tbsp: Percent buffer pool hit ratio, data+index

Description

Returns the overall buffer pool hit ratio (as a percent) for the table space during the monitoring interval. This hit ratio includes both index and data page activity.

CLI **tbspPoolHitRatio** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The overall buffer pool hit ratio indicates the percentage of requests for which the database manager did not need to load a page from disk to service. That is, the page was already in the buffer pool. The greater the buffer pool hit ratio, the lower the frequency of disk I/O.

If the hit ratio is low (for example, a ratio of zero indicates that pages needed to be read in for every request), increasing the number of buffer pool pages might improve performance. For information on increasing the buffer pool size using the buffpage configuration parameter, see the *DB2 Administration Guide* for the version of DB2 that you are using. See “db: buffpage” on page 117 for information on monitoring to tune the buffpage parameter.

For a large table space, increasing the buffer pool size might have a minimal effect on the buffer pool hit ratio. Such a table space can have so large a number of data pages, that the statistical chance of a hit is not increased by increasing buffpage. However, even though the data might be too large to fit in the buffer pool, the entire index might fit. In this case, you could tune buffpage until the overall buffer pool hit ratio stops increasing, then tune buffpage until the buffer pool index hit ratio (returned by the “tbsp: Percent buffer pool hit ratio, index” monitoring source) no longer increases.

See “tbsp: Percent buffer pool hit ratio, index” for more information.

tbsp: Percent buffer pool hit ratio, index

Description

Returns the table space’s index page hit ratio (as a percent) for the buffer pool during the monitoring interval.

CLI **tbspIndxPlHitRatio** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The index page hit ratio for the buffer pool indicates the percent of index page requests that the database manager did not need to load an index page from disk to service. That is, the index page was already in the buffer pool. The greater the returned value, the lower the frequency of disk I/O, and the faster the performance.

If the hit ratio is low, increasing the number of buffer pool pages might improve performance. For information on increasing the buffer pool size using the buffpage configuration parameter, see the *DB2 Administration Guide* for the version of DB2 that you are using.

tbsp: Percent prefetch satisfied

Description

Returns the percentage of asynchronous read requests that were satisfied for a table space during the last monitoring interval.

CLI **tbspPctAsyDatRdRqS** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine the amount of asynchronous I/O done in each interaction with the prefetcher. See “db: num_ioservers” on page 125 for more information about prefetchers.

tbsp: Percent total I/O

Description
Returns the I/O operations that occurred during the monitoring interval for the table space as a percentage of the total I/O operations for the database that contains the table space.

CLI **tbspPctTotalIo** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine which table spaces in a particular database account for the most I/O operations. This information can help you determine which of your devices to use for each table space.

tbsp: Prefetch requests in an interval

Description
Returns the number of asynchronous read requests for the table space during the monitoring interval.

CLI **tbspPlAsyncDatRdRq** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is used in calculations for the “tbsp: Percent prefetch satisfied” monitoring source (page 110). This value can help you determine the amount of asynchronous I/O done in each interaction with the prefetcher. For more information on prefetchers, see “db: num_ioservers” on page 125 .

tbsp: Synchronous read time (ms)

Description

Returns the time (in milliseconds) spent in synchronous reads for the table space.

CLI **tbspSyncReadTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Compare the returned value to the value returned by the “tbsp: Buffer pool async read time (ms)” monitoring source (page 102) to understand where I/O time for this table space is spent.

tbsp: Synchronous write time (ms)

Description

Returns the time (in milliseconds) spent in synchronous writes to disk for the table space.

CLI **tbspSyncWriteTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Compare the returned value to the value returned by the “tbsp: Buffer pool async write time (ms)” monitoring source (page 103) to understand where I/O time for this table space is spent.

tbsp: Total direct I/O time (ms)

Description

Returns the total time (in milliseconds) for direct reads to and writes from the table space.

CLI **tbspTotalDirectIoTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value indicates the amount of time spent in direct reads and writes to and from this table space. A high time might indicate the presence of an I/O conflict.

tbsp: Total pool physical I/O time (ms)

Description

Returns the total time (in milliseconds) for physical buffer pool I/O operations (including both synchronous and asynchronous reads and writes) for the table space.

CLI **tbspTotalPoolIoTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

A high returned value (as compared to the total number of physical buffer pool I/O operations) might indicate the presence of an I/O wait, which could indicate that you should move data to a different device.

tbsp: Total pool physical read time (ms)

Description

Returns the total time (in milliseconds) for processing read requests that caused data or index pages to be physically read from disk to the buffer pool for the table space.

CLI **tbspTotalReadTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used to calculate the average pool read time, which is returned by the “tbsp: Average pool read time (ms)” monitoring source (page 98). This average can indicate the presence of an I/O wait, which in turn might indicate that you should move data to a different device.

tbsp: Total pool physical write time (ms)

Description

Returns the total time (in milliseconds) for buffer pool physical writes (including asynchronous writes) for the table space.

CLI **tbspTotalWriteTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is used to calculate the average pool write time, which is returned by the "tbsp: Average pool write time (ms)" monitoring source (page 99). This average can indicate the presence of an I/O wait, which in turn might indicate that you should move data for this table space to a different device.

tbsp: Total synchronous I/O

Description
Returns the total number of synchronous reads and writes for both data and index pages for the table space.

CLI **tbspTotalSyncIos** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to gain insight into how well the prefetchers and page cleaners are working and to help you tune the num_ioservers and num_iocleaners configuration parameters. For more information on these parameters, see "db: num_ioservers" on page 125, "db: num_iocleaners" on page 124 , or the *DB2 Administration Guide* for the version of DB2 that you are using.

Because synchronous I/O operations are performed by database agents, a high number of them can slow performance.

tbsp: Total synchronous I/O time (ms)

Description
Returns the total time (in milliseconds) for processing requests for synchronous reads or writes within the table space.

CLI **tbspTotalSyncIoTime** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is the sum of the returned values from the tbsp: "Synchronous read time (ms)" monitoring source (page 112) and "tbsp:

Synchronous write time (ms)" monitoring source (see page 112). This time is the amount of time that database agents spend doing synchronous reads and writes for the table space.

DB2 Configuration Monitoring Collection

Collection name:
DB2_Config_Monitors

The monitoring sources in this collection provide information about the configuration parameters for database managers and their databases. Typically, you should tune the values of these parameters to achieve maximum performance for your database environment. For descriptions of these configuration parameters, their default values, and how you can tune them to improve performance of your instances and databases, see the *DB2 Administration Guide*for the version of DB2 that you are using.

Table 31 lists the monitoring sources in this collection.

Table 31. Monitoring Sources in the DB2 Configuration Monitoring Collection

GUI Name	CLI Name	Returned Value Format
db: app_ctl_heap_sz	dbAppCtlHeapSz	Numeric: 4 KB pages
db: applheapsz	dbApplheapsz	Numeric: 4 KB pages
db: avg_appls	dbAvgAppls	Numeric
db: buffpage	dbBuffpage	Numeric
db: catalogcache_sz	dbCatalogcacheSz	Numeric: 4 KB pages
db: chngpgs_thresh	dbChngpgsThresh	Numeric
db: dbheap	dbDbheap	Numeric: 4 KB pages
db: locklist	dbLocklist	Numeric: 4 KB pages
db: logbufsz	dbLogbufsz	Numeric: 4 KB pages
db: logprimary	dbLogprimary	Numeric
db: maxappls	dbMaxappls	Numeric
db: maxlocks	dbMaxlocks	Percentage
db: mincommit	dbMincommit	Numeric
db: newlogpath	dbNewlogpath	Numeric
db: num_iocleaners	dbNumIocleaners	Numeric
db: num_ioservers	dbNumIoservers	Numeric
db: pckcachesz	dbPckcachesz	Numeric: 4 KB pages
db: Percent space used in database heap	dbPctHeapSpaceUsed	Percentage
db: restore_pending	dbRestorePending	String: ″NO″ or ″YES″
db: seqdetect	dbSeqdetect	Numeric
db: sortheap	dbSortheap	Numeric: 4 KB pages
dbms: agentpri	dbmsAgentpri	Numeric

Table 31. Monitoring Sources in the DB2 Configuration Monitoring Collection (continued)

GUI Name	CLI Name	Returned Value Format
dbms: aslheapsz	dbmsAslheapsz	Numeric: 4 KB pages
dbms: fcm_num_anchors	dbmsFcmNumAnchors	Numeric
dbms: fcm_num_buffers	dbmsFcmNumBuffers	Numeric: 4 KB pages
dbms: fcm_num_connect	dbmsFcmNumConnect	Numeric
dbms: fcm_num_rqb	dbmsFcmNumRqb	Numeric
dbms: max_agents	dbmsMaxAgents	Numeric
dbms: max_coordagents	dbmsMaxCoordagents	Numeric
dbms: maxcagents	dbmsMaxcagents	Numeric
dbms: Maximum idle agents in the agent pool	dbmsMaxIdlePoolAgents	Numeric
dbms: Minimum committed memory	dbmsMinCommPrivMem	Numeric: 4 KB pages
dbms: mon_heap_sz	dbmsMonHeapSz	Numeric: 4 KB pages
dbms: num_poolagents	dbmsNumPoolagents	Numeric
dbms: Percent private memory used	dbmsPctPrivMemUsed	Percentage
dbms: query_heap_sz	dbmsQueryHeapSz	Numeric: 4 KB pages
dbms: rqrioblk	dbmsRqrioblk	Numeric
dbms: sheapthres	dbmsSheapthres	Numeric: 4 KB pages
dbms: sqlstmtsz	dbmsSqlstmtsz	Numeric

db: app_ctl_heap_sz

Description

Returns the maximum size (in 4 KB pages) for the application control heap in the database during the monitoring interval.

CLI **dbApplCtlHeapSz**

DB2 version

This monitoring source is supported by DB2 UDB only.

Range for parameter

1 ≤ returned value ≤ 64000

Usage notes

The heap is required to share information among agents working on behalf of the same application at a node in a *massively parallel processing* (MPP) or a *symmetric multiprocessor* (SMP) system. If complex applications are being run or the MPP configuration has a large number of nodes, you should increase the size of this heap.

Start with the default value. You might have to set the value higher if you are running complex applications or if you have a system that contains a large number of database partitions.

In a partitioned database environment, this heap is used to store copies of the executing section of SQL statements for agents and subagents. Symmetric multiprocessor agents (SMP), subagents, and agents in all other environments, however, use applheapsz (see “db: applheapsz” on page 117).

db: applheapsz

Description
Returns the size (in 4 KB pages) of the application heap that is available for each individual agent in the database during the monitoring interval.

CLI **dbApplheapsz**

Range for parameter
16 ≤ returned value ≤ 60000

DB2 version
This monitoring source is supported for DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes
Increase the value of the parameter if your application receives an error indicating that there is not enough storage in the application heap.

The heap is allocated when an agent or subagent is initialized for an application. The amount allocated will be the minimum amount needed to process the request given to the agent or subagent. When the agent or subagent requires more heap space to process larger SQL statements, the database manager will allocate memory as needed, up to the maximum specified by the parameter.

Use the results from the “db: app_ctl_heap_sz” monitoring source to help you tune the applheapsz parameter.

db: avg_appls

Description
Returns the value of the average number of active applications (avg_appls) configuration parameter.

CLI **dbAvgAppls**

Range for parameter
1 ≤ returned value ≤ *maxappls*, where *maxappls* is the value of the maxappls configuration parameter.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
This value is used by the SQL optimizer to estimate how much of the buffer pool will be available at run-time for the SQL query access plan chosen. Increasing this parameter can influence the optimizer to choose an access plan that will be more conservative in its buffer pool usage.

db: buffpage

Description
Returns the value of the buffer pool size (buffpage) configuration parameter.

CLI **dbBuffpage**

Range for parameter
(2 × *maxappls*) ≤ returned value ≤ 524288 (in 4 KB pages), where *maxappls* is the value of the maxappls configuration parameter.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This value is the size of the buffer pool. This parameter is the most important parameter affecting database performance. There is one buffer pool per database, and it is available to all applications using the database. Its memory is allocated on the machine where the database is located. The size of the buffer pool is used by the SQL optimizer in determining SQL query access plans. If the buffer pool is large enough to keep the data required by applications in memory, less disk I/O activity occurs, which results in better performance.

Use the results from the monitoring sources in the DB2 buffer pool I/O monitoring collection to help you tune the buffpage parameter (see “DB2 Buffer Pool I/O Monitoring Collection” on page 47.)

db: catalogcache_sz

Description

Returns the value of the catalog cache size (catalogcache_sz) configuration parameter.

CLI **dbCatalogcacheSz**

Range for parameter

1 ≤ returned value ≤ *dbheap*, where *dbheap* is the value of the dbheap configuration parameter.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This value is the maximum amount of space that the catalog cache can use from the database heap (dbheap). The catalog cache is referenced whenever a table, view, or alias name is processed during the compilation of an SQL statement. It is dynamically allocated from dbheap, as required, until the catalogcache_sz is reached.

Use the results from the following monitoring sources to help you tune the catalogcache_sz parameter:

- “db: # of catalog cache heap full” on page 19
- “db: # of catalog cache inserts” on page 19
- “db: # of catalog cache lookups” on page 20
- “db: # of catalog cache overflows” on page 20
- “db: Percent catalog cache hit ratio” on page 81
- “db: Percent DDL SQL” on page 191

db: chngpgs_thresh

Description

Returns the value of the changed pages threshold (chngpgs_thresh) configuration parameter.

CLI **dbChngpgsThresh**

Range for parameter

5 ≤ returned value ≤ 80%

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This value sets a limit on how much buffer pool space can be occupied by changed pages before the asynchronous page cleaners will be started, if they are not currently active.

Asynchronous page cleaners write changed pages from the buffer pool to disk before the space in the buffer pool is required by a database agent. This means that the agents do not need to wait for a changed page to be written out before being able to read a page, and application transactions run faster.

Use the results from the following monitoring source to help you tune the chngpgs_thresh parameter:

- “db: # of cleaners due to dirty threshold” on page 60

db: dbheap

Description

Returns the value of the database heap (dbheap) configuration parameter.

CLI **dbDbheap**

Range for parameter

32 ≤ returned value ≤ 60000 (in 4 KB pages)

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This value is the maximum amount of memory allowed for a database heap. There is one database heap per database. It is used on behalf of all applications connected to the database.

Tuning dbheap has minimal impact on performance. The main function of this parameter is to prevent the database manager from allocating an excessive amount of space for a particular database.

Use the results from the following monitoring sources to help you tune the dbheap parameter:

- “db: # of catalog cache heap full” on page 19
- “db: Max database heap allocated” on page 29
- “db: Percent space used in database heap” on page 126

db: locklist

Description

Returns the value of the maximum storage for lock lists (locklist) configuration parameter.

CLI **dbLocklist**

Range for parameter
4 ≤ returned value ≤ 60000 (in 4 KB pages)

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
This value is the amount of storage that is allocated to the lock list. There is one lock list per database, and it contains the locks held by all applications concurrently connected to the database. Too small a value can lead to excessive lock waits. Too high a value can deprive the system of resources or memory.

Use the results from the following monitoring sources to help you tune the locklist parameter:

- “conn: Lock escalations during last interval” on page 145
- “conn: Percent locklist space used by application” on page 17
- “conn: Lock wait time (s)” on page 146
- “conn: Lock waits” on page 146
- “conn: Locks held” on page 147
- “db: Avg lock escalations per connection” on page 27
- “db: Lock list in use (bytes)” on page 150
- “db: Lock waits” on page 150
- “db: Locks held” on page 151
- “db: Time waited on locks” on page 152
- “db: Total exclusive lock escalations” on page 153
- “db: Percent locklist space used by database” on page 32

db: logbufsz

Description
Returns the value of the log buffer size (logbufsz) configuration parameter.

CLI **dbLogbufsz**

Range for parameter
4 ≤ returned value ≤ 128 (in 4 KB pages)

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
This value specifies the amount of the database heap (dbheap) to use as a buffer for log records before writing these records to disk. It is important that the log buffer be able to hold the amount of log space used by an average transaction. Otherwise, logging performance will decrease and slow the overall system.

Use the results from the following monitoring sources to help you tune the logbufsz parameter:

- “db: Max primary log space used” on page 30

- “db: # of secondary logs allocated currently” on page 27
- “db: Max secondary logs space used” on page 30
- “db: Max total log space used” on page 30
- “db: Primary log space allocated” on page 35
- “db: Percent used in primary log” on page 33
- “db: Percent used in secondary log” on page 34

db: logprimary

Description

Returns the value of the number of primary log files (logprimary) configuration parameter.

CLI **dbLogprimary**

Range for parameter

12 ≤ returned value ≤ 128

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This value is the number of primary log files to be preallocated by the database manager. If you find that secondary log files are frequently being allocated, you might be able to improve performance by increasing the value of logprimary.

Use the results from the following monitoring sources to help you tune the logprimary parameter:

- “db: Max primary log space used” on page 30
- “db: # of secondary logs allocated currently” on page 27
- “db: Max secondary logs space used” on page 30
- “db: Max total log space used” on page 30
- “db: Primary log space allocated” on page 35
- “db: Percent used in primary log” on page 33
- “db: Percent used in secondary log” on page 34

db: maxappls

Description

Returns the value of the maximum number of active applications (maxappls) configuration parameter.

CLI **dbMaxappls**

Range for parameter

- 1 ≤ returned value ≤ 5000 (database on UNIX)
- 1 ≤ returned value ≤ 1500 (database on NT)

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This value is the maximum number of concurrent applications that can be connected (both local and remote) to a database. Because each application that attaches to a database causes some private memory to be allocated, allowing a large number of concurrent applications will potentially use more memory.

Increasing the value of this parameter without lowering the maxlocks parameter or increasing the locklist parameter might cause you to reach the database limit on locks (locklist) rather than the application limit. The result can be pervasive lock escalation problems.

Use the results from the following monitoring sources to help you tune the maxappls parameter:

- “db: avg_appls” on page 117
- “db: buffpage” on page 117
- “db: locklist” on page 119
- “dbms: Maximum agents registered” on page 11
- “db: Percent connections used” on page 31

db: maxlocks

Description

Returns the value of the maximum percentage of lock list before escalation (maxlocks) configuration parameter.

CLI **dbMaxlocks**

Range for parameter

1 ≤ returned value ≤ 100%

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This value specifies the percentage of the lock list that an application that can hold before the database manager performs lock escalation. Lock escalation can increase contention, which reduces system throughput and increases user response time.

The values for maxlocks and maxappls must satisfy (maxlocks × maxappls) greater than or equal to 100, and each lock uses 32 bytes. It is recommended that you rebind application packages after changing this parameter.

Use the results from the following monitoring sources to help you tune the maxlocks parameter:

- “conn: Lock escalations during last interval” on page 145
- “db: Lock escalations during last interval” on page 149
- “db: locklist” on page 119
- “db: maxappls” on page 121
- “db: Total exclusive lock escalations” on page 153

db: mincommit

Description
Returns the value of the number of commits to group (mincommit) configuration parameter.

CLI **dbMincommit**

Range for parameter
1 ≤ returned value ≤ 25

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
This parameter allows you to delay the writing of log records to disk until a minimum number of commits have been performed. This delay can help reduce the overhead associated with writing log records and improve performance.

The default value for mincommit is 1, which might be too low for your environment. By sampling the number of transactions per second throughout the day, you can determine the peak per second rate and adjust mincommit to accommodate all or some large percent of it. This adjustment would minimize the number of log writes under the heaviest conditions.

You can determine the peak number of transactions per second by adding the values returned from these two monitoring sources:

- “db: Committed statements per second” on page 190
- “db: Rollbacks per second” on page 192

As you increase mincommit, you might also need to increase the log buffer size (logbufsz) to avoid filling the log buffer. Filling the log buffer also forces the writing of log records to disk.

If you change mincommit, you also need to change the value for the logbufsz configuration parameter.

Use the results from the following monitoring sources to help you tune the mincommit parameter:

- “db: # of commit statements attempted” on page 21
- “db: # of internal commits” on page 22
- “db: Committed statements per second” on page 190
- “db: Percent DDL SQL” on page 191

db: newlogpath

Description
Returns the current value of the newlogpath configuration parameter.

CLI **dbNewlogpath**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

You use the newlogpath configuration parameter to specify a new location for the log files. The specified path does not become the current log path until both:

- The database is in a consistent state.
- All users are disconnected from the database.

When the first new connection is made to the database, the database manager moves the logs to this location.

db: num_iocleaners

Description

Returns the current value of the number of asynchronous page cleaners (num_iocleaners) configuration parameter.

CLI **dbNumIocleaners**

Range for parameter

0 ≤ returned value ≤ 255

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This parameter specifies the number of asynchronous page cleaners for a database.

Page cleaners monitor the buffer pool and asynchronously write out changed pages to disk to free space in the buffer pool. Page cleaners are invoked when:

- More space in the buffer pool is needed for a new page.
- The percentage of space occupied by changed pages in the buffer pool has exceeded the value specified for the chngpgs_thresh configuration parameter.
- The size of the log that would need to be read during database recovery would exceed the value given by the following formula: logprimary × logfilsiz × softmax, where logprimary, logfilsiz, and softmax are database configuration parameters for logging.

Page cleaners perform I/O that would otherwise need to be performed by the database agents. As a result, applications can run faster.

Use the following guidelines to help you tune this configuration parameter:

- The num_iocleaners parameter can be reduced if both of the following conditions are true:
  - The returned value from the “db: Buffer pool data writes” monitoring source is approximately equal to the returned value from the “db: Buffer pool async data writes” monitoring source.
  - The returned value from the “db: Buffer pool index writes” monitoring source is approximately equal to the returned value from the “db: Buffer pool async index writes” monitoring source.
- The num_iocleaners parameter should be increased if either of the following conditions are true:

- The returned value from the “db: Buffer pool data writes” monitoring source is much greater than the returned value from the “db: Buffer pool async data writes” monitoring source.
- The returned value from the “db: Buffer pool index writes” monitoring source is much greater than the returned value from the “db: Buffer pool async index writes” monitoring source.

For descriptions of these monitoring sources, see:

- “db: Buffer pool data writes” on page 76
- “db: Buffer pool async data writes” on page 73
- “db: Buffer pool index writes” on page 77
- “db: Buffer pool async index writes” on page 73

You can also use the results from the following monitoring sources to help you tune the num_iocleaners parameter:

- “db: # of synchronous I/O” on page 65
- “db: # of synchronous writes” on page 66
- “db: # of victim page cleaners invoked” on page 67
- “db: Avg # of pages for each cleaner” on page 72

db: num_ioservers

Description

Returns the current value of the number of I/O servers (num_ioservers) configuration parameter.

CLI **dbNumIoservers**

Range for parameter

1 ≤ returned value ≤ 255

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This value specifies the number of I/O servers for a database. I/O servers are used on behalf of the database agents to perform asynchronous I/O for utilities such as backup and restore, and to perform prefetch I/O (in which case, they are called prefetchers).

Prefetchers read pages from disk into the buffer pool in anticipation of their use. In most situations, these pages are read just before they are needed. However, prefetchers can cause unnecessary I/O by reading pages into the buffer pool that might not be used. For example, an application starts reading through a table and prefetchers read consecutive pages into the buffer pool before the pages are required by the application. But then, the application fills an application buffer and stops reading. Meanwhile, the prefetchers already have performed the I/O for additional pages and the buffer pool is partially taken up with those pages.

To fully exploit all the I/O devices in the system, a good value for num_ioservers to use is generally one or two more than the number of physical devices on which the database resides.

DB2 Configuration Monitoring Collection

You can also use the results from the following monitoring sources to help you tune the num_ioservers parameter:

- “db: # of synchronous I/O” on page 65
- “db: # of synchronous reads” on page 66
- “db: Buffer pool async data reads (4 KB pages)” on page 72

db: pckcachesz

Description

Returns the current value of the package cache size (pckcachesz) configuration parameter.

CLI **dbPckcachesz**

Range for parameter

1 ≤ returned value ≤ *applheapsz* (in 4 KB pages), where *applheapsz* is the value of the applheapsz configuration parameter.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This value controls the amount of application heap memory to be used for caching a package’s static and dynamic SQL statements.

You will need to experiment with the size of the package cache to find the optimal number of the pckcachesz parameter. For example, you might be able to use a smaller package cache size if there is no increase in the number of package cache inserts when you decrease the size of the cache. Decreasing the package cache size frees up system resources for other work. However, increasing the package cache size might improve overall system performance if it results in a decrease of package cache inserts.

Use the results from the following monitoring sources to help you tune the pckcachesz parameter:

- “conn: Percent package cache hit ratio” on page 58
- “db: Percent package cache hit ratio” on page 32

db: Percent space used in database heap

Description

Returns the percentage of maximum space used in the database heap.

CLI **dbPctHeapSpaceUsed**

Range for parameter

0 ≤ returned value ≤ 100%

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to evaluate your setting of the dbheap configuration parameter. A returned value of 100% indicates that an application has probably received an error indicating that there was not enough storage available. The dbheap parameter limits the amount of storage that can be allocated for database heap. For more information about the dbheap

parameter, see “db: dbheap” on page 119 or the *DB2 Administration Guide* for the version of DB2 that you are using.

Default thresholds and actions

Table 32. Default Thresholds and Actions for db: Percent space used in database heap

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 85%	Send Tivoli notice Change icon
warning	Greater than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: restore_pending

Description
Returns the RESTORE PENDING status in the database during the last monitoring interval.

CLI **dbRestorePending**

DB2 version
This monitoring source is supported by DB2 UDB only.

db: seqdetect

Description
Returns the current value of the sequential detection flag (seqdetect) configuration parameter.

CLI **dbSeqdetect**

Range for parameter
Either ″yes″ or ″no″, indicating whether the database manager should perform sequential detection.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The seqdetect parameter controls whether the database manager should perform sequential detection.

The database manager can monitor I/O. If sequential page reading is occurring, the database manager can activate I/O prefetching. This type of sequential prefetch is known as sequential detection.

If this configuration parameter is set to ″no″, prefetching takes place only if the database manager knows it will be useful (for example, in table sorts). For information on default prefetching and the default prefetch size

(dft_prefetch_sz) configuration parameter, see the *DB2 Administration Guide* for the version of DB2 that you are using.

db: sortheap

Description

Returns the current value of the sort heap size (sortheap) configuration parameter.

CLI **dbSortheap**

Range for parameter

16 ≤ returned value ≤ 524288 (in 4 KB pages)

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This value is the maximum amount of memory that can be allocated as sort heap for each sort within a database. The sort heap is the memory block where data is sorted.

Use the following monitoring sources to help you tune the sortheap parameter:

- “db: Active sorts” on page 171
- “db: Percent sort overflowed” on page 172
- “db: Sort heap allocated (4 KB pages)” on page 172
- “db: Sort overflows” on page 173
- “db: Total sort time (ms)” on page 173
- “db: Total sorts” on page 173
- “dbms: Percent piped sorts rejected” on page 174
- “dbms: Percent sort heap allocated” on page 174
- “dbms: Sort heap allocated (4 KB pages)” on page 175

dbms: agentpri

Description

Returns the current value of the priority of agents (agentpri) configuration parameter.

CLI **dbmsAgentpri**

Range for parameter

A signed number

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This value is the priority that the operating system scheduler gives to agent and other database manager instance processes and threads. This priority determines how the operating system gives CPU time to the DB2 processes and threads relative to the other processes and threads running on the machine.

A value of −1 indicates that no special action is taken and the operating system schedules the database manager in the normal way that it schedules all processes and threads. Any other value indicates the database manager creates its processes and threads with a static priority set to this value.

For more information, see the *DB2 Administration Guide* for the version of DB2 that you are using.

dbms: aslheapsz

Description
Returns the current value of the application support layer heap size (aslheapsz) configuration parameter.

CLI **dbmsAslheapsz**

Range for parameter
1 ≤ returned value ≤ 524288 (in 4 KB pages)

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
This value is the amount of memory that is allocated for the application support layer heap. This heap is used as a communication buffer between the local application and its associated agent. In addition, this value is used to determine the I/O block size when a blocking cursor is opened.

dbms: fcm_num_anchors

Description
Returns the number of FCM message anchors for the DB2 instance during the monitoring interval.

CLI **dbmsFcmNumAnchors**

Range for parameter
−1 ≤ returned value ≤ (128 − *fcm_num_rqb*), where *fcm_num_rqb* is the value for the fcm_num_rqb configuration parameter.

DB2 version
This monitoring source is supported by DB2 Enterprise-Extended Edition only.

Usage notes
Agents use the message anchors to send messages among themselves.

Use the results from the following monitoring sources and parameters to help you tune the fcm_num_anchors parameter:

- “dbms: fcm_num_rqb” on page 130
- intra_parallel parameter (see the *DB2 Administration Guide* for the version of DB2 that you are using)

dbms: fcm_num_buffers

Description
Returns the number of 4 KB buffers that are used for internal

communications (messages) among the nodes and within the nodes in a DB2 instance during the monitoring interval.

CLI **dbmsFcmNumBuffers**

Range for parameter
128 ≤ returned value ≤ 65300

DB2 version
This monitoring source is supported by DB2 Enterprise-Extended Edition only.

Usage notes
You might need to increase the value of this parameter if you have:

- Multiple logical nodes on a processor
- Too many users, nodes, or complex applications that exceed the buffer limit

dbms: fcm_num_connect

Description
Returns the number of FCM connection entries for the DB2 instance during the monitoring interval.

CLI **dbmsFcmNumConnect**

Range for parameter
−1 ≤ returned value ≤ (128 − *fcm_num_rqb*), where *fcm_num_rqb* is the value of the fcm_num_rqb parameter.

DB2 version
This monitoring source is supported by DB2 Enterprise-Extended Edition only.

Usage notes
Agents use connection entries to pass data among themselves.

Use the results from the dbms: fcm_num_rqb monitoring source 130 to help you tune the fcm_num_connect parameter.

dbms: fcm_num_rqb

Description
Returns the number of FCM request blocks for the DB2 instance during the monitoring interval.

CLI **dbmsFcmNumRqb**

Range for parameter
128 ≤ returned value ≤ 120000

DB2 version
This monitoring source is supported by DB2 Enterprise-Extended Edition only.

Usage notes
Request blocks are the media through which information is passed between the FCM daemon and an agent.

The requirement for request blocks will vary according to the number of users on the system, the number of database partition servers in the system, and the complexity of queries that are run.

dbms: max_agents

Description

Returns the current value of the maximum number of agents (maxagents) configuration parameter.

CLI **dbmsMaxAgents**

Range for parameter

1 ≤ returned value ≤ 64000

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This value is the maximum number of database manager agents available at any given time to accept application requests. It limits the total number of applications that can connect to all databases in the DB2 instance at a given time.

The value of maxagents should be the sum of the values for maxappls in each database allowed to be accessed concurrently. Increasing maxagents can increase resource use because resources for each agent are allocated when the DB2 instance is started.

Use the results from the following monitoring sources to help you tune the maxagents parameter:

- “db: # of applications connected currently” on page 18
- “dbms: # of local connections” on page 35
- “dbms: # of remote connections” on page 37
- “dbms: Maximum agents registered” on page 11

dbms: max_coordagents

Description

Returns the maximum number of database manager coordinating agents that can exist on a server in a partitioned or nonpartitioned database environment.

CLI **dbmsMaxCoordagents**

Range for parameter

−1 ≤ returned value ≤ 0 − *maxagents* where *maxagents* is the value for the maxagents configuration parameter.

DB2 version

This monitoring source is supported for DB2 UDB and DB2 Enterprise-Extended Edition.

Usage notes

One coordinating agent is acquired for each local or remote application that connects to a database or attaches to an instance. Requests that require

an instance attachment include CREATE DATABASE, DROP DATABASE, and Database System Monitor commands.

Use the results from the following parameters and monitoring sources to help you tune the max_coordagents parameter:

- intra_parallel parameter (see the *DB2 Administration Guide* for the version of DB2 that you are using)
- "dbms: max_agents" on page 131
- "dbms: num_poolagents" on page 134
- min_priv_mem parameter (see the *DB2 Administration Guide* for the version of DB2 that you are using)
- num_initagents parameter (see the *DB2 Administration Guide* for the version of DB2 that you are using)

dbms: maxcagents

Description

Returns the maximum number of database manager coordinator agents that can be concurrently executing a database manager transaction in the DB2 instance during the monitoring interval.

CLI **dbmsMaxcagents**

Range for parameter

−1 ≤ returned value ≤ (1 − *maxagents*), where *maxagents* is the value for the maxagents configuration parameter.

DB2 version

This monitoring source is supported for DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

When this monitoring source is used with DB2 for common servers, the maxcagents value is the default. This value is the maximum number of database manager agents that can be concurrently executing a database manager transaction. Use the maxcagents parameter to control the load on the system during periods of high simultaneous application activity.

A value of −1 indicates that the limit is equal to the maximum number of agents (maxagents).

The maxcagents parameter does not limit the number of applications that can have connections to the database.

Use the results from the following monitoring sources to help you tune the maxcagents parameter:

- "dbms: # of local connections executing" on page 36
- "dbms: # of remote connections executing" on page 38
- "dbms: Agents waiting for a token" on page 10
- "dbms: Maximum agents waiting" on page 11
- "dbms: Percent agents waiting" on page 11

When this monitor is used with DB2 Enterprise-Extended Edition, the max_coordagent is the default. This parameter determines the maximum number of coordinating agents that can exist at one time on a node.

For additional information on the maxcagents parameter, see the *DB2 Administration Guide* for the version of DB2 that you are using.

dbms: Maximum idle agents in the agent pool

Description

Returns the maximum number of idle agents in the agent pool within the DB2 instance during the monitoring interval.

CLI **dbmsMaxIdlePoolAgents**

DB2 version

This monitoring source is supported for DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

When this monitoring source is used with DB2 for common servers, it returns the max_idleagents configuration parameter (see the *DB2 Administration Guide* for the version of DB2 that you are using) to control the maximum number of idle agents in the agent pool.

When this monitoring source is used with DB2 UDB and DB2 Enterprise-Extended Edition, it returns the max_coordagents configuration parameter (see “dbms: max_coordagents” on page 131) to control the maximum number of idled coordinator agents in the agent pool. The agent pool contains subagents and idle agents. The num_poolagents parameter (see “dbms: num_poolagents” on page 134) specifies the maximum number of subagents and idle agents allowed in the agent pool.

dbms: Minimum committed private memory

Description

Returns the current value of the minimum committed private memory (min_priv_mem) configuration parameter.

CLI **dbmsMinCommPrivMem**

Range for parameter

32 ≤ returned value ≤ 112000 (in 4 KB pages)

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This value is the number of pages that the database server process reserves as private virtual memory when a DB2 instance is started. If the server later requires more private memory, it tries to obtain more from the operating system at that time.

dbms: mon_heap_sz

Description

Returns the current value of the database system monitor heap size (mon_heap_sz) configuration parameter.

CLI **dbmsMonHeapSz**

Range for parameter
0 ≤ returned value ≤ 60000 (in 4 KB pages)

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
This value is the amount of memory that is allocated for database system monitor data. A value of zero prevents the database manager from collecting database system monitor data.

dbms: num_poolagents

Description
Returns the maximum number of agents allowed in the agent pool within the DB2 instance during the monitoring interval.

CLI **dbmsNumPoolagents**

Range for parameter
1 ≤ returned value ≤ (0 − *maxagents*) where *maxagents* is the value of the maxagents configuration parameter.

DB2 version
This monitoring source is supported by DB2 Enterprise-Extended Edition only.

Usage notes
The agent pool contains both idle agents (such as for DB2/6000 Version 2) and MPP and SMP associated subagents. If more agents are created, they will be terminated and not return to the pool when they finish executing.

If the value of this parameter is calculated at run time using other configuration parameters, the label (calculated) appears to the right of the value shown in the output for GET DATABASE MANAGER CONFIGURATION. If −1 (calculated) is shown in the output, the request was issued from a client, and the value was not available.

Use the results from the following parameters and monitoring sources to help you tune the num_poolagents configuration parameter:

- num_initagents parameter (see the *DB2 Administration Guide* for the version of DB2 that you are using)
- “dbms: max_agents” on page 131
- max_querydegree parameter (see the *DB2 Administration Guide* for the version of DB2 that you are using)
- “dbms: max_coordagents” on page 131

dbms: Percent private memory used

Description
Returns the percentage of private memory used.

CLI **dbmsPctPrivMemUsed**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to evaluate your setting of the minimum committed private memory (min_priv_mem) configuration parameter. The min_priv_mem parameter sets the amount of storage that can be reserved for private virtual memory for a DB2 instance. A returned value of around 100% indicates that you are using all of the private memory that was reserved when the instance was started.

If the returned value is consistently less than 100% over a period of time, your value for min_priv_mem might be too high. In this case, you are reserving more memory than you are typically using.

A returned value higher than 100% indicates that there was not enough memory available and the server probably tried to obtain more from the operating system. If the returned value is consistently higher than 100% over a period of time, your value for min_priv_memory might be too low. In this case, your overhead is increased because more operating system resources are used.

For more information about the min_priv_mem parameter, see “dbms: Minimum committed private memory” on page 133 or the *DB2 Administration Guide* for the version of DB2 that you are using.

Default monitoring schedule

Every 60 minutes without any time restrictions.

dbms: query_heap_sz

Description

Returns the maximum amount of memory (in 4 KB pages) that could be allocated for the query heap within the DB2 instance during the monitoring interval.

CLI **dbmsQueryHeapSz**

Range for parameter

2 ≤ returned value ≤ 524288

DB2 version

This monitoring source is supported for DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

Use a query heap to store each query in the agent’s private memory.

Use the results from the “dbms: aslheapsz” on page 129 monitoring source to help you tune the query_heap_sz parameter.

dbms: rqrioblk

Description

Returns the current value of the client I/O block size (rqrioblk) configuration parameter.

CLI **dbmsRqrioblk**

Range for parameter

4096 ≤ returned value ≤ 65535 (in bytes)

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This value is the amount of memory that is allocated for the communication buffer between remote applications and their database agents on the database server. When a database client requests a connection to a remote database, this communication buffer is allocated on the client. On the database server, a communication buffer of 32767 bytes is initially allocated, until a connection is established and the server can determine the value of rqrioblk at the client.

In addition to this communication buffer, this parameter is also used to determine the I/O block size at the database client when a blocking cursor is opened.

dbms: sheapthres

Description

Returns the current value of the sort heap threshold (sheapthres) configuration parameter.

CLI **dbmsSheapthres**

Range for parameter

250 ≤ returned value ≤ 524288 (in 4 KB pages)

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

This value is the maximum amount of memory that the database manager allocates for piped sorts. Piped sorts perform better than nonpiped sorts and tend to be used more often. However, their use can impact performance.

Use the following monitoring sources to help you tune the sheapthres parameter:

- “db: Active sorts” on page 171
- “db: Percent sort overflowed” on page 172
- “db: Sort heap allocated (4 KB pages)” on page 172
- “db: Sort overflows” on page 173
- “db: Total sorts” on page 173
- “dbms: # of piped sorts accepted” on page 37
- “dbms: # of piped sorts requested” on page 37
- “dbms: Percent piped sort hit ratio” on page 39
- “dbms: Post threshold sorts” on page 175

The value for sheapthres should be at least two times the largest sortheap defined for any database within the instance. For more information, see the *DB2 Administration Guide* for the version of DB2 that you are using.

dbms: sqlstmtsz

Description
Returns the current value of the SQL statement size (sqlstmtsz) configuration parameter.

CLI **dbmsSqlstmtsz**

Range for parameter
0 ≤ returned value ≤ 32767 (in bytes)

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
This value is the maximum amount of dynamic SQL statement text that can be returned by the database system monitor.

If the amount of the text returned by the database system monitor exceeds the limit set by sqlstmtsz, the excess statement text is truncated. If text truncation is a problem, then increase the value of sqlstmtsz.

Use the following monitoring sources to help you tune the sqlstmtsz parameter:

- "db: Dynamic SQL statements" on page 190
- "db: Static SQL statements" on page 193

DB2 Data Replication Relational Monitoring Collection

Collection name:
DB2_DataReplication_Monitors

The monitoring sources in this collection provide information about data replication activity from DataPropagator Relational (DPROPR), which works with DB2 for common servers, and IBM Replication, which is an integrated component of DB2 UDB and DB2 Enterprise-Extended Edition. DPROPR and IBM Replication copy relational data from source tables to target tables.

DPROPR and IBM Replication use two components: the Capture program and the Apply program. The Capture program reads the DB2 database log or journal records to capture data about changes made to source tables. The Apply program refreshes or updates a target table, depending on the applicable source-to-target rules.

For more information about data replication and DB2, see either the *DataPropagator Relational Guide* (for replication using DPROPR) or the *Replication Guide and Reference* (for replication using IBM Replication).

Table 33 lists the monitoring sources in this collection.

Table 33. Monitoring Sources in the DB2 Data Replication Relational Monitoring Collection

GUI Name	CLI Name	Returned Value Format
DataJoiner status	DataJoinerStatus	String: "up" or "down"

DB2 Data Replication Relational Monitoring Collection

Table 33. Monitoring Sources in the DB2 Data Replication Relational Monitoring Collection (continued)

GUI Name	CLI Name	Returned Value Format
Data replication Apply status	DproprApplyStatus	String: ″up″ or ″down″
SNA link status	SnaLinkStatus	String: ″up″ or ″down″
TCP/IP communication link status	TcpLinkStatus	String: ″up″ or ″down″
db: # of Apply lagging subscriptions	dbDproprAppTotSubLag	Numeric
db: Apply failed subscriptions	dbDproprAppFailedSub	Numeric
db: Apply requires full refresh	dbDproprAppNeedFullRefresh	Numeric: **0** No **> 0** Yes
db: Apply subscriptions lag time	dbDproprAppSubLagTime	Numeric: minutes
db: Apply subscription status	dbDproprAppSubStatus	Numeric: **0** Completed **1** In progress **3** Failed
db: Capture errors	dbDproprCapErr	Numeric
db: Capture lag	dbDproprCapLag	Numeric: minutes
db: Capture pruning	dbDpropCapPruning	Numeric

Eight of these monitoring sources can be used to monitor the Capture and Apply programs executing at DB2 for MVS/ESA databases. These monitors are:

- db: # of Apply lagging subscriptions
- db: Apply failed subscriptions
- db: Apply requires full refresh
- db: Apply subscriptions lag time
- db: Apply subscription status
- db: Capture errors
- db: Capture lag
- db: Capture pruning

Before you can use these eight monitoring sources on DB2 for MVS/ESA databases, connectivity must be established to the machine that the databases are on, and the databases must be defined to the DB2 instances (on AIX or NT) that are managed by DB2 ECC. Then, you can use these eight monitoring sources with these databases as you would with any other database in the DB2 instance. For information on defining DB2 for MVS/ESA databases to a DB2 instance on AIX or NT, see the *DB2 Enterprise Control Center for TME 10 User's Guide*.

DataJoiner status

Description

Checks whether the DataJoiner instance is up or down.

CLI **DataJoinerStatus** *DataJoiner_instance_name*

Arguments

DataJoiner_instance_name
Specify the name of the DataJoiner instance.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is a string (″up″ or ″down″) that indicates whether the DataJoiner instance is currently up or down.

Data replication Apply status

Description
Checks whether the Apply program is running.

CLI **DproprApplyStatus**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is a string (″up″ or ″down″) that indicates whether the Apply program is currently up or down.

SNA link status

Description
Checks the status of the SNA link to a target SNA link station. This monitor is supported on AIX only.

CLI **SnaLinkStatus** *Link_station_name*

Arguments

Link_station_name
Specify the name of the target SNA link station.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is a string (″up″ or ″down″) that indicates whether the SNA link to a target SNA link station is up or down.

Because your replication environment is highly dependent on the communications between the source and target databases, use this monitoring source to monitor this communication link often.

TCP/IP communication link status

Description
Returns the status of the TCP/IP communications link to a target host.

CLI **TcpLinkStatus** *Host_name*

Arguments

Host_name
Specify the name of the target host.

DB2 Data Replication Relational Monitoring Collection

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is a string (″up″ or ″down″) that indicates whether the TCP/IP link to a target host is up or down.

Because your replication environment is highly dependent on the communications between the source and target databases, use this monitoring source to monitor this communication link often.

db: # of Apply lagging subscriptions

Description

Returns the number of lagging subscriptions for the Apply program.

CLI **dbDproprAppTotSubLag** *Apply_ID*

Arguments

Apply_ID

Specify the subscriber user ID that started this invocation of the Apply program.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine how many of your subscriptions are not processed on time according to their subscription frequencies and priorities.

db: Apply failed subscriptions

Description

Returns the number of subscriptions that the Apply program failed to replicate.

CLI **dbDproprAppFailedSub** *Apply_ID*

Arguments

Apply_ID

Specify the subscriber user ID that started this invocation of the Apply program.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to see whether any subscriptions failed to be replicated.

db: Apply requires full refresh

Description

Returns a number that indicates whether a target table requires a full refresh.

CLI **dbDproprAppNeedFullRefresh** *Apply_ID Owner_name Table_name*

Arguments

Apply_ID
Specify the subscriber user ID that started this invocation of the Apply program. If you are running this monitoring source on DB2 for common servers, this variable is optional.

Owner_name
Specify the owner of the target table.

Table_name
Specify the name of the target table.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine whether an Apply target table needs a full refresh. The returned value is one of the following values:

Value	Refresh needed
0	No
> 0	Yes

If you disabled full refreshes from automatically occurring in your replication environment, you might want to use this monitoring source to check whether any critical subscriptions need a full refresh.

db: Apply subscriptions lag time

Description
Returns a number indicating whether the subscription is lagging or is due to run again.

CLI **dbDproprAppSubLagTime** *Apply_ID Owner_name Table_name*

Arguments

Apply_ID
Specify the subscriber user ID that started this invocation of the Apply program.

Owner_name
Specify the owner of the target table for the subscription.

Table_name
Specify the name of the target table for the subscription.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
A positive number indicates that the subscription is lagging by that number of minutes. A negative number indicates that the subscription is due to run again in that number of minutes.

Value	Subscription status
+|*n*|	Lagging by *n* minutes
−|*n*|	Will run again in *n* minutes

where *n* is the value returned by the monitor.

Even if more than one subscription is eligible for refresh concurrently, the Apply program processes a single subscription at a time. The subscription frequency and priority specified for each subscription determine the order of the subscription processing in this situation. The Apply program attempts to replicate your data based on the specified frequencies and priorities. However, if one subscription takes 30 minutes to process, the next subscription is queued behind the 30-minute subscription. In this case, the Apply program might not be able to meet the specified subscription interval for the second subscription and the subscription is said to be lagging. The lag time is the difference between the current time and the time when the subscription was supposed to run according to its subscription frequency.

db: Apply subscription status

Description

Returns the status of the Apply subscription for a target table.

CLI **dbDproprAppSubStatus** *Apply_ID Owner_name Table_name*

Arguments

Apply_ID
Specify the subscriber user ID that started this invocation of the Apply program.

Owner_name
Specify the owner of the target table.

Table_name
Specify the name of the target table.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned values are:

Value	Subscription status
0	Completed successfully
1	In progress
3	Error encountered by the Apply program during processing

When using this monitoring source, make sure that the monitoring interval is in synch with the subscription frequency and with the average frequency at which the Apply program processes this subscription.

db: Capture errors

Description

Returns the number of errors encountered by the Capture program within the last five minutes.

CLI **dbDproprCapErr**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine whether the Capture program encountered an error that prevented it from running. If any errors are detected, it means that the Capture program came down at the time the error occurred. It does not mean that the Capture program is still down.

The Capture program is the most critical replication component in the replication system. If the Capture program is not active, there will be no new change records to apply to the target systems. If your data concurrency requirements are high and you want to ensure that the Capture program runs continuously, use this monitoring source to determine whenever the Capture program encounters an error that prevents it from running.

db: Capture lag

Description

Returns the time difference (in minutes) between the current timestamp and the last timestamp recorded by the Capture program. This time difference is the Capture lag.

CLI **dbDproprCapLag**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine whether the Capture program is keeping up with the DB2 database log. The Capture program uses an interface to the DB2 database log or journal to detect and save changes to the data in the tables registered for replication.

db: Capture pruning

Description

Returns the number of rows in the unit-of-work (UOW) table.

CLI **dbDpropCapPruning**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you determine whether you need to prune the UOW table or the change data (CD) table.

DB2 Locks and Deadlocks Monitoring Collection

Collection name:

DB2_Lock_Monitors

The monitoring sources in this collection provide statistics about locks and deadlocks for applications and databases.

Table 34 on page 144 lists the monitoring sources in this collection.

DB2 Locks and Deadlocks Monitoring Collection

Table 34. Monitoring Sources in the DB2 Locks and Deadlocks Monitoring Collection

GUI Name	CLI Name	Returned Value Format
conn: Deadlocks found during last interval	connDeadlocksDelta	Numeric
conn: Lock escalations during last interval	connLockEscalsDelta	Numeric
conn: Lock wait time (s)	connLockWaitTimeDelta	Numeric: seconds
conn: Lock waits	connLockWaits	Numeric
conn: Locks held	connLocksHeld	Numeric
conn: UOW lock wait time (s)	connUowLockWaitTime	Numeric: seconds
db: Applications in lock wait	dbApplsInLkwt	Numeric
db: Average lock wait time (s)	dbAvgLockWaitTime	Numeric: seconds
db: Average locks held per application	dbLocksHeldPerAppl	Numeric
db: Deadlocks found during last interval	dbDeadlocksDelta	Numeric
db: Lock escalations during last interval	dbLockEscalationsDelta	Numeric
db: Lock list in use (bytes)	dbLockListInUse	Numeric
db: Lock timeouts during last interval	dbLockTimeoutsDelta	Numeric
db: Lock waits	dbLockWaits	Numeric
db: Lock waits in an interval	dbLockWaitsDelta	Numeric
db: Locks held	dbLocksHeld	Numeric
db: Percent application in lock wait	dbPctApplsInLckWt	Percentage
db: Percent deadlock rollbacks	dbPctDlckRollbacks	Percentage
db: Time waited on locks	dbLockWaitTime	Numeric
db: Total exclusive lock escalations	dbXLockEscals	Numeric

conn: Deadlocks found during last interval

Description

Returns the total number of deadlocks that occurred for the application during the monitoring interval.

CLI **connDeadlocksDelta** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine whether the application is experiencing contention problems. Modify the application to better enable it to execute concurrently.

conn: Lock escalations during last interval

Description

Returns the total number of lock escalations for the application during the monitoring interval.

CLI **connLockEscalsDelta** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you evaluate the settings of the locklist and maxlocks configuration parameters. Lock escalations can result in a decrease in concurrency between applications connected to a database.

A lock is escalated when the total number of locks held by the application reaches the maximum amount of lock list space available to the application. Possible causes of lock escalations are:

- The lock list size is too small for the number of concurrent applications. The lock list size is determined by the value of the locklist parameter.
- The percentage of the lock list usable by each application is too small. This percentage is determined by the value of the maxlocks parameter.
- One or more applications is using an excessive number of locks.

To resolve these causes, consider:

- Increasing the value of locklist.
- Increasing the value of maxlocks.
- Identifying the applications that have a large number of locks. These can cause lock escalations in other applications. These applications might need to use table locks instead of row locks, although table locks can cause an increase in the values for the "conn: Lock wait time (s)" and "conn: Locks wait" monitoring sources in this collection.
- Identifying applications holding too much of the lock list, by using the returned value from the "conn: Percent locklist space used by application" monitoring source (page 17).

conn: Lock wait time (s)

Description

Returns the total time (in seconds) that the application spent waiting for a lock to be granted to it during the monitoring interval.

CLI **connLockWaitTimeDelta** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value with the value returned by the “conn: Lock waits” monitoring source (page 146) to calculate the average lock wait time for this application.

If the average lock wait time is high, look for applications that are holding many locks or have lock escalations, so you can tune those applications to improve concurrency. If lock escalations are the reason for a high average lock wait time, then you might need to increase the values of the locklist and maxlocks configuration parameters.

conn: Lock waits

Description

Returns the total number of times that the application requested a lock but had to wait because another connection was already holding a lock on the data.

CLI **connLockWaits** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value with the value returned by the “conn: Lock wait time (s)” monitoring source to calculate the average lock wait time for this application. See “conn: Lock wait time (s)” on page 146 for the implications of a high average lock wait time.

conn: Locks held

Description

Returns the total number of locks currently held by the application.

CLI **connLocksHeld** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine if the application is approaching the maximum number of locks available to it, as set by the maxlocks configuration parameter. The maxlocks parameter indicates the percentage of the lock list that each application can use before lock escalations occur.

If the application holds a large number of locks, it might need to perform more commits so that some of the locks can be released.

conn: UOW lock wait time (s)

Description

Returns the total amount of elapsed time (in seconds) that the unit of work (UOW) spent waiting for locks.

CLI **connUowLockWaitTime** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to help you determine the severity of the resource contention problem.

db: Applications in lock wait

Description
Returns the number of applications that are currently waiting on locks in the database.

CLI **dbApplsInLkwt**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
If the returned value is high compared to the total number of applications currently connected to the database, the applications might be having concurrency problems. In this case, you should identify applications that are holding locks or exclusive locks for long periods of time.

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: Average lock wait time (s)

Description
Returns the average time (in seconds) that applications wait for a lock within the database during the monitoring interval.

CLI **dbAvgLockWaitTime**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
If the returned value is high, you should look for the applications that hold many locks, or have lock escalations. You can tune those applications to improve concurrency. If lock escalations are the reason for a high average lock wait time (see “db: Lock escalations during last interval” on page 149 for more information), then you might need to increase the values of the locklist and maxlocks configuration parameters.

db: Average locks held per application

Description
Returns the average number of locks held per currently connected application in the database.

CLI **dbLocksHeldPerAppl**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

If the returned value is high, it might indicate that one or more applications is using an excessive number of locks. Tune such applications to improve performance.

db: Deadlocks found during last interval

Description

Returns the number of deadlocks detected in the database during the last monitoring interval.

CLI **dbDeadlocksDelta**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine whether applications are experiencing contention problems in the database. These problems could be caused by the following situations:

- Lock escalations are occurring for the database.
- An application might be locking tables explicitly when system-generated row locks might be sufficient.
- An application might be using an inappropriate isolation level when binding.
- Database catalog tables are locked for repeatable read.
- Applications are getting the same locks in different orders, resulting in deadlock.

You might be able to resolve the problem by determining in which applications the deadlocks are occurring. You could then try to modify the applications to better enable them to execute concurrently.

Default monitoring schedule

Every 60 minutes without any time restrictions.

db: Lock escalations during last interval

Description

Returns the number of lock escalations that occurred in the database during the monitoring interval. Exclusive lock escalations are included in this number.

CLI **dbLockEscalationsDelta**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you evaluate the settings of the locklist and maxlocks configuration parameters. Lock escalations can result in a decrease in concurrency between applications connected to a database.

See “conn: Lock escalations during last interval” on page 145 for a discussion on resolving lock escalation problems.

Default monitoring schedule

Every 60 minutes without any time restrictions.

db: Lock list in use (bytes)

Description
Returns the total amount of lock list memory (in bytes) that is in use in the database.

CLI **dbLockListInUse**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
If the returned value is approximately the value of the locklist configuration parameter, you might want to consider increasing the size of locklist. Because the value of locklist is given in 4 KB pages, make sure that you convert it to bytes (locklist × 4096 bytes) to compare the two numbers.

db: Lock timeouts during last interval

Description
Returns the number of times that a request to lock an object timed out instead of being granted during the monitoring interval.

CLI **dbLockTimeoutsDelta**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to help you evaluate the locktimeout configuration parameter. If the returned value is high when compared to normal operating levels, you might have an application that is holding locks for long durations. In this case, use some of the other monitoring sources in this collection to determine whether you have an application problem.

If this value is low, your locktimeout parameter might be set too high. In this situation, your applications might be waiting excessively to obtain a lock. For more information, see the *DB2 Administration Guide* for the version of DB2 that you are using.

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: Lock waits

Description
Returns the total number of times that applications had to wait for locks within the database.

CLI **dbLockWaits**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is used with the value returned by the “db: Time waited on locks” monitoring source (page 152) to calculate the average lock wait time for applications using the database. See “db: Average lock wait time (s)” on page 148 for the implications of a high average lock wait time.

db: Lock waits in an interval

Description
Returns the number of times that applications had to wait for locks in the database during the monitoring interval.

CLI **dbLockWaitsDelta**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value as an indication of how much time is spent in waiting on locks for a particular monitoring interval.

db: Locks held

Description
Returns the total number of locks currently held by all applications in the database.

CLI **dbLocksHeld**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value is used in calculating the returned value for the “db: Percent locklist space used by database” monitoring source (page 32).

db: Percent application in lock wait

Description
Returns the percentage of currently connected applications that are waiting on a lock in the database.

CLI **dbPctApplsInLckWt**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
If the returned value is high, the applications could have concurrency problems. You should identify applications that are holding locks or exclusive locks for long periods of time and determine whether they can release their locks more often.

Default thresholds and actions

Table 35. Default Thresholds and Actions for db: Percent application in lock wait

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 85%	Send Tivoli notice Change icon

Table 35. Default Thresholds and Actions for db: Percent application in lock wait (continued)

Response Level	Trigger When	Default Actions
warning	Greater than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: Percent deadlock rollbacks

Description
Returns the percentage of rollbacks that were due to deadlock during the monitoring interval.

CLI **dbPctDlckRollbacks**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
A high returned value indicates that applications are experiencing contention problems. Rollbacks due to deadlock lower the throughput of the database. You might be able to resolve the problem by determining in which applications the deadlocks are occurring. Then, modify those applications to better enable them to execute concurrently.

Default thresholds and actions

Table 36. Default Thresholds and Actions for db: Percent deadlock rollbacks

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 90%	Send Tivoli notice Change icon
warning	Greater than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: Time waited on locks

Description
Returns the total elapsed time (in seconds) that all applications waited for a lock within the database.

CLI **dbLockWaitTime**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used with the value returned by the “db: Lock waits” monitoring source (page 150) to calculate the average lock wait time for applications using the database. See “db: Average lock wait time (s)” on page 148 for the implications of a high average lock wait time.

db: Total exclusive lock escalations

Description

Returns the total number of times that locks were escalated from several row locks to one exclusive table lock, or the number of times an exclusive lock on a row caused the table lock to become an exclusive lock.

CLI **dbXLockEscals**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to track exclusive locks in the database. Because other applications cannot access data held by an exclusive lock, these locks can result in a decrease in concurrency between applications connected to the database.

A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application. Possible causes of exclusive lock escalations are:

- The lock list size is too small for the number of concurrent applications. The lock list size is determined by the value of the locklist configuration parameter.
- The percentage of the lock list usable by each application is too small. This percentage is determined by the value of the maxlocks configuration parameter.
- One or more applications is using an excessive number of locks.
- An application might be using exclusive locks when share locks are sufficient.

To resolve these causes, consider:

- Increasing the value of locklist.
- Increasing the value of maxlocks.
- Identifying the applications that have a large number of locks. These can cause lock escalations in other applications. These applications might need to use table locks instead of row locks, although table locks can cause an increase in the values for the “conn: Lock wait time (s)” and “conn: Locks wait” monitoring sources in this collection.

DB2 Query Monitoring Collection

Collection name:

DB2_Query_Monitors

The monitoring sources in this collection provide general statistics about database objects.

Table 37 on page 154 lists the monitoring sources in this collection.

Table 37. Monitoring Sources in the DB2 Query Monitoring Collection

GUI Name	CLI Name	Returned Value Format
db: # of database table spaces	dbTotalDbTbsp	Numeric
db: # of event monitors	dbTotalEventMonitor	Numeric
db: # of invalid triggers	dbTotalInvTrigger	Numeric
db: # of system table space	dbTotalSystemTbsp	Numeric
db: # of tables	dbTotalTable	Numeric
db: # of table spaces	dbTotalTablespace	Numeric
db: # of table spaces with LONG data	dbTotalTbspLONG	Numeric
db: # of triggers	dbTotalTrigger	Numeric
db: # of user indexes	dbTotalUser	Numeric
db: # of views	dbTotalView	Numeric
db: User-defined SQL (numeric result)	dbUserDefinedSqlN	Numeric
db: User-defined SQL (string result)	dbUserDefinedSqlS	String
table: # of rows	tableTotalRow	Numeric
tbsp: # of tables	tbspTotalTable	Numeric
user: # of tables	userTotalTable	Numeric

db: # of database table spaces

Description

Returns the number of Database Managed Space (DMS) table spaces in the database.

CLI **dbTotalDbTbsp**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to evaluate your use of DMS table spaces and their effects on performance. When reading table data using the buffer pool, DMS table spaces do not use the file system or its cache for data storage, and therefore do not have double buffering that can occur for system managed space (SMS) table spaces (for a discussion of this topic, see the *DB2 Administration Guide* for the version of DB2 that you are using). As a result, if you have many DMS table spaces compared to the number of SMS table spaces, you might want to increase the size of the database buffer pool and reduce the file system cache for better I/O performance.

db: # of event monitors

Description

Returns the number of event monitors defined in the database.

CLI **dbTotalEventMonitor**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine how many event monitors are defined for the database. When you define an event monitor, its definition is stored in the database system catalog table. You can create any number of event monitors. However, the maximum number of event monitors that can be active for a database at any given time is 32.

db: # of invalid triggers

Description
Returns the number of triggers that are marked invalid in the database.

CLI **dbTotalInvTrigger**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine the number of triggers that must be revalidated.

A trigger is marked invalid if an object on which the trigger depends is dropped. To revalidate an invalid trigger, you must retrieve its definition from the database system catalog and submit a new CREATE TRIGGER statement.

Default thresholds and actions

Table 38. Default Thresholds and Actions for db: # of invalid triggers

Response Level	Trigger When	Default Actions
critical	Greater than 20	Send Tivoli notice Change icon
severe	Greater than 15	Send Tivoli notice Change icon
warning	Greater than 1	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: # of system table spaces

Description
Returns the number of SMS table spaces in the database.

CLI **dbTotalSystemTbsp**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to evaluate your use of SMS table spaces and their effects on performance. Table data that is read from disk is normally available in the database's buffer pool. Sometimes a data page is freed from the buffer pool before it is used. For SMS table spaces, when the database manager requests that data page from the file system, the data page might still be in the file system's own cache. Having the page in the file system's cache saves an I/O operation that would otherwise have been required. (For more information, see the *DB2 Administration Guide* for the version of DB2 that you are using.) If you have many SMS table spaces, you might want to increase the size of the file system cache to take advantage of this extra buffering.

db: # of tables

Description

Returns the number of tables in the database.

CLI **dbTotalTable**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use this monitoring source to track database growth due to an increased number of tables over a period of time.

db: # of table spaces

Description

Returns the number of table spaces in the database.

CLI **dbTotalTablespace**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use this monitoring source to track database growth over a period of time.

db: # of table spaces with LONG data

Description

Returns the number of table spaces that store LONG data in the database.

CLI **dbTotalTbspLONG**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use this monitoring source to track database growth over a period of time. LONG data can take up a large amount of space in a database.

db: # of triggers

Description

Returns the number of triggers defined in the database.

CLI **dbTotalTrigger**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use this monitoring source to track the use of triggers in the database. There are benefits to using triggers, including faster application development, easier maintenance, and global enforcement of business rules. For more information, see the *DB2 Administration Guide* for the version of DB2 that you are using.

db: # of user indexes

Description
Returns the number of indexes created by users in the database. Indexes created by SYSIBM are not counted.

CLI **dbTotalUser**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use this monitoring source to track the use of indexes in the database.

The use of indexes can improve performance; for example, faster sorting of data. However, indexes can also have adverse effects on performance; for example, each INSERT or DELETE operation performed on a table requires additional updating of each index on that table. For a discussion of this topic, see the *DB2 Administration Guide* for the version of DB2 that you are using.

db: # of views

Description
Returns the number of views in the database.

CLI **dbTotalView**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use this monitoring source to track the use of views in the database.

Views can be created to limit access to sensitive data, while allowing more general access to other data. This provides flexibility in the way your programs and end-user queries can look at the table data.

db: User-defined SQL (numeric result)

Description
Runs a user-defined SQL statement to monitor database resources.

CLI **dbUserDefinedSqlN** *SQL_statement*

Arguments

SQL_statements
Specify the DB2 SQL statement.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
You enter a complete DB2 SQL statement. The SQL statement is expected to return a numeric value.

db: User-defined SQL (string result)

Description
Runs a user-defined SQL statement to monitor database resources.

CLI **dbUserDefinedSqlS** *SQL_statement*

Arguments

SQL_statement
Specify the DB2 SQL statement.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
You enter a complete DB2 SQL statement. The SQL statement is expected to return a string value.

table: # of rows

Description
Returns the number of rows in the table.

CLI **tableTotalRow** *Owner_name Table_name*

Arguments

Owner_name
Specify the owner of the table.

Table_name
Specify the name of the table.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use this monitoring source to track growth of the table over a period of time.

tbsp: # of tables

Description
Returns the number of tables in the table space.

CLI **tbspTotalTable** *DB2_table_space_name*

Arguments

DB2_table_space_name
Specify the name of the DB2 table space.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use this monitoring source to track usage of the table space over a period of time.

user: # of tables

Description
Returns the number of tables owned by the database user.

CLI **userTotalTable** *User_name*

Arguments

User_name
Specify the name of the user.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use this monitoring source to track the use of database resources for tables by a particular user over a period of time.

DB2 SNMP Monitoring Collection

Collection name:
DB2_SNMP_Monitors

The monitoring sources in this collection provide information from the DB2 Simple Network Management Protocol (SNMP) subagent. This subagent is a program that resides on a managed node and facilitates DB2 systems management through products that conform to the SNMP protocol.

The SNMP subagent:

- Generates alerts to an SNMP manager in the case of a severe DB2 error, such as a condition that requires operator intervention.
- Supplies information about the status of the DB2 server and databases. The information provided is documented in the Internet Engineering Task Force standard (RFC 1697). This monitoring collection implements a subset of this available information.

You can use the monitoring sources in this collection to monitor key aspects of your DB2 environment without specifically defining DB2 endpoints for them, or to monitor DB2 installations on machines where Tivoli framework services are unavailable. For example, you can use these monitoring sources to monitor DB2 instances and databases that reside on OS/2 machines. See the *DB2 Enterprise Control Center for TME 10 User's Guide* for more information.

Table 39 on page 160 lists the monitoring sources in this collection.

DB2 SNMP Monitoring Collection

Table 39. Monitoring Sources in the DB2 SNMP Monitoring Collection

GUI Name	CLI Name	Returned Value Format
DB2 database state	dbRelStat	Numeric: **1** Undetermined **2** Active **3** Available **4** Quiesced **5** Unavailable
DB2 server accumulated connections	srvAccumInAssoc	Numeric
DB2 server current connections	srvInAssoc	Numeric
DB2 server finished transactions	srvInfoFinishedTransactions	Numeric
DB2 server status	srvStatus	Numeric: **1** Up **2** Down **3** Halted **4** Congested **5** Restarting
DB2 server version	srvVersion	String
Maximum number of agents allowed to register at the same time	srvMaxInboundAssociations	Numeric
Maximum number of agents registered at the same time	srvHighwaterInboundAssociations	Numeric
Total Logical bytes read by DB2 server for all active databases	srvLogicalReads	Numeric
Total logical pages read by DB2 server for all active databases	srvPageReads	Numeric
Total physical bytes read by DB2 server for all active databases	srvDiskReads	Numeric
Total physical bytes written by DB2 server for all active databases	srvDiskWrites	Numeric
Total physical pages written by DB2 server for all active databases	srvPageWrites	Numeric
Total rows selected returned by DB2 server for all active databases	srvRequestSends	Numeric
Total SQL statements handled by DB2 server for all active databases	srvHandledRequests	Numeric
Total SQL statements received by DB2 server for all active databases	srvRequestRecvs	Numeric

DB2 database state

Description

Returns a number that indicates the current state of the database.

CLI **dbRelStat** *Community_name Instance_name Database_name*

Arguments

Community_name
Specify the identifier used to group together a set of SNMP agents that are managed together. Unless explicitly set by the user, the default community name honored by SNMP agents is the string ″public″.

Instance_name
Specify the name of the DB2 instance that contains the database.

Database_name
Specify the name of the DB2 database.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine the state of the database. The returned value is one of the following values:

Value	Database state
1	Other (undetermined)
2	Active
3	Available
4	Quiesced
5	Unavailable (DB2 instance down)

DB2 server accumulated connections

Description

Returns the sum of the total number of connections that are made for all active databases in the DB2 server.

CLI **srvAccumInAssoc** *Community_name Instance_name*

Arguments

Community_name
Specify the identifier used to group together a set of SNMP agents that are managed together. Unless explicitly set by the user, the default community name honored by SNMP agents is the string ″public″.

Instance_name
Specify the name of the DB2 instance.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to evaluate the frequency at which applications connect to databases managed by this DB2 server.

DB2 server current connections

Description

Returns the number of current local and remote connections to the databases in the DB2 server.

CLI **srvInAssoc** *Community_name Instance_name*

Arguments

Community_name

Specify the identifier used to group together a set of SNMP agents that are managed together. Unless explicitly set by the user, the default community name honored by SNMP agents is the string ″public″.

Instance_name

Specify the name of the DB2 instance.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to evaluate the level of activity within the DB2 server and the amount of system resource being used.

DB2 server finished transactions

Description

Returns the total number of SQL COMMIT and ROLLBACK statements for all active databases in the DB2 server.

CLI **srvInfoFinishedTransactions** *Community_name Instance_name*

Arguments

Community_name

Specify the identifier used to group together a set of SNMP agents that are managed together. Unless explicitly set by the user, the default community name honored by SNMP agents is the string ″public″.

Instance_name

Specify the name of the DB2 instance.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you determine the level of database activity for all databases managed by this DB2 server. The returned value is an indication of the units of work performed by the databases.

DB2 server status

Description

Returns a number that indicates the current status of the DB2 server.

CLI **srvStatus** *Community_name Instance_name*

Arguments

Community_name
Specify the identifier used to group together a set of SNMP agents that are managed together. Unless explicitly set by the user, the default community name honored by SNMP agents is the string ″public″.

Instance_name
Specify the name of the DB2 instance.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine the status of the DB2 server. The returned value is one of the following values:

Value	Server status
1	Up
2	Down
3	Halted
4	Congested
5	Restarting

DB2 server version

Description

Returns the DB2 server product version ID.

CLI **srvVersion** *Community_name Instance_name*

Arguments

Community_name
Specify the identifier used to group together a set of SNMP agents that are managed together. Unless explicitly set by the user, the default community name honored by SNMP agents is the string ″public″.

Instance_name
Specify the name of the DB2 instance.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine the product version ID of the DB2 server. The returned value is in the form *pppvvrrm*, where:

ppp SQL

vv A 2-digit version number (with high-order 0 in the case of a 1-digit version)

rr A 2-digit release number (with high-order 0 in the case of a 1-digit release)

m A 1-digit modification level

Maximum number of agents allowed to register at the same time

Description

Returns the maximum number of agents that can be registered at the same time for the DB2 server.

CLI **srvMaxInboundAssociations** *Community_name Instance_name*

Arguments

Community_name
Specify the identifier used to group together a set of SNMP agents that are managed together. Unless explicitly set by the user, the default community name honored by SNMP agents is the string ″public″.

Instance_name
Specify the name of the DB2 instance.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value with the value returned by the “Maximum number of agents registered at the same time” monitoring source (page 164) to help you evaluate the appropriateness of the current settings for the maximum agents allowed (maxagents) configuration parameter for the DB2 server. If the values from the two monitoring sources are comparable, you might want to increase the number of maximum agents allowed.

Maximum number of agents registered at the same time

Description

Returns the highest number of agents that the DB2 server has registered at the same time.

CLI **srvHighwaterInboundAssociations** *Community_name Instance_name*

Arguments

Community_name
Specify the identifier used to group together a set of SNMP agents that are managed together. Unless explicitly set by the user, the default community name honored by SNMP agents is the string ″public″.

Instance_name
Specify the name of the DB2 instance.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you evaluate the appropriateness of the current settings for the maximum agents allowed (maxagents) configuration parameter for the DB2 server. If this returned value is near the value for the maximum agents allowed, you might want to increase the number of maximum agents allowed.

Total Logical bytes read by DB2 server for all active databases

Description

Returns the total number of logical bytes read from disk for all active databases in the DB2 server.

CLI **srvLogicalReads** *Community_name Instance_name*

Arguments

Community_name

Specify the identifier used to group together a set of SNMP agents that are managed together. Unless explicitly set by the user, the default community name honored by SNMP agents is the string ″public″.

Instance_name

Specify the name of the DB2 instance.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you determine the level of I/O activity due to logical reads for all databases managed by this server.

Total logical pages read by DB2 server for all active databases

Description

Returns the total number of logical pages read from disk for all active databases in the DB2 server. Only data and index reads are included. Other I/O operations, such as reads from the log files, backups, and restores, are not included in this number.

CLI **srvPageReads** *Community_name Instance_name*

Arguments

Community_name

Specify the identifier used to group together a set of SNMP agents that are managed together. Unless explicitly set by the user, the default community name honored by SNMP agents is the string ″public″.

Instance_name

Specify the name of the DB2 instance.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you determine the level of I/O activity due to logical reads for all databases managed by this server.

Total physical bytes read by DB2 server

Description

Returns the total number of physical bytes read from disk for all active databases in the DB2 server. Only data and index reads are included. Other I/O operations, such as reads from the log files, backups, and restores, are not included in this number.

CLI **srvDiskReads** *Community_name Instance_name*

Arguments

Community_name
Specify the identifier used to group together a set of SNMP agents that are managed together. Unless explicitly set by the user, the default community name honored by SNMP agents is the string ″public″.

Instance_name
Specify the name of the DB2 instance.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you determine the level of I/O activity due to physical reads for all databases managed by this server.

Total physical bytes written by DB2 server for all active databases

Description

Returns the total number of physical bytes written to disk for all active databases in the DB2 server. Only data and index writes are included. Other I/O operations, such as writes to the log files, backups, and restores, are not included in this number.

CLI **srvDiskWrites** *Community_name Instance_name*

Arguments

Community_name
Specify the identifier used to group together a set of SNMP agents that are managed together. Unless explicitly set by the user, the default community name honored by SNMP agents is the string ″public″.

Instance_name
Specify the name of the DB2 instance.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you determine the level of I/O activity due to physical writes for all databases managed by this server.

Total physical pages written by DB2 server for all active databases

Description

Returns the total number of physical pages written to disk for all active

databases in the DB2 server. Only data and index reads are included. Other I/O operations, such as writes to the log files, backups, and restores, are not included in this number.

CLI **srvPageWrites** *Community_name Instance_name*

Arguments

Community_name
Specify the identifier used to group together a set of SNMP agents that are managed together. Unless explicitly set by the user, the default community name honored by SNMP agents is the string ″public″.

Instance_name
Specify the name of the DB2 instance.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to help you determine the level of I/O activity due to physical writes for all databases managed by this server.

Total rows selected returned by DB2 server for all active databases

Description
Returns the total number of rows selected for all active databases in the DB2 server. Only SQL select requests are included.

CLI **srvRequestSends** *Community_name Instance_name*

Arguments

Community_name
Specify the identifier used to group together a set of SNMP agents that are managed together. Unless explicitly set by the user, the default community name honored by SNMP agents is the string ″public″.

Instance_name
Specify the name of the DB2 instance.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to gain insight into the current level of activity within the active databases in the DB2 server.

Total SQL statements handled by DB2 server for all active databases

Description
Returns the total number of SQL statements handled for all active databases in the DB2 server. Only SQL statements are included. Other statements, such as SQL DDL statements, are not included in this number.

CLI **srvHandledRequests** *Community_name Instance_name*

Arguments

Community_name
Specify the identifier used to group together a set of SNMP agents that are managed together. Unless explicitly set by the user, the default community name honored by SNMP agents is the string ″public″.

Instance_name
Specify the name of the DB2 instance.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value indicates the rate of database activity for databases managed by this DB2 server. Use the returned value with the value returned by the “Total SQL statements received by DB2 server for all active databases” monitoring source (page 168) to calculate the number of failed SQL statements (total received minus total handled). This calculation can help you determine reasons for poor performance, because failed statements indicate lower throughput for databases.

Total SQL statements received by DB2 server for all active databases

Description
Returns the total number of SQL statements received for all active databases. Only SQL statements are included. Other statements, such as SQL DDL statements, are not included in this number.

CLI **srvRequestRecvs** *Community_name Instance_name*

Arguments

Community_name
Specify the identifier used to group together a set of SNMP agents that are managed together. Unless explicitly set by the user, the default community name honored by SNMP agents is the string ″public″.

Instance_name
Specify the name of the DB2 instance.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value indicates the rate of database activity for databases managed by this DB2 server. Use the returned value with the value returned by the “Total SQL statements handled by DB2 server for all active databases” monitoring source (page 167) to calculate the number of failed SQL statements (total received minus total handled). This calculation can help you determine reasons for poor performance, because failed statements indicate lower throughput for databases.

DB2 Sort Work Monitoring Collection

Collection name:
DB2_Sort_Monitors

DB2 Sort Work Monitoring Collection

The monitoring sources in this collection provide information about the database manager sort work. Sorting is often required during a query. Therefore, proper configuration of the sort heap areas can be crucial to improving the query's performance. By monitoring sort work, you can determine the current sort work performance and use the information to tune the sort heap configuration parameters (sortheap and sheapthres) and achieve better performance.

Sorting involves two steps:

- The sort phase
- The return of the results of the sort phase

In the sort phase, if the information being sorted cannot fit entirely into the sort heap, the sort overflows and the information must be stored in temporary database tables. Sorts that do not overflow perform better than those that do.

In the return phase, if the sorted information can return directly through the sort heap, it is a piped sort. A piped sort performs better than a nonpiped sort.

For more information on sorting and the sort heap configuration parameters, see the *DB2 Administration Guide* for the version of DB2 that you are using.

Table 40 lists the monitoring sources in this collection.

Table 40. Monitoring Sources in the DB2 Sort Work Monitoring Collection

GUI Name	CLI Name	Returned Value Format
conn: Application sorts	connSorts	Numeric
conn: Sort time (s)	connSortTime	Numeric
conn: Statement sorts	connStmtSorts	Numeric
db: Active sorts	dbActiveSorts	Numeric
db: Average sort time (ms)	dbAvgSortTime	Numeric: milliseconds
db: Percent sort overflowed	dbPctSortOvflowed	Percentage
db: Sort heap allocated (4 KB pages)	dbSortHeapAllocated	Numeric
db: Sort overflows	dbSortOverflows	Numeric
db: Total sort time (ms)	dbTotalSortTime	Numeric: milliseconds
db: Total sorts	dbTotalSorts	Numeric
dbms: Percent piped sorts rejected	dbmsPctPipSrtsRej	Percentage
dbms: Percent sort heap allocated	dbmsPctSortHeapAlloc	Percentage
dbms: Post threshold sorts	dbmsPostThreshSorts	Numeric
dbms: Sort heap allocated (4 KB pages)	dbmsSortHeapAllocated	Numeric
dbms: Total piped sorts rejected	dbmsPipSrtsRej	Numeric

conn: Application sorts

Description

Returns the total number of sorts performed by the application.

CLI **connSorts** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID
Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use this value with the value returned by the "conn: Sort time (s)" monitoring source (page 170) to calculate the average time that the application spends in sorting. This average time can indicate whether sorting is lowering the performance of the application.

conn: Sort time (s)

Description
Returns the time (in seconds) spent in sorting for the application during the last monitoring interval.

CLI **connSortTime** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID
Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value with the value returned by the "conn: Application sorts" monitoring source (page 169) to calculate the average time that the application spends in sorting. This average time can indicate whether sorting is lowering the performance of the application.

conn: Statement sorts

Description
Returns the total number of times that a set of data was sorted to process the OPEN operation of the current SQL statement.

CLI **connStmtSorts** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID
Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to help identify the need for an index, because indexes can reduce the need for sorting a set of data. Identify the SQL statement for which this returned value is providing sort information. Then, analyze this SQL statement to determine index candidates by looking at columns that are being sorted. For example, a column used in an ORDER BY clause might be an index candidate.

db: Active sorts

Description
Returns the number of sorts in the database that currently have a sort heap allocated.

CLI **dbActiveSorts**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value with the value from the "db: Sort heap allocated (4 KB pages)" monitoring source (page 172) to determine the average sort heap space used by each sort in the database. If the sortheap configuration parameter is substantially larger than the average sort heap used, you might be able to lower the value of this parameter.

This value includes heaps for sorts of temporary tables that are created during relational operations.

db: Average sort time (ms)

Description
Returns the average time (in milliseconds) spent in sorting at the database level.

CLI **dbAvgSortTime**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine whether sorting is lowering the performance of the database.

db: Percent sort overflowed

Description

Returns the percentage of sorts that overflowed for the database during the monitoring interval.

CLI **dbPctSortOvflowed**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

A sort overflows when it runs out of sort heap and needs to store information in temporary tables for sorting. If the returned value is high, you might want to adjust the applheapsz, stmtheap, stat_heap_sz, sortheap, and sheapthres configuration parameters.

When a sort overflows, performance might be lowered because the overflowed sort needs a merge phase and could require more I/O.

Default thresholds and actions

Table 41. Default Thresholds and Actions for db: Percent sort overflowed

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 90%	Send Tivoli notice Change icon
warning	Greater than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule

Every 60 minutes without any time restrictions.

db: Sort heap allocated (4 KB pages)

Description

Returns the total amount (in 4 KB pages) of sort heap currently allocated for all sorts in the database.

CLI **dbSortHeapAllocated**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value with the value from the “db: Active sorts” monitoring source (page 171) to determine the average sort heap space used by each sort in the database. If the average sort heap used is substantially less than the value of the sortheap configuration parameter, you might be able to lower the value of sortheap and reduce the memory allocated for the sort heap. If you adjust the value of sortheap, also adjust the applheapsz, stmtheap, stat_heap_sz, and sheapthres configuration parameters.

db: Sort overflows

Description

Returns the total number of sorts that overflowed within a database. A sort overflows when it runs out of sort heap and needs to store information in temporary tables for sorting.

CLI **dbSortOverflows**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

When a sort overflows, performance might be lowered because the overflowed sort needs a merge phase and could require more I/O. If the returned value is high, use the "db: Percent sort overflowed" monitoring source (page 172) to determine whether you want to increase the sort heap size. If you adjust the value of sortheap, also adjust the applheapsz, stmtheap, stat_heap_sz, and sheapthres configuration parameters.

If the number of sort overflows is small with respect to the total sorts, then increasing the sort heap size might have little impact on performance, unless the buffer pool size is increased substantially. For more information on the buffer pool, see "db: buffpage" on page 117 or the *DB2 Administration Guide* for the version of DB2 that you are using.

db: Total sort time (ms)

Description

Returns the total elapsed time (in milliseconds) for all sorts that were executed within a database during the monitoring interval.

CLI **dbTotalSortTime**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculating the average sort time (see "db: Average sort time (ms)" on page 171). Use the average sort time to determine whether sorting is lowering the performance of the database.

The returned value includes the sort time of temporary tables that are created during related operations.

db: Total sorts

Description

Returns the total number of sorts that were executed within a database during the monitoring interval.

CLI **dbTotalSorts**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is used in calculating the average sort time (see "db:

Average sort time (ms)" on page 171) and in calculating the percentage of sorts that need more heap space (see "db: Percent sort overflowed" on page 172).

dbms: Percent piped sorts rejected

Description

Returns the percentage of piped sort requests that were rejected for the DB2 instance during the monitoring interval.

CLI **dbmsPctPipSrtsRej**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

In the sort return phase, if the sorted information can return directly through the sort heap, it is a piped sort. However, even if the optimizer requests a piped sort, this request is rejected at run time if the total amount of sort heap memory for all sorts on the database is close to exceeding the sheapthres value.

If this returned value is high, you might consider decreasing your sort heap (using the sortheap configuration parameter) or increasing your sort heap threshold (using the sheapthres configuration parameter). However, be aware of the implications of these options. If you increase the sort heap threshold, more memory might remain allocated for sorting, causing the paging of memory to disk. If you decrease the sort heap, you might require an extra merge phase that could slow down the sort.

Default thresholds and actions

Table 42. Default Thresholds and Actions for dbms: Percent piped sorts rejected

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon
severe	Greater than 90%	Send Tivoli notice Change icon
warning	Greater than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule

Every 60 minutes without any time restrictions.

dbms: Percent sort heap allocated

Description

Returns the percentage of the allocated sort heap that the DB2 instance used during the monitoring interval.

CLI **dbmsPctSortHeapAlloc**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
If this returned percentage is high, the percentage of piped sorts rejected might be increasing. See “dbms: Percent piped sorts rejected” on page 174 for more information.

Default monitoring schedule
Every 60 minutes without any time restrictions.

dbms: Post threshold sorts

Description
Returns the number of sorts that requested heaps after the sort heap threshold was reached during the monitoring interval.

CLI **dbmsPostThreshSorts**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
If the returned value is high, you could improve your sort performance by increasing the sheapthres configuration parameter.

Sort heaps are allocated at the beginning of sorts and the beginning of merge phases. If at any time during a sort, a request to allocate a sort heap would exceed the sheapthres, then the sort is considered to be a post threshold sort. In post threshold sorts, the amount of sort heap allocated is smaller than the sortheap value.

dbms: Sort heap allocated (4 KB pages)

Description
Returns the total amount (in 4 KB pages) of sort heap currently allocated for all sorts in all active databases in the DB2 instance.

CLI **dbmsSortHeapAllocated**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to help you tune the sheapthres configuration parameter. If the returned value is greater than or equal to the sheapthres parameter, the sorts are not getting the full sort heap as defined by the sortheap configuration parameter.

dbms: Total piped sorts rejected

Description
Returns the total number of piped sorts that were rejected during the monitoring interval.

CLI **dbmsPipSrtsRej**

DB2 Sort Work Monitoring Collection

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

In the sort return phase, if the sorted information can return directly through the sort heap, it is a piped sort. However, even if the optimizer requests a piped sort, this request is rejected at run time if the total amount of sort heap memory for all sorts on the database is close to exceeding the sheapthres value.

If this returned value is high compared to the total number of sorts requested, you might consider decreasing your sort heap (using the sortheap configuration parameter) or increasing your sort heap threshold (using the sheapthres configuration parameter). However, be aware of the implications of these options. If you increase the sort heap threshold, more memory might remain allocated for sorting, causing the paging of memory to disk. If you decrease the sort heap, you might require an extra merge phase that could slow down the sort.

DB2 SQL Statement Activity Monitoring Collection

Collection name:

DB2_Stmt_Monitors

The monitoring sources in this collection provide statistics on SQL statement activity. SQL activity is a measure of database throughput and performance.

Recognizing the SQL pattern in a database can help you make decisions about tuning the database and database manager configuration parameters. For example, you might choose to tune certain parameters based on whether you are running query-only or update-intensive applications.

Table 43 lists the monitoring sources in this collection.

Table 43. Monitoring Sources in the DB2 SQL Statement Activity Monitoring Collection

GUI Name	CLI Name	Returned Value Format
conn: Application section inserts	connApplSectionInserts	Numeric
conn: Application section lookups	connApplSectionLookups	Numeric
conn: Binds/precompiles attempted	connBindsPrecompAttempted	Numeric
conn: Commits	connCommits	Numeric
conn: DDL SQL statements	connDdlSqlStmts	Numeric
conn: Dynamic SQL statements	connDynamicSqlStmts	Numeric
conn: Explicit commits	connCommitSqlStmts	Numeric
conn: Explicit rollbacks	connRollbackSqlStmts	Numeric
conn: Failed SQL statements	connFailedSqlStmts	Numeric
conn: Open block cursors	connOpenBlkCursors	Numeric
conn: Open cursors	connOpenCursors	Numeric
conn: Percent DDL SQL	connPctDdlSql	Percentage

Table 43. Monitoring Sources in the DB2 SQL Statement Activity Monitoring Collection (continued)

GUI Name	CLI Name	Returned Value Format
conn: Percent UID SQL	connPctUidSQL	Percentage
conn: Rollbacks	connRollbacks	Numeric
conn: Rows deleted	connRowsDeleted	Numeric
conn: Rows inserted	connRowsInserted	Numeric
conn: Rows selected	connRowsSelected	Numeric
conn: Rows updated	connRowsUpdated	Numeric
conn: Select SQL statements	connSelectSqlStmts	Numeric
conn: SQL requests since last commit	connSQLReqSinceLastCommit	Numeric
conn: Static SQL statements	connStaticSqlStmts	Numeric
conn: Update/Insert/Delete SQL	connUidSqlStmts	Numeric
db: Committed statements per second	dbCommitRate	Numeric
db: DDL SQL statements	dbDdlSqlStmts	Numeric
db: Dynamic SQL statements	dbDySqlStmts	Numeric
db: Failed SQL statements	dbFailedSqlStmts	Numeric
db: Percent DDL SQL	dbPctDdlSqlStmts	Percentage
db: Percent failed SQL	dbPctFailedSqlStmts	Percentage
db: Percent select SQL	dbPctSelectSqlStmts	Percentage
db: Percent UID SQL	dbPctUidSqlStmts	Percentage
db: Rollbacks per second	dbRollbackRate	Numeric
db: Select SQL statements	dbSelectSqlStmts	Numeric
db: SQL statements per second	dbSqlStmtRate	Numeric
db: Static SQL statements	dbStSqlStmts	Numeric
db: Update/Insert/Delete SQL	dbUidSqlStmts	Numeric

conn: Application section inserts

Description

Returns the number of inserts of SQL sections by an application from its SQL work area.

CLI **connApplSectionInserts** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 SQL Statement Activity Monitoring Collection

DB2 version

This monitoring source is supported for DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

The working copy of any executable section is stored in a unique SQL work area. The returned value is a count of how many times a copy was not available and had to be inserted. See “conn: Application section lookups” for more information on using sections.

conn: Application section lookups

Description

Returns the number of lookups of SQL sections by an application from its SQL work area.

CLI **connApplSectionLookups** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 UDB and DB2 Enterprise-Extended Edition.

Usage notes

Each agent has access to a unique SQL work area where the working copy of any executable section is kept. In partitioned databases, this work area is shared by all non-SMP agents. In other environments and with SMP agents, each agent has its own unique SQL work area.

The returned value indicates how many times the SQL work area was accessed by agents for an application. The returned value is a cumulative total of all lookups on all SQL work heaps for agents working for this application.

Use the returned value in conjunction with “conn: Application section inserts” on page 177 to tune the size of the heap used for the SQL work area. In partitioned databases this size is controlled by the app_ctl_heap_sz configuration parameter (see “db: app_ctl_heap_sz” on page 116). SQL work area size in other database environments use the appleheapsz configuration parameter (see “db: applheapsz” on page 117). The size of the SQL work area for SMP agents is controlled by appleheapsz in all environments.

conn: Binds/precompiles attempted

Description

Returns the number of binds and precompiles attempted by an application.

CLI **connBindsPrecompAttempted** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 UDB and DB2 Enterprise-Extended Edition.

Usage notes

Use the returned value to determine the current level of activity in the database manager.

The number that is returned does not include the count of Internal Automatic Rebinds, but it does include binds that are a result of the **REBIND PACKAGE** command (see "ECC_Rebind_Package" on page 229).

conn: Commits

Description

Returns the total number of SQL COMMIT statements that were attempted by the application. This value includes SQL COMMIT statements that are issued from the application and internal commits that are initiated by the database manager for the application.

CLI **connCommits** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

If this returned value has a small rate of change over a period of time, applications might not be doing frequent commits, leading to problems with logging and data concurrency.

This number can be used to calculate the total number of units of work (UOW) done by this application: UOW = COMMITs (total number) + ROLLBACK statements attempted + internal rollbacks. This calculation includes only those units of work done since the later of:

- The time of the connection to the database
- The last reset of the database monitor counters

The number of units of work is an indication of the level of database activity for this application.

conn: DDL SQL statements

Description

Returns the total number of SQL DDL statements that were executed by the application.

CLI **connDdlSqlStmts** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

DDL statements are expensive to run due to their impact on the system catalog tables. If this returned value is high, determine the cause and restrict this activity from being performed.

This value is also used to calculate the percentage of DDL activity for this application (see “conn: Percent DDL SQL” on page 184 for more information).

conn: Dynamic SQL statements

Description

Returns the total number of dynamic SQL statements that were attempted by the application.

CLI **connDynamicSqlStmts** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID
Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine the level of SQL statement activity attributable to this application.

conn: Explicit commits

Description
Returns the total number of SQL COMMIT statements that were explicitly attempted by the application.

CLI **connCommitSqlStmts** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID
Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
If this returned value has a small rate of change over a period of time, applications might not be doing frequent commits, leading to problems with logging and data concurrency.

This number can be used to calculate the total number of units of work (UOW) done by this application: UOW = COMMIT statements attempted + internal commits + ROLLBACK statements attempted + internal rollbacks. This calculation includes only those units of work done since the later of:

- The time of the connection to the database
- The last reset of the database monitor counters

The number of units of work is an indication of the level of database activity for this application.

conn: Explicit rollbacks

Description

Returns the total number of rollbacks explicitly attempted by the application. This value includes only the number of SQL ROLLBACK statements issued from the application.

CLI **connRollbackSqlStmts** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you determine the level of database activity for the application and the amount of conflict with other applications. Try to minimize the number of rollbacks, because higher rollback activity results in lower throughput for the database.

This number can be used to calculate the total number of units of work (UOW) done by this application: UOW = COMMIT statements attempted + internal commits + ROLLBACK statements attempted + internal rollbacks. This calculation includes only those units of work done since the later of:

- The time of the connection to the database
- The last reset of the database monitor counters

The number of units of work is an indication of the level of database activity for this application.

conn: Failed SQL statements

Description

Returns the total number of failed SQL statements for the application.

CLI **connFailedSqlStmts** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you determine reasons for poor performance. Failed statements indicate that time is wasted by the database manager and result in lower throughput for the database.

conn: Open block cursors

Description

Returns the total number of blocking cursors (local and remote) that are currently open for the application.

CLI **connOpenBlkCursors** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value along with the value from the “conn: Open cursors” monitoring source to calculate the percentage of cursors that are blocking cursors. If the percentage is low, you could improve performance by improving the row blocking in the application. You can improve row blocking by:

- Checking the pre-compile options for record blocking for treatment of ambiguous cursors.
- Redefining cursors to allow for blocking (for example, specify FOR FETCH ONLY on your cursors).

conn: Open cursors

Description

Returns the total number of cursors (local and remote) that are currently open for the application. This value also includes the number of local and remote blocking cursors currently open for the application.

CLI **connOpenCursors** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value along with the value from the “conn: Open block cursors” monitoring source to calculate the percentage of cursors that are blocking cursors. If the percentage is low, you could improve performance by improving the row blocking in the application (see “conn: Open block cursors” on page 183 for more information).

conn: Percent DDL SQL

Description

Returns the percentage of total SQL statements that were SQL DDL statements executed by the application during the monitoring interval.

CLI **connPctDdlSql** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See “Appendix. DB2 Snapshot Application ID Formats” on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

DDL statements are expensive to run because they cause a lot of activity in the system catalog tables. If this returned value is high, determine the cause and restrict this activity from being performed.

You can also use the returned value to help you tune the package cache hit ratio for this application. DDL statements can also impact the package cache, by invalidating sections that are stored there and causing additional system overhead due to section recompilation.

Examples of DDL statements are CREATE TABLE, CREATE VIEW, ALTER TABLE, and DROP INDEX.

conn: Percent UID SQL

Description

Returns the percentage of total SQL statements that were SQL UPDATE, INSERT, and DELETE statements executed by the application during the monitoring interval.

CLI **connPctUidSQL** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID
Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to evaluate whether the application is heavy on updates. If the returned value is low, the application is query-based. Otherwise, it is update-based.

Knowing what type of applications you have (query-based or update-based) can aid you in tuning the database configuration parameters.

conn: Rollbacks

Description
Returns the total number of rollbacks attempted by the application. This value includes SQL ROLLBACK statements that are issued from the application and internal rollbacks that are initiated by the database manager for the application.

CLI **connRollbacks** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID
Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to help you determine the level of database activity for the application, and the amount of conflict with other applications. Try to minimize the number of rollbacks, because higher rollback activity results in lower throughput for the database.

This number can be used to calculate the total number of units of work (UOW) done by this application: UOW = COMMIT statements attempted + internal commits + ROLLBACKs (total). This calculation includes only those units of work done since the later of:

- The time of the connection to the database
- The last reset of the database monitor counters

The number of units of work is an indication of the level of database activity for this application.

conn: Rows deleted

Description

Returns the total number of row deletions attempted by the application. This value includes the number of rows deleted from the database as a result of internal database activity performed for the application.

CLI **connRowsDeleted** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value as an indication of the level of database activity generated by the application.

conn: Rows inserted

Description

Returns the total number of row inserts attempted by the application. This value includes the number of rows inserted into the database as a result of internal database activity performed for the application.

CLI **connRowsInserted** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value as an indication of the level of database activity generated by the application.

conn: Rows selected

Description

Returns the number of rows that were selected and returned to the application.

CLI **connRowsSelected** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value as an indication of the level of database activity generated by the application. This value does not include a count of rows read for actions such as COUNT(*) or joins.

conn: Rows updated

Description

Returns the total number of row updates attempted by the application. This value includes the number of rows updated in the database as a result of internal database activity performed for the application.

CLI **connRowsUpdated** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 SQL Statement Activity Monitoring Collection

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value as an indication of the level of database activity generated by the application.

conn: Select SQL statements

Description
Returns the number of SQL SELECT statements that were executed by the application.

CLI **connSelectSqlStmts** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID
Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine the level of database select SQL statement activity generated by the application.

conn: SQL requests since last commit

Description
Returns the number of SQL requests that were submitted by the application since the last commit.

CLI **connSQLReqSinceLastCommit** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID
Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version
This monitoring source is supported for DB2 UDB and DB2 Enterprise-Extended Edition.

Usage notes

Use the returned value to monitor the progress of a transaction.

conn: Static SQL statements

Description

Returns the number of static SQL statements that were attempted by the application.

CLI **connStaticSqlStmts** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine the level of SQL statement activity attributable to this application.

conn: Update/Insert/Delete SQL

Description

Returns the number of SQL UPDATE, INSERT, and DELETE statements that were executed by the application.

CLI **connUidSqlStmts** *DB2_snapshot_application_ID*

Arguments

DB2_snapshot_application_ID

Specify the application ID for the application. This identifier is generated when the application connects to the database and is unique across the network.

There are different formats for the application ID, which are dependent on the communication protocol between the client and the server machine on which the database manager is running. See "Appendix. DB2 Snapshot Application ID Formats" on page 259 for information on these formats.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine the level of database update activity generated by the application.

The returned value is also used in calculations for the “conn: Percent UID SQL” monitoring source (page 184).

db: Committed statements per second

Description

Returns the rate (counts per second) at which unit-of-work commits were attempted for the database during the monitoring interval. Unit-of-work commits include SQL COMMIT statements that are issued from applications and internal commits that are initiated by the database manager.

CLI **dbCommitRate**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to help you determine rates of database activity.

db: DDL SQL statements

Description

Returns the number of SQL DDL statements that were executed in the database during the monitoring interval.

CLI **dbDdlSqlStmts**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

DDL statements are expensive to run due to their impact on the system catalog tables. If this returned value is high, determine the cause and restrict this activity from being performed.

This value is also used to calculate the percentage of DDL activity for the database (see “db: Percent DDL SQL” on page 191 for more information).

db: Dynamic SQL statements

Description

Returns the number of dynamic SQL statements that were attempted for the database during the monitoring interval.

CLI **dbDySqlStmts**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine the level of SQL statement activity in the database.

db: Failed SQL statements

Description

Returns the number of failed SQL statements for the database during the monitoring interval.

CLI **dbFailedSqlStmts**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to help you determine reasons for poor performance. Failed statements indicate time is wasted by the database manager and result in lower throughput for the database.

db: Percent DDL SQL

Description
Returns the percentage of total SQL statements that were SQL DDL statements during the monitoring interval.

CLI **dbPctDdlSqlStmts**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
DDL statements are expensive to run due to their impact on the system catalog tables. If this returned value is high, determine the cause and restrict this activity from being performed.

DDL statements can also impact the package cache, by invalidating sections that are stored there and causing additional system overhead due to section recompilation.

Examples of DDL statements are CREATE TABLE, CREATE VIEW, ALTER TABLE, and DROP INDEX.

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: Percent failed SQL

Description
Returns the percentage of total SQL statements that failed during the monitoring interval.

CLI **dbPctFailedSqlStmts**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value can help you determine reasons for poor performance, because failed statements indicate time wasted by the database manager and result in lower throughput for the database.

Default thresholds and actions

Table 44. Default Thresholds and Actions for db: Percent failed SQL

Response Level	Trigger When	Default Actions
critical	Greater than 95%	Send Tivoli notice Change icon

Table 44. Default Thresholds and Actions for db: Percent failed SQL (continued)

Response Level	Trigger When	Default Actions
severe	Greater than 90%	Send Tivoli notice Change icon
warning	Greater than 80%	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: Percent select SQL

Description
Returns the percentage of total SQL statements that were SQL SELECT statements during the monitoring interval.

CLI **dbPctSelectSqlStmts**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine the level of application activity and throughput for the database.

db: Percent UID SQL

Description
Returns the percentage of total SQL statements that were SQL UPDATE, INSERT, and DELETE statements during the monitoring interval.

CLI **dbPctUidSqlStmts**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine the level of database data change activity.

db: Rollbacks per second

Description
Returns the rate (counts per second) at which unit-of-work rollbacks were attempted during the monitoring interval. Unit-of-work rollbacks include SQL ROLLBACK statements that are issued from applications and internal rollbacks that are initiated by the database manager.

CLI **dbRollbackRate**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to help you determine the rates of database activity

and the amount of conflict in the database. Try to minimize the number of rollbacks, because higher rollback activity results in lower throughput for the database.

db: Select SQL statements

Description
Returns the number of SQL SELECT statements that were executed during the monitoring interval.

CLI **dbSelectSqlStmts**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine the level of database activity.

The returned value is also used in calculations for the “db: Percent select SQL” monitoring source (page 192).

db: SQL statements per second

Description
Returns the rate (statements per second) at which SQL statements were executed during the monitoring interval.

CLI **dbSqlStmtRate**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
The returned value indicates the rate of database SQL statement activity.

db: Static SQL statements

Description
Returns the number of static SQL statements that were attempted during the monitoring interval.

CLI **dbStSqlStmts**

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value to determine the level of SQL statement activity in the database.

db: Update/Insert/Delete SQL

Description
Returns the number of SQL UPDATE, INSERT, and DELETE statements that were executed during the monitoring interval.

CLI **dbUidSqlStmts**

DB2 SQL Statement Activity Monitoring Collection

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value to determine the level of database activity.

The returned value is also used in calculations for the “db: Percent UID SQL” monitoring source (page 192). The result can be useful for analyzing activity and throughput.

DB2 Table Monitoring Collection

Collection name:

DB2_Table_Monitors

The monitoring sources in this collection provide information on:

- Invalid packages
- Table rows (read, written, and overflowed)
- The last time the monitor counters were reset at a table header level
- Whether tables need reorganization

Table 45 lists the monitoring sources in this collection.

Table 45. Monitoring Sources in the DB2 Table Monitoring Collection

GUI Name	CLI Name	Returned Value Format
db: Number of all invalid packages	dbInvalidPkgAll	Numeric
db: Number of invalid SYSTEM packages	dbInvalidPkgSystem	Numeric
db: Number of invalid user packages	dbInvalidPkgUser	Numeric
table: Overflow accesses	tableOverflowAcc	Numeric
table: Reorg needed	tableReorgChkOne	String: YY YN NY NN
table: Rows read per second	tableRowsRead	Numeric
table: Rows written per second	tableRowsWritten	Numeric

db: Number of all invalid packages

Description

Returns the number of all packages that are currently marked invalid in the database.

CLI **dbInvalidPkgAll**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value as an indication of the current number of invalid

packages. A package is marked invalid if it depends on an object (for example, a table) and that object is dropped.

The number of invalid packages can indicate how many automatic rebinds are necessary in the database. An invalid package is usually automatically rebound the next time it is accessed, unless it was marked invalid because a trigger was dropped or because the dropped object was not re-created. Automatic rebinds can significantly lower performance and should be minimized where possible.

Default thresholds and actions

Table 46. Default Thresholds and Actions for db: Number of all invalid packages

Response Level	Trigger When	Default Actions
critical	Greater than 20	Send Tivoli notice Change icon
severe	Greater than 15	Send Tivoli notice Change icon
warning	Greater than 1	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule

Every 60 minutes without any time restrictions.

db: Number of invalid SYSTEM packages

Description

Returns the number of SYSTEM packages that are currently marked invalid in the database.

CLI **dbInvalidPkgSystem**

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

Use the returned value as an indication of the current number of invalid packages owned by SYSTEM. A package is marked invalid if it depends on an object (for example, a table) and that object is dropped.

The number of invalid packages can indicate how many automatic rebinds are necessary in the database. An invalid package is usually automatically rebound the next time it is accessed, unless it was marked invalid because a trigger was dropped or because the dropped object was not re-created. Automatic rebinds can significantly lower performance and should be minimized where possible.

Default thresholds and actions

Table 47. Default Thresholds and Actions for db: Number of invalid SYSTEM packages

Response Level	Trigger When	Default Actions
critical	Greater than 20	Send Tivoli notice Change icon

Table 47. Default Thresholds and Actions for db: Number of invalid SYSTEM packages (continued)

Response Level	Trigger When	Default Actions
severe	Greater than 15	Send Tivoli notice Change icon
warning	Greater than 1	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule
Every 60 minutes without any time restrictions.

db: Number of invalid user packages

Description
Returns the number of packages owned by a user that are currently marked invalid in the database.

CLI **dbInvalidPkgUser**

Arguments

Owner_name
Specify the owner of the package.

DB2 version
This monitoring source is supported for DB2 for common servers and later versions.

Usage notes
Use the returned value as an indication of the current number of invalid packages owned by the user. A package is marked invalid if it depends on an object (for example, a table) and that object is dropped.

The number of invalid packages can indicate how many automatic rebinds are necessary in the database. An invalid package is usually automatically rebound the next time it is accessed, unless it was marked invalid because a trigger was dropped or because the dropped object was not re-created. Automatic rebinds can significantly lower performance and should be minimized where possible.

table: Overflow accesses

Description
Returns the number of accesses (reads and writes) made to overflowed rows of this table.

CLI **tableOverflowAcc** *Owner_name Table_name*

Arguments

Owner_name
Specify the owner of the table.

Table_name
Specify the name of the table.

Usage notes
Overflowed records indicate that data fragmentation occurred. If this

number is high, you might be able to improve table performance by reorganizing the table. See “table: Reorg needed” for a way to tell if a table needs reorganizing.

A row overflows if it is updated and no longer fits in the data pages where it was originally written. Row overflow usually happens as a result of an update of a VARCHAR column or from an ALTER TABLE statement.

table: Reorg needed

Description

Returns a string indicating whether the specified table or its indexes need to be reorganized.

CLI **tableReorgChkOne** *Owner_name Table_name*

Arguments

Owner_name
Specify the name of the table owner.

Table_name
Specify the name of the table.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

The returned value is a two-character string that indicates whether the table or any of its indexes need to be reorganized:

YN The table needs to be reorganized.

NY At least one of the indexes needs to be reorganized.

YY The table and at least one of the indexes need to be reorganized.

Reorganizing a table can improve access performance. When you reorganize a table, the table data is rearranged into a physical sequence, usually according to a specified index. As a result, SQL statements on that data can be processed more efficiently. In addition, the reorganization process removes unused, empty space from the table, and the reorganized table is stored more compactly.

To calculate the returned value, DB2 ECC uses statistics from six DB2 formulas to determine if performance has deteriorated or can be improved by reorganization (for more information, see the **REORGCHK** command in the *DB2 Command Reference* for the version of DB2 that you are using):

1. Formulas F1, F2, and F3 are used to check the table. If the results indicate that the table should be reorganized, the first character of the returned string is a Y.
2. Formulas F4, F5, and F6 are used to check the indexes one by one. If the results indicate that at least one of the indexes should be reorganized, the second character of the returned string is a Y. If the table has no indexes, the second character is always an N.

Current table statistics are used when calculating the results. The DB2 recommended results are:

F1 Less than 5

F2 Greater than 70

F3 Greater than 80

F4 Greater than 80

F5 Greater than 50

F6 Less than 100

Along with the returned value, DB2 ECC writes additional information in a comment in the Tivoli notice or message. The first line of the comment lists the table owner, table name, and the results of formulas F1, F2, and F3 (*Table_owner.Table_name*=(resultF1resultF2resultF3)). Subsequent lines of the comment list the index owner, index name, and the results of formulas F4, F5, and F6 (*Index_owner.Index_name*=(resultF4resultF5resultF6)). One line is listed for each index.

Example:

Suppose you have a table with owner USERID, table name TABLE1, and two indexes, INDX1 and INDX2. INDX1 has the owner USERID and INDX2 has the owner DBADMIN. You use the “table: Reorg needed” monitoring source to create a monitor that sends you a notice if the returned string contains a Y (when the table or either index needs reorganizing).

Table 48 shows the results for each formula for the table and indexes.

Table 48. Results of the Reorganize Formulas for TABLE1, INDX1, and INDX2

Name	F1	F2	F3	F4	F5	F6
TABLE1	29	58	60			
INDX1				97	64	11
INDX2			97	48	77	

Both TABLE1 and INDX2 need reorganizing; a notice is sent that contains the following lines:

```
YY
 USERID.TABLE1 = (295860)
 USERID.INDX1 = (976411)
 DBADMIN.INDX2 = (974877)
```

Default thresholds and actions

Table 49. Default Thresholds and Actions for table: Reorg needed

Response Level	Trigger When	Default Actions
critical	N/A	None
severe	N/A	None
warning	Matches ’Y’ (in any position)	Send Tivoli notice Change icon
normal	N/A	None
always	N/A	None

Default monitoring schedule

Every 24 hours without any time restrictions.

table: Rows read per second

Description

Returns the rate at which rows were read from the table during the monitoring interval.

CLI **tableRowsRead** *Owner_name Table_name*

Arguments

Owner_name
Specify the owner of the table.

Table_name
Specify the name of the table.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

A high returned value indicates that there is heavy usage of the table. You might want to create additional indexes for the table. To avoid the maintenance of unnecessary indexes, use the SQL EXPLAIN statement, described in the *DB2 Administration Guide* for the version of DB2 that you are using, to determine if your packages use an index for the table.

This count includes the value returned by the “table: Overflow accesses” monitoring source.

table: Rows written per second

Description

Returns the rate at which rows were changed (inserted, deleted, or updated) in the table during the monitoring interval.

CLI **tableRowsWritten** *Owner_name Table_name*

Arguments

Owner_name
Specify the owner of the table.

Table_name
Specify the name of the table.

DB2 version

This monitoring source is supported for DB2 for common servers and later versions.

Usage notes

A high returned value indicates that there is heavy usage of the table. You might want to use the DB2 Run Statistics (RUNSTATS) utility to maintain the efficiency of the packages used for this table.

Chapter 2. DB2 ECC Task Libraries

The DB2 ECC task libraries are collections of predefined management tasks, such as backing up databases and running statistics. Running, scheduling, or automatically executing these predefined tasks helps ease overall management workload in a complex environment by carrying out actions against all subscribed task endpoints without having to define the task for each endpoint separately.

The tasks are grouped into three libraries based on function:

Monitor and Administration Tasks
Contains tasks used for DB2 ECC monitoring and administration.

DB2 Database Tasks
Contains tasks used with DB2 and the database manager.

Data Replication Tasks
Contains tasks used for data replication with IBM Replication (previously DPROPR).

This chapter also briefly explains how to run DB2 ECC tasks.

DB2 ECC supports tasks run with DPROPR, which works with DB2 for common servers, and IBM Replication, which is an integrated component of DB2 UDB and DB2 Enterprise-Extended Edition. You can run tasks using either DPROPR or IBM Replication with DB2 for common servers, DB2 UDB, or DB2 Enterprise-Extended Edition endpoints. You can specify the correct replication version using the *drep_vers* and *drep_inst* variables. You can specify the version of DB2 you want to use by selecting the correct endpoint. For more information on the *drep_vers* and *drep_inst* variables, see the Arguments section of the task you want to run.

During installation, the default IDs and groups for each task are set. Most tasks default to the db2ecc ID, and on AIX, some default to the root ID.

Running DB2 ECC Tasks

Tasks run on database, instance, managed node, partition, and partition group endpoints. The databases must be local on the managed node.

You can run a DB2 ECC task directly, or have a DB2 ECC task run automatically as the result of a monitor. In this chapter, the Argument section of a task description describes the arguments needed for that task, if any.

Running DB2 ECC Tasks Directly

Using the TME 10 desktop, you can run or schedule a DB2 ECC task. When you run or schedule a DB2 ECC task, you specify values for its arguments in the Task Library window. See the *DB2 Enterprise Control Center for TME 10 User's Guide* for information on the Task Library window and instructions on how to run and schedule tasks.

Running DB2 ECC Tasks from Monitors

To automatically run a DB2 ECC task as an action to be taken when the specified condition for a monitor is met, you must select the **Run program** check box in the Edit Sentry Monitor window and type this command:

```
wecctlib Task_name argument1 argument2 ... argument
```

where wecctlib is one of the following commands:

wecctlib
Is the DB2 ECC program that runs tasks for the DB2 ECC monitors, except for the data replication tasks on a Windows NT host.

wecctlib2
Is the DB2 ECC program that runs all data replication tasks for the DB2 ECC monitors on a Windows NT host.

For additional information on the **wecctlib2** command, see "Data Replication Tasks" on page 244.

Task_name
Is the name of the DB2 ECC task that you want run when the condition for the monitor is met. For example, ECC_Quiesce_Tablespaces.

argument1 argument2 **...** *argument*
Are the arguments for the specified DB2 ECC task.

For details on the Edit Sentry Monitor window, see the *DB2 Enterprise Control Center for TME 10 User's Guide*.

Example of Running a Task from a Monitor

If you have a monitor based on the "table: Reorg needed" monitoring source, and you want the table (schema USERID and name TABLE1) reorganized automatically when a particular threshold is reached, you enter the following command for **Run program**:

```
wecctlib ECC_Reorganize_Table USERID TABLE1
```

Syntax

In this chapter, the Run program syntax (from a monitor) section of a task description shows the syntax of the command that you need to enter for that particular task. In addition:

- You must specify all arguments in the Run program syntax (from a monitor) section in the order in which they are shown.
- Any quotation marks shown are required.
- Brackets around arguments ([*argument1 argument2*]) indicate those arguments are optional.
- If you do not want to include a variable but want to use a variable that follows it, you must use NONE as the variable place holder. For example, if the command and variables are:

  ```
  wecctlib ECC_Backup_Database username password online adsm adsmnum target
  numbuf bufsize parallelism libname tablespace
  ```

  And you do not want to include the *username* and *password* variables, but you do want to specify the *online* variable, you might enter the following command:

  ```
  wecctlib ECC_Backup_Database NONE NONE YES YES 2 NONE 4 1024
  ```

Monitor and Administration Tasks

Library name:
DB2_ECC-AdminTasks

This section provides information about the function and uses of the tasks in the DB2_ECC-AdminTasks library. The tasks are listed alphabetically.

For more information on these tasks, see the *TME 10 Framework Reference Manual*.

Table 50 lists the tasks in this library and the endpoints they run on.

Table 50. Monitoring and Administration Task Library and Task Endpoints

Task	Description	Task Endpoints*				
		DB	DB Part.	DB Part. Grp.	Inst.	Man. Node
ECC_Broadcast_Message	Broadcast a message to all TME desktops	✓	✓	✓	✓	✓
ECC_Reset_Counters	Reset the ECC counters for a DB2 instance		✓		✓	
ECC_Send_Notice	Send a notice to all TME administrators	✓	✓	✓	✓	✓
ECC_Set_Up_Monitoring_Agent	Set up the monitoring agent	✓	✓	✓	✓	✓
ECC_Start_DB2SNMP_Agent	Start the DB2 SNMP agent	✓	✓	✓	✓	
ECC_Start_Monitoring_Agent	Start the ECC monitoring agent	✓	✓	✓	✓	✓
ECC_Stop_DB2SNMP_Agent	Stop a DB2 SNMP daemon	✓	✓	✓	✓	✓
ECC_Stop_Monitoring_Agent	Stop the ECC monitoring agent	✓	✓	✓	✓	✓

*Task Endpoints
DB= database, DB Part. = DB2 partition, DB Part. Grp. = DB2 partition group, Inst. = instance (partitioned or nonpartitioned), Man. Node = managed node

ECC_Broadcast_Message

Description
Broadcasts a message to all TME 10 desktops.

TMR roles

- Super
- Senior
- Admin

Run program syntax (from a monitor)

```
wecctlib ECC_Broadcast_Message "message"
```

Arguments

message
Specify the message that you want to broadcast. You must specify the quotation marks around the message.

Example

```
wecctlib ECC_Broadcast_Message "Backing up databases now."
```

DB2 version
This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes
This task reads the input data and broadcasts the message text to all desktops in the TME 10 installation.

You should select only one endpoint in your TME. If you select multiple endpoints, the message is sent multiple times. For example, if you select three endpoints, the message is sent to all endpoints in the TME three times.

For more information, see the **wbroadcast** command in the *TME 10 Framework Reference Manual*.

ECC_Reset_Counters

Description
Resets the ECC counters for a DB2 instance.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Reset_Counters
```

DB2 version
This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes
Use this task to force counters to reset that are normally only reset when an application or the database manager is started.

This task must run on a DB2 instance endpoint.

For more information, see “dbms: Last reset timestamp” on page 38.

ECC_Send_Notice

Description
Sends a notice to the notification server.

TMR roles

- Super
- Senior
- User

Run program syntax (from a monitor)

```
wecctlib ECC_Send_Notice "notice_group" priority "notice"
```

Arguments

notice_group
Specify the notice group to be used. You must specify the quotation marks around the name.

priority Specify the priority to be used for the notification:

- Critical
- Error
- Warning
- Notice
- Debug

notice Specify the notice you want sent.

Example

```
wecctlib ECC_Send_Notice "TME Administration" Critical "Shutting down
for maintenance."
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

This task translates the input data into a message structure and sends it to the notification server. You can specify that the message be sent to any notice group.

For more information, see the **wsndnotif** command in the *TME 10 Framework Reference Manual*.

ECC_Set_Up_Monitoring_Agent

Description

Sets up the DB2 ECC monitoring agent.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

`wecctlib ECC_Set_Up_Monitoring_Agent` *`action`*

Arguments

action Specify one of the following actions:

AddStart
Add an inittab entry for UNIX or install the DB2 ECC monitoring service for NT, then start the monitoring agent.

AddNoStart
Add an inittab entry for UNIX or install the DB2 ECC monitoring service for NT.

Password
Windows NT only: Specify the password that will be used when installing the monitoring service.

RemoveStop
Remove the inittab entry for UNIX.

RemoveNoStop
Remove the inittab entry for UNIX or remove the DB2 ECC monitoring service for NT.

User ID
Windows NT only: Specify the user ID that will be used when installing the monitoring service.

Example

```
wecctlib ECC_Set_Up_Monitoring_Agent AddNoStart
```

DB2 version
This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes
For more information on the DB2 ECC monitoring agent, see “DB2 ECC monitoring communication agent status” on page 15.

ECC_Start_DB2SNMP_Agent

Description
Starts a DB2 SNMP daemon to run any DB2 ECC SNMP monitors.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Start_DB2SNMP_Agent hostname community_name
```

Arguments

hostname
Specify the name of the host on which SNMP is to be started.

community_name
Specify the community name. The community name is the identifier used to group together a set of SNMP agents that are managed together.

Example

```
wecctlib ECC_Start_DB2SNMP_Agent mysystem public
```

DB2 version
This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes
This task runs only on AIX.

For more information on SNMP monitors, see “DB2 SNMP Monitoring Collection” on page 159 .

ECC_Start_Monitoring_Agent

Description

Starts the DB2 ECC monitoring agent to run DB2 monitors.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Start_Monitoring_Agent
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

For more information on the DB2 ECC monitoring agent, see “DB2 ECC monitoring communication agent status” on page 15.

ECC_Stop_DB2SNMP_Agent

Description

Stops the DB2 SNMP daemon.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Stop_DB2SNMP_Agent
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

This task runs only on AIX. The DB2 SNMP daemon at the managed node is terminated using the **KILL** command. There is no DB2 SNMP command to terminate the daemon.

For more information on SNMP monitors, see “DB2 SNMP Monitoring Collection” on page 159 .

ECC_Stop_Monitoring_Agent

Description

Stops the DB2 ECC monitoring agent.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Stop_Monitoring_Agent
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Monitor and Administration Tasks

For more information on the DB2 ECC monitoring agent, see “DB2 ECC monitoring communication agent status” on page 15.

DB2 Database Tasks

Library name:
DB2_ECC-DatabaseTasks

This section provides information about the tasks in the DB2_ECC-DatabaseTasks library, which provide basic administration functions for DB2 and the database manager.

The tasks are listed alphabetically.

For more information on these tasks, see the:

- *DB2 API Reference*
- *DB2 Command Reference*
- *DB2 SQL Reference*

for the version of DB2 that you are using.

Table 51 lists the tasks in this library and the endpoints they run on.

Table 51. DB2 Database Task Library and Task Endpoints

Task	Description	Task Endpoints*				
		DB	DB Part.	DB Part. Grp.	Inst.	Man. Node
ECC_Activate_Database	Activate a database	✓	✓	✓		
ECC_Alter_Bufferpool	Modify the size of the bufferpool	✓	✓	✓		
ECC_Backup_Database	Create a backup of a DB2 database	✓	✓	✓		
ECC_Create_Admin_Server	Create the DB2 administration server	✓			✓	
ECC_Create_Explain_Tables	Create explain tables	✓	✓	✓		
ECC_Create_Sample_Database	Create a DB2 sample database	✓	✓	✓	✓	
ECC_Deactivate_Database	Deactivate the database	✓	✓	✓		
ECC_Drop_Admin_Server	Delete the DB2 administration server instance	✓			✓	
ECC_Drop_Sample_Database	Drop the DB2 sample database	✓	✓	✓	✓	
ECC_Force_All_Applications	Force all DB2 applications off the system	✓	✓	✓	✓	
ECC_Force_Applications	Force specific DB2 applications off the system	✓	✓	✓	✓	
ECC_Get_Admin_ Configuration	Get the DB2 administration configuration	✓	✓		✓	
ECC_Get_Admin_Server_Instance	Get the DB2 administration server instance name	✓	✓	✓	✓	

Table 51. DB2 Database Task Library and Task Endpoints (continued)

Task	Description	Task Endpoints*				
		DB	DB Part.	DB Part. Grp.	Inst.	Man. Node
ECC_Get_Database_Configuration	Return the values of individual entries in the database configuration file	✓	✓	✓		
ECC_Get_Database_Manager_ Configuration	Return the values of individual entries in the database manager configuration file	✓	✓		✓	
ECC_Invoke_Stored_Procedure	Invoke a DB2 stored procedure	✓	✓	✓		
ECC_List_Applications	List the DB2 applications	✓	✓	✓	✓	
ECC_List_Backup_Recovery_File	List the DB2 backup and recovery history file	✓	✓	✓		
ECC_List_DCS_Applications	List the DB2 DCS applications	✓	✓	✓	✓	
ECC_List_Node_Directory	List the contents of the node directory	✓	✓		✓	
ECC_Prune_Recovery _ History_File	Prune the DB2 recovery history file	✓	✓	✓		
ECC_Quiesce_Tablespaces	Quiesce DB2 table spaces	✓	✓	✓		
ECC_Rebind_All_Packages	Rebind all DB2 packages	✓	✓	✓		
ECC_Rebind_Package	Rebind a specific DB2 package	✓	✓	✓		
ECC_Reorganize_Table	Reorganize a DB2 table	✓	✓	✓		
ECC_Reorgchk	Run the DB2 REORGCHK command	✓	✓	✓		
ECC_Reset_Admin_Configuration	Reset the parameters in the database manager configuration file, relevant to the DB2 administration server, to the system defaults	✓	✓		✓	
ECC_Reset_Database_ Configuration	Reset the configuration of a specific database to the system defaults	✓	✓	✓		
ECC_Reset_Database_ Manager_Configuration	Reset the parameters in the database manager configuration file to the system defaults	✓	✓		✓	
ECC_Restart_Database	Restart a DB2 database	✓	✓	✓		
ECC_Run_Statistics	Run DB2 statistics	✓	✓	✓		
ECC_Set_Admin_Server_ID	Establish or modify the user account associated with the DB2 administration server instance	✓	✓	✓	✓	
ECC_Start_Admin_Server	Start the DB2 administration server	✓	✓	✓	✓	
ECC_Start_DB2	Start DB2	✓	✓	✓	✓	
ECC_Start_DB2_NT_Security	Start the DB2 security service on a Windows NT system	✓			✓	✓

DB2 Database Tasks

Table 51. DB2 Database Task Library and Task Endpoints (continued)

Task	Description	Task Endpoints*				
		DB	DB Part.	DB Part. Grp.	Inst.	Man. Node
ECC_Stop_Admin_Server	Stop the DB2 administration server	✓	✓	✓	✓	
ECC_Stop_DB2	Stop DB2	✓	✓	✓	✓	
ECC_Stop_DB2_NT_Security	Stop the DB2 security service on a Windows NT system	✓			✓	✓
ECC_Update_Admin_Configuration	Modify individual entries in the database manager configuration file that are relevant to the DB2 administration server	✓	✓		✓	
ECC_Update_Database_ Configuration	Update the DB2 database configuration	✓	✓	✓		
ECC_Update_Database_ Manager_Configuration	Update the DB2 database manager configuration	✓	✓		✓	

*Task Endpoints
DB= database, DB Part. = DB2 partition, DB Part. Grp. = DB2 partition group, Inst. = instance (partitioned or nonpartitioned), Man. Node = managed node

ECC_Activate_Database

Description

Activates the database that you specify and starts all necessary database services so that the database is available for connection and use by any application.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Activate_Database [username password]
```

Arguments

username
Specify the name of the user starting the database. This variable is optional, but it is required if you specify *password*.

password
Specify the password for the *username*. This variable is optional, but it is required if you specify *username*.

Example

```
wecctlib ECC_Activate_Database joe joepw
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

When this task is run on DB2 Enterprise-Extended Edition, it activates the specified database on all nodes within the system.

If a database has not been started and a **CONNECT TO** command is encountered in an application, the application must wait while the database manager starts the required database. However, after the database is started, other applications can connect and use it without starting it up.

Databases initialized by ECC_Activate_Database can be shut down only by running a deactivate database command (such as ECC_Deactivate_Database).

For more information, see the **ACTIVATE DATABASE** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Alter_Bufferpool

Description

Used to: modify the bufferpool size for a single node or all nodes; turn the use of extended storage on or off; and add the bufferpool definition to a new node group.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

`wecctlib ECC_Alter_Bufferpool` *`name action value`* [*`username password`*]

Arguments

name Specify the name of the bufferpool that will be altered.

action

Specify one of the following arguments:

ALL Specify to alter the bufferpool size on all nodes that use the default bufferpool size (that do not have a size specified in the *except-on-nodes-clause*). For more information, see the **CREATE BUFFERPOOL** command in the *DB2 SQL Reference*.

NODE Specify to alter the size of the bufferpool for the DB2 node represented by the endpoint you selected.

SET Specify to set the extended storage action

ADD Specify to add the bufferpool definition to a node group.

value Specify one of the following arguments:

size If you specify **ALL** or **NODE** for the *action* variable, type the size of the bufferpool (specified as the number of pages). If `-1` is specified, the bufferpool size is taken from the BUFFPAGE database configuration parameter.

cache If you specified **SET** for the *action* variable, specify one of the following parameters:

YES If the extended storage configuration is turned on, pages that are migrating out of this bufferpool will be cached in the extended storage.

NO Even if the extended storage configuration is turned on, pages that are migrating out of this bufferpool will not be cached in the extended storage.

ng_name
If you specified **ADD** for the *action* variable, specify the nodegroup name this bufferpool definition will be added to. The nodegroup name is added to all nodes in the specified node group if it currently does not exist. If the bufferpool is defined on a node, no action is taken.

username
Specify the name of the user to attach to a remote partition. If you run this task with DB2 Enterprise-Extended Edition on a partition group endpoint and at least one of the partitions in the partition group is on a remote host, you must specify a user name. The user name is used for all nodes included in the partition group whether they are on a local or remote host.

password
Specify the password used to authenticate with the *username* variable.

Example

```
wecctlib ECC_Alter_Bufferpool BUFF2 20 6
wecctlib ECC_Alter_Bufferpool BUFF2 STORAGE YES
```

DB2 version
This task runs on DB2 UDB and DB2 Enterprise-Extended Edition.

For more information, see the **ALTER BUFFERPOOL** statement in the *DB2 SQL Reference*.

ECC_Backup_Database

Description
Creates a backup copy of a database.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)
`wecctlib ECC_Backup_Database` *`username password online adsm adsmnum target numbuf bufsize`* [*`parallelism libname tablespace`*]

Arguments

username
Specify the name with which to back up the database. If you run this task with DB2 Enterprise-Extended Edition on a partition group endpoint and at least one of the partitions in the partition group is on a remote host, you must specify a user name. The user name is used for all nodes included in the partition group whether they are on a local or remote host.

NONE Specifies that the current user name is used.

username
Specify a particular user name.

password
Specify the password for *username*:

NONE Specifies that no password is supplied. Use NONE if the *username* is NONE.

password
Specify the password used to authenticate the user name entered for *username*. This value is required if you specify a user name.

online Specify **YES** or **NO** to indicate whether the backup is online.

adsm Specify **YES** or **NO** to indicate whether to use managed output (managed by ADSM or a vendor product) for the backup.

adsmnum
Specify the number of I/O sessions to be used with managed output (managed by ADSM or a vendor product). A typical value to use for *adsmnum* is 2.

If you are not using managed output, specify any number (it will be ignored).

target Specify a directory or tape device name to which the backup is written. If you specify a directory, you must enter the full path on which the directory resides. The directory must already exist.

numbuf
Specify the number of buffers to use during the backup process. A typical value to use for *numbuf* is 1.

bufsize
Specify the number of pages for the buffer that is used when building the backup image. The minimum size is 16 pages. A typical value to use for *bufsize* is 1024 (1024 pages).

parallelism
Specify the number of buffer manipulators to spawn during the restore process. The default is 1. This variable is available only for DB2 Enterprise-Extended Edition.

libname
Specify the name of the shared library that contains the vendor backup and restore I/O functions to be used. The *libname* variable can contain the full path and file name of the library. If the full path is not specified, the path defaults to where the user exit program resides. This variable is optional.

tablespace
Specify one or more table spaces to backup. If no table space is specified, all table spaces are backed up. To specify more than one table space, place the names in double quotes separated by commas. For example: `"tblspc01,tblspc02"`.

Example

```
wecctlib ECC_Backup_Database NONE NONE YES YES 2 NONE 4 1024
wecctlib ECC_Backup_Database joe joepw NO NO 0 /dev/rmt0
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

When this task is run on DB2 Enterprise-Extended Edition, it backs up only the node on which it is executed.

If a database becomes damaged or corrupted, it can be returned to the state of the backed up copy. If a successfully restored database was enabled for roll-forward recovery at the time of the backup, it can be returned to the state that it was in prior to the occurrence of damage. The backup might be directed to hard disk, diskette, tape, ADSM utility, or to other vendor products enabled for DB2.

For more information, see the **BACKUP DATABASE** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Create_Admin_Server

Description

Creates the DB2 Administration Server on a Windows NT host.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

`wecctlib ECC_Create_Admin_Server [user password]`

Arguments

user Specifies the user account. This argument is optional, but it is required if a password is defined.

password
Specifies the password for the user account. This argument is optional, but it is required if a user is defined.

Example

```
wecctlib ECC_CREATE_ADMIN_SERVER joe joepw
```

DB2 version

This task runs only on DB2 UDB on a Windows NT host.

Usage notes

If the *user* and *password* variables are specified, the DB2 Administration Server instance is associated with this user account. The specified user account must be a valid SQL identifier and must exist in the security database.

For more information, see the **db2admin** system command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Create_Explain_Tables

Description

Creates DB2 Explain tables in the specified database.

TMR roles

- Super
- Senior
- Admin
- User

Run program syntax (from a monitor)

```
wecctlib ECC_Create_Explain_Tables
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

This task uses the EXPLAIN.DLL provided by DB2 to create the following explain tables:

- EXPLAIN_ARGUMENT
- EXPLAIN_INSTANCE
- EXPLAIN_OBJECT
- EXPLAIN_OPERATOR
- EXPLAIN_PREDICATE
- EXPLAIN_STATEMENT
- EXPLAIN_STREAM

For more information, see the *DB2 SQL Reference* for the version of DB2 that you are using.

ECC_Create_Sample_Database

Description

Creates a DB2 sample database named SAMPLE.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Create_Sample_Database [path]
```

Arguments

path Specify the path on which to create the SAMPLE database:

NONE Specifies that SAMPLE is created on the default database path.

path Specify a path. On Windows NT, the path is a single drive letter.

Example

- For AIX:

```
wecctlib ECC_Create_Sample_Database /tmp
```

- For Windows NT:

```
wecctlib ECC_Create_Sample_Database E
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

When this task is run on DB2 Enterprise-Extended Edition, it affects all nodes in the node group.

You can run this task only on DB2 server nodes. The SAMPLE database cannot be created on nodes that are database clients only. The database is created with the instance authentication type that is specified by the database manager configuration parameter authentication. The user ID determines the qualifiers for the tables. If the SAMPLE database already exists, this task creates the tables for the user ID issuing the command and grants the appropriate privileges.

The default time-out is 60 seconds. If the task does not complete in this amount of time, the completion status is not displayed. The task will, however, continue to completion. To help ensure that the status displays, you can specify a longer time-out period.

For more information, see the **db2sampl** system command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Deactivate_Database

Description

Stops the database that you specify.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Deactivate_Database [username password]
```

Arguments

username
Specify the name of the user stopping the database. This variable is optional, but it is required if you specify the *password* variable.

password
Specify the password for the user name. This variable is optional, but it is required if you specify the *username* variable.

Example

```
wecctlib ECC_Deactivate_Database joe joepw
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

When this task is run on DB2 Enterprise-Extended Edition, it deactivates the specified database on all nodes within the system.

Databases initialized with an activate database command (such as “ECC_Activate_Database” on page 210) can be shut down only with a deactivate database command (such as “ECC_Deactivate_Database”). If a database was initialized with an activate database command, the last application disconnecting from the database will not shut down the database.

For more information, see the **DEACTIVATE DATABASE** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Drop_Admin_Server

Description

Deletes the DB2 Administration Server instance on a Windows NT host.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Drop_Admin_Server
```

DB2 version

This task runs only on DB2 UDB on a Windows NT host.

For more information, see the **db2admin** system command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Drop_Sample_Database

Description

Deletes the contents and all log files of the SAMPLE database, uncatalogs the database, and deletes the database subdirectory.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Drop_Sample_Database
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

When this task is run on DB2 Enterprise-Extended Edition, it affects all nodes that are listed in the db2nodes.cfg file.

The SAMPLE database must not be in use when this task is run; all users must be disconnected before the database can be dropped.

The SAMPLE database must be cataloged in the system database directory. If the SAMPLE database is the only entry in the local database directory, the local database directory is deleted automatically.

For more information, see the **DROP DATABASE** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Force_All_Applications

Description

Forces local or remote users or applications off a DB2 instance to allow for maintenance on a server.

TMR roles

- Super

- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Force_All_Applications
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

When this task is run on DB2 Enterprise-Extended Edition, it affects all nodes that are listed in the $HOME/sqllib/db2nodes.cfg file.

Forcing a user off a DB2 instance results in the loss of that user's connections to all databases. To preserve database integrity, only users who are idling or running interruptible database operations can be terminated. Users creating a database cannot be forced.

If an operation that cannot be interrupted is forced, the operation must be successfully rerun before the database becomes available. ECC_Stop_DB2 cannot be run during a force. The database manager remains active so that subsequent database manager operations can be handled without the need to run ECC_Start_DB2.

After a **FORCE** command is issued, the database stops accepting requests. Additional force commands might be required to completely force all users off.

For more information, see the **sqlefrce** system command in the *DB2 API Reference* for the version of DB2 that you are using.

ECC_Force_Applications

Description

Forces specific local or remote DB2 applications and users off the system to allow for maintenance on a server.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

`wecctlib ECC_Force_Applications` *`application01`* [*`application02 ... application10`*]

Arguments

application

Specify one to ten agent IDs if you are running DB2 for common servers or one to ten application handles if you are running DB2 UDB or DB2 Enterprise-Extended Edition. For example, you might specify 41408.

You can use the ECC_List_Applications task to find the agent IDs or application handles of currently running applications. You must specify at least one agent ID or application handle.

Example

```
wecctlib ECC_Force_Applications 41408 42623
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

When this task is run on DB2 Enterprise-Extended Edition, it affects all nodes that are listed in the $HOME/sqllib/db2nodes.cfg file.

Forcing a user off a DB2 instance results in the loss of that user's connections to all databases. To preserve database integrity, only users who are idling or running interruptible database operations can be terminated. Users creating a database cannot be forced.

If an operation that cannot be interrupted is forced, the operation must be successfully rerun before the database becomes available. ECC_Stop_DB2 cannot be run during a force. The database manager remains active so that subsequent database manager operations can be handled without the need to run ECC_Start_DB2.

After a **FORCE** command is issued, the database stops accepting requests to connect. Additional force commands might be required to completely force all users off.

For more information, see the **FORCE APPLICATION** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Get_Admin_Configuration

Description

Gets the DB2 administration configuration.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Get_Admin_Configuration
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

The DB2 Administration Server is a special instance that enables remote administration of DB2 servers. For DB2 Enterprise-Extended Edition, this command returns information on all nodes that share the same $HOME/sqllib directory, and it can be issued from any of these nodes.

For more information, see the **GET ADMIN CONFIGURATION** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Get_Admin_Server_Instance

Description

Gets the DB2 Administration Server instance name.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Get_Admin_Server_Instance
```

DB2 version

This task runs on DB2 UDB and DB2 Enterprise-Extended Edition.

For more information, see the **db2admin** system command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Get_Database_Configuration

Description

Returns the values of individual entries in the database configuration file.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Get_Database_Configuration [key01 ... key05 [username
password]]
```

Arguments

key Specify the name of the database configuration parameter that you want to retrieve. You can specify up to five keys. This argument is optional.

username
Specify the name of the user to attach to a remote partition. If you run this task with DB2 Enterprise-Extended Edition on a partition group endpoint and at least one of the partitions in the partition group is on a remote host, you must specify a user name. The user name is used for all nodes included in the partition group whether they are on a local or remote host.

password
Specify the password used to authenticate with the *username* variable.

Example

```
wecctlib ECC_Get_Database_Configuraiton APPLHEAPSZ
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

When this command is run on DB2 Enterprise-Extended Edition, it returns information only for the node on which it is executed.

If no keys are specified, all database configuration parameter values are returned. If one or more keys are specified, only the values for the specified parameters are returned.

For more information, see the **GET DATABASE CONFIGURATION** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using and the **sqlfxdb** command in the *DB2 API Reference* for the version of DB2 that you are using.

ECC_Get_Database_Manager_Configuration

Description

Returns the values of individual entries in the database manager configuration file.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Get_Database_Manager_Configuration [key01 ... key05]
```

Arguments

key Specify the name of the database manager configuration parameter that you want to retrieve. You can specify up to five keys. This argument is optional.

Example

```
wecctlib ECC_Get_Database_Manager_Configuration KEEPDARI MAXDARI
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

If no keys are specified, all database manager configuration parameter values are returned. If one or more keys are specified, only the values for the specified parameters are returned.

For more information, see the **GET DATABASE MANAGER CONFIGURATION** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using and the **sqlfxsys** command in the *DB2 API Reference* for the version of DB2 that you are using.

ECC_Invoke_Stored_Procedure

Description

Invokes a specified procedure that you created using DB2, which is stored at the location of the database.

TMR roles

- Super
- Senior
- Admin
- User

Run program syntax (from a monitor)

```
wecctlib ECC_Invoke_Stored_Procedure name ["data"]
```

Arguments

name Specify the name of the procedure.

data Specify any data to pass to the server routine. Quotes are required if you specify more than one piece of data. This variable is optional.

Example

```
wecctlib ECC_Invoke_Stored_Procedure proc1 "123 xray"
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

The server procedure runs at the location of the database and returns data to the client application.

The program runs in two parts, one on the client and the other on the server. The server procedure at the database runs within the same transaction as the client application. If the client application and the server procedure are on the same node, the server procedure is run locally.

For more information, see the **INVOKE STORED PROCEDURE** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_List_Applications

Description

Lists application program name, authorization ID (user name), agent ID, application ID, and database name.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

`wecctlib ECC_List_Applications [`*`detail username password`*`]`

Arguments

detail Specify **YES** to show details. The default is **NO**. This variable is optional.

username
Specify the name of the user to attach to a remote partition. If you run this task with DB2 Enterprise-Extended Edition on a partition group endpoint and at least one of the partitions in the partition group is on a remote host, you must specify a user name. The user name is used for all nodes included in the partition group whether they are on a local or remote host.

password
Specify the password used to authenticate with the *username* variable.

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

If this task runs on a DB2 instance endpoint, all applications are listed. If this task is run on a DB2 database endpoint, only the applications for the database are listed.

When this task is run on DB2 Enterprise-Extended Edition, this task lists only those applications on the selected node.

For more information, see the **LIST APPLICATIONS** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_List_Backup_Recovery_File

Description

Lists a DB2 backup and recovery file.

TMR roles

- Super
- Senior
- Admin
- User

Run program syntax (from a monitor)

```
wecctlib ECC_List_Backup_Recovery_File [type [timestamp [schema name
username [password]]]]
```

Arguments

type Specify the type of list request. This variable is optional.

ALL Lists all entries in the recovery history file. This parameter is the default.

SINCE
Lists entries based on the timestamp specified in the *timestamp* variable.

CONTAINING
Lists the table or table space specified in the *name* variable.

timestamp
Specify a complete timestamp (*yyyymmddhhnnss*) or an initial prefix timestamp where the minimum is yyyy. If you specify the **ALL** or **CONTAINING** parameters in the *type* variable, this variable is ignored. This variable is optional.

schema
Specify a schema name to use as a qualifier for a table name. Specify **NONE** to uniquely identify a table space with the *name* variable. If you specify the **ALL** or **SINCE** parameters in the *type* variable, this variable is ignored. This variable is optional.

name If you specified a schema name, specify the table name to use. The table is identified by 'schema.name'. If you specified **NONE** for the *schema* variable, specify the unique identifier for a table space. If you specify the **ALL** or **SINCE** parameters in the *type* variable, this variable is ignored. For more information, see the *DB2 SQL Reference* for the version of DB2 that you are using.

username
Specify the name of the user to attach to a remote partition. If you run this task with DB2 Enterprise-Extended Edition on a partition group endpoint and at least one of the partitions in the partition group is on a remote host, you must specify a user name. The user name is used for all nodes included in the partition group whether they are on a local or remote host.

password
Specify the password used to authenticate with the *username* variable.

Example

```
wecctlib ECC_List_Backup_Recovery_File SINCE 1996
```

DB2 version
This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

For more information, see the **LIST HISTORY** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_List_DCS_Applications

Description
Lists the contents of the Database Connection Services (DCS) directory.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_List_DCS_Applications show_detail [username password]
```

Argument

show_detail
Specify **YES** or **NO** to indicate whether the output should include detailed information.

If you specify **NO**, basic information is returned. The basic information includes:

- Host authorization ID (username)
- Application program name
- Agent ID
- Outbound application ID (luwid)

If you specify **YES**, basic and detailed information is returned. The detailed information includes:

- Application ID
- Application sequence number
- Client database alias
- Client product ID
- Code page ID
- Outbound sequence number
- Host database name
- Host product ID

username
Specify the name of the user to attach to a remote partition. If you run this task with DB2 Enterprise-Extended Edition on a partition group endpoint and at least one of the partitions in the partition

group is on a remote host, you must specify a user name. The user name is used for all nodes included in the partition group whether they are on a local or remote host.

password
Specify the password used to authenticate with the *username* variable.

Example

```
wecctlib ECC_List_DCS_Applications YES
```

DB2 version
This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes
You can use this task to match client application connections to the gateway with corresponding host connections from the gateway. You can also use agent ID information to force specified applications off a DDCS server.

For more information, see the **LIST DCS APPLICATIONS** command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_List_Node_Directory

Description
Lists the contents of the node directory.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_List_Node_Directory [admin [detail]]
```

Arguments

admin Specify **YES** to select Administration Server nodes. The default is **NO**. This argument is optional.

detail Specify **YES** to show details. The default is **NO**. This argument is optional.

Example

```
wecctlib ECC_List_Node_Directory YES
```

DB2 version
This task runs on DB2 UDB and DB2 Enterprise-Extended Edition.

Usage notes
A node directory is created and maintained on each database client and contains an entry for each remote workstation that has a database that the client can access. This task identifies a range of entries in the recovery history files that are deleted. All entries with timestamps equal to or less than the timestamp provided are deleted. Entries can be pruned even if some entries from the most recent restore set are deleted from the file.

For more information, see the **LIST NODE DIRECTORY** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Prune_Recovery_History_File

Description

Deletes entries from the recovery history file.

TMR roles

- Super
- Senior
- Admin

Run program syntax (from a monitor)

`wecctlib ECC_Prune_Recovery_History_File` *`timestamp force`* [*`username password`*]

Arguments

timestamp
Specify a complete timestamp (of the form *yyyymmddhhmmss*) or an initial prefix (minimum form is *yyyy*). All entries in the recovery history file that have timestamps equal to or less than *timestamp* are deleted from the file.

An example of a complete timestamp is: 19970614102030

An example of an initial prefix is: 199712

force
Specify **YES** or **NO** to indicate whether entries from the most recent store set should be deleted from the file if they meet the timestamp criteria.

username
Specify the name of the user to attach to a remote partition. If you run this task with DB2 Enterprise-Extended Edition on a partition group endpoint and at least one of the partitions in the partition group is on a remote host, you must specify a user name. The user name is used for all nodes included in the partition group whether they are on a local or remote host.

password
Specify the password used to authenticate with the *username* variable.

Example

```
wecctlib ECC_Prune_Recovery_History_File 199712 YES
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

This task identifies a range of entries in the recovery history file that are deleted. All entries with timestamps equal to or less than the timestamp provided are deleted. Entries can be pruned even if some entries from the most recent restore set are deleted from the file.

For more information, see the **PRUNE HISTORY** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Quiesce_Tablespaces

Description

Quiesces table spaces for a DB2 table.

TMR roles

- Super
- Senior
- Admin

Run program syntax (from a monitor)

`wecctlib ECC_Quiesce_Tablespaces` *`schema name mode`* [*`username password`*]

Arguments

schema
Specify the qualifier for the table name. For example, if the qualified name of a table is USERID.TABLE1, the *schema* is USERID.

If you enter `_default_`, the current user ID is used.

name Specify the unqualified table name. For example, if the qualified name of a table is USERID.TABLE1, the *name* is TABLE1.

mode Specify the quiesce mode in which you want to put the table spaces for the table:

SHARE
Specifies that the table spaces are put in shared mode. In this mode, all users (yourself included) can read but not change the table data.

INTENT
Specifies that the table spaces are put in update mode. In this mode, only you can update the table data. Other users can read, but not update the data.

EXCLUSIVE
Specifies that the table spaces are put in exclusive mode. In this mode, only you can read or update the table data.

RESET
Specifies that the table spaces are put in normal mode.

username
Specify the name of the user to attach to a remote partition. If you run this task with DB2 Enterprise-Extended Edition on a partition group endpoint and at least one of the partitions in the partition group is on a remote host, you must specify a user name. The user name is used for all nodes included in the partition group whether they are on a local or remote host.

password
Specify the password used to authenticate with the *username* variable.

Example

```
wecctlib ECC_Quiesce_Tablespaces joe table1 SHARE
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

When this task runs on DB2 Enterprise-Extended Edition in a single node environment, it quiesces all table spaces in a load operation in exclusive mode for the duration of the load. In an MPP environment, this task acts locally on a node and quiesces only the portion of table spaces that belong to the node where the load is performed.

If the table being quiesced does not have a specified owner, the db2ecc user ID is used as the default.

There are 3 valid quiesce modes: share, intent to update, and exclusive. There are 3 possible states resulting from this task: QUIESCED SHARE, QUIESCED UPDATE, and QUIESCED EXCLUSIVE.

When the quiesce share request is received, the transaction requests intent share locks for the table spaces and a share lock for the table. When the transaction obtains the locks, the state of the table spaces is changed to QUIESCED SHARE. The state is granted to the quiescer only if there is no conflicting state held by other users. The state of the table spaces is recorded in the table space table, along with the authorization ID and the database agent ID of the quiescer, so that the state is persistent.

The table cannot be changed while the table spaces for the table are in QUIESCED SHARE state. Other share mode requests to the table and table spaces will be allowed. When the transaction commits or rolls back, the locks are released, but the table spaces for the table remain in QUIESCED SHARE state until the state is explicitly reset.

When the quiesce exclusive request is made, the transaction requests super exclusive locks on the table spaces, and a super exclusive lock on the table. When the transaction obtains the locks, the state of the table spaces changes to QUIESCED EXCLUSIVE. The state of the table spaces, along with the authorization ID and the database agent ID of the quiescer, are recorded in the table space table. Because the table spaces are held in super exclusive mode, no other access to the table spaces is allowed. The user who invokes the quiesce function (the quiescer), however, has exclusive access to the table and the table spaces.

When a quiesce update request is made, the table spaces are locked in intent exclusive (IX) mode, and the table is locked in update (U) mode. The state of the table spaces with the quiescer is recorded in the table space table.

For more information, see the **QUIESCE TABLESPACES FOR TABLE** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Rebind_All_Packages

Description

Rebinds all DB2 packages in a database.

TMR roles

- Super
- Senior

- Admin

Run program syntax (from a monitor)

```
wecctlib ECC_Rebind_All_Packages log_file [username password]
```

Arguments

log_file
Specify the path (optional) and the file name (mandatory) to be used for recording errors that result from the package revalidation procedure.

username
Specify the name of the user who is rebinding the database. This variable is optional, but it required if you specify *password*.

password
Specify the password for the user name. This variable is optional, but it is required if you specify *username*.

Example

```
wecctlib ECC_Rebind_All_Packages /tmp/pkg.log
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

When this task is run on DB2 Enterprise-Extended Edition, it affects all nodes in the node group.

This task uses the CLP **REBIND** command to attempt the revalidation of all packages in a database. You can allow package revalidation to occur implicitly when the packages are first used.

For more information, see the **db2rbind** system command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Rebind_Package

Description

Lets users re-create a package stored in the database without needing a bind file.

TMR roles

- Super
- Senior
- Admin

Run program syntax (from a monitor)

```
wecctlib ECC_Rebind_Package schema name
```

Arguments

schema
Specify the qualifier for the package name. For example, if the qualified name of a package is USERID.PACK1, the *schema* is USERID.

If you specify `_default_`, the current user ID is used.

name Specify the unqualified package name. For example, if the qualified name of a package is USERID.PACK1, the *name* is PACK1.

Example

```
wecctlib ECC_Rebind_Package joe pkg1
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

The **REBIND** command does not automatically commit the transaction following a successful rebind. The user must explicitly commit the transaction. This enables ″what if″ analysis, in which the user updates certain statistics, and then tries to rebind the package to see what changes. It also permits multiple rebinds within a unit of work.

The REBIND command commits the transaction if auto-commit is enabled.

For more information, see the **REBIND** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Reorganize_Table

Description

Reorganizes a table by reconstructing the rows to eliminate fragmented data, and by compacting information.

TMR roles

- Super
- Senior
- Admin

Run program syntax (from a monitor)

`wecctlib ECC_Reorganize_Table` *`schema name`* `[`*`i_schema i_name`* `[`*`ts_name`*`]]`

Arguments

schema
Specify the qualifier for the table name. For example, if the qualified name of a table is USERID.TABLE1, the *schema* is USERID.

name Specify the unqualified table name. For example, if the qualified name of a table is USERID.TABLE1, the *name* is TABLE1.

The following arguments are optional:

i_schema
Specify the qualifier for the index name.

i_name
Specify the unqualified index name.

ts_name
Specify the unqualified table space name.

Example

```
wecctlib ECC_Reorganize_Table joe table1
wecctlib ECC_Reorganize_Table joe table1 NONE NONE tblspc1
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

When this task is run on DB2 Enterprise-Extended Edition, it affects all nodes in the node group.

Tables that have been modified so many times that data is fragmented and access performance is noticeably slow are candidates for reorganization. Use ECC_Reorgchk to determine if a table needs reorganizing. Be sure to complete all database operations and release all locks before running ECC_Reorganize_Table. After reorganizing a table, use ECC_Run_Statistics to update the table statistics, and ECC_Rebind_Packages to rebind the packages that use this table.

If the reorganization is not successful, do not delete temporary files. The database manager uses these files to recover the database.

For more information, see the **REORGANIZE TABLE** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Reorgchk

Description

Calculates statistics on the database to determine if tables need to be reorganized.

TMR roles

- Super
- Senior
- Admin

Run program syntax (from a monitor)

`wecctlib ECC_Reorgchk` *`stats on_table`* [*`schema name`*]

Arguments

stats Specify the type of table statistics to use:

UPDATE
Specifies that the DB2 RUNSTATS utility is used to update table statistics. These updated statistics are then used to determine if reorganization is needed.

CURRENT
Specifies that the current table statistics are used to determine if table reorganization is needed.

on_table
Specify the tables to check:

USER Specifies that all tables owned by the current authorization ID are to be checked.

SYSTEM
Specifies that all system tables are to be checked.

ALL Specifies that all user-owned and system-owned tables are to be checked.

TABLE
Specifies that a particular table (specified with *schema* and *name*) is to be checked.

schema
Specify the qualifier for the table name. For example, if the qualified name of a table is USERID.TABLE1, the *schema* is USERID.

name Specify the unqualified table name. For example, if the qualified name of a table is USERID.TABLE1, the *name* is TABLE1.

Example

```
wecctlib ECC_Reorgchk UPDATE SYSTEM
wecctlib ECC_Reorgchk CURRENT TABLE joe table1
```

DB2 version
This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes
When this task is run on DB2 Enterprise-Extended Edition, it affects all nodes in the node group.

This task calculates statistics obtained from six different formulas to determine if performance has deteriorated or can be improved by reorganizing a table.

For more information, see the **REORGCHK** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Reset_Admin_Configuration

Description
Resets the parameters in the database manager configuration file (that are relevant to the DB2 Administration Server) to the system defaults.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Reset_Admin_Configuration
```

DB2 version
This task runs on DB2 UDB and DB2 Enterprise-Extended Edition.

Usage notes
The DB2 Administration Server is a special instance that enables remote administration of DB2 servers. This command resets the database manager configuration file, $HOME/sqllib/db2systm. This command affects all nodes that are listed in the $HOME/sqllib/db2nodes.cfg file.

For more information, see the **RESET ADMIN CONFIGURATION** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Reset_Database_Configuration

Description
Resets the configuration of a specific database to the system defaults.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Reset_Database_Configuration [username password]
```

Arguments

username
Specify the name of the user to attach to a remote partition. If you run this task with DB2 Enterprise-Extended Edition on a partition group endpoint and at least one of the partitions in the partition group is on a remote host, you must specify a user name. The user name is used for all nodes included in the partition group whether they are on a local or remote host.

password
Specify the password used to authenticate with the *username* variable.

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

When this task is run on DB2 Enterprise-Extended Edition, it affects only the node on which it is run.

Changes to the database configuration file become effective only after they are loaded into memory. All applications must disconnect from the database before the changes can occur.

If an error occurs, the database configuration file does not change. The database configuration file cannot be reset if the checksum is invalid. The checksum might be invalid if the database configuration file is changed without using the appropriate command. If this problem occurs, the database must be restored to reset the database configuration file.

For more information, see the **RESET DATABASE CONFIGURATION** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Reset_Database_Manager_Configuration

Description

Resets the parameters in the database manager configuration file to the system defaults.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Reset_Database_Manager_Configuration
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

Changes to the database manager configuration file become effective only after they are loaded into memory. For a server configuration parameter, the changes occur when ECC_Start_DB2 runs. For a client configuration parameter, changes occur when the application is restarted.

If an error occurs, the database manager configuration file does not change. The database manager configuration file cannot be reset if the checksum is invalid. The checksum might be invalid if the database manager configuration file is changed without using the appropriate command. If this problem occurs, the database manager must be installed again to reset the database manager configuration file.

For more information, see the **RESET DATABASE MANAGER CONFIGURATION** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Restart_Database

Description

Restarts a database that has been abnormally terminated and left in an inconsistent state.

TMR roles

- Super
- Senior
- Admin
- User

Run program syntax (from a monitor)

```
wecctlib ECC_Restart_Database [username password]
```

Arguments

username

Specify the user who is restarting the database. If you run this task with DB2 Enterprise-Extended Edition on a partition group endpoint and at least one of the partitions in the partition group is on a remote host, you must specify a user name. The user name is used for all nodes included in the partition group whether they are on a local or remote host.

password

Specify the password for the user name. This parameter is required if you specify the *username* parameter.

Example

```
wecctlib ECC_Restart_Database joe joepw
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

When this task is run on DB2 Enterprise-Extended Edition, it affects only the node on which it is executed.

Run this task if an attempt to connect to a database returns an error message indicating that the database must be restarted. An error occurs only if the previous session with this database terminated abnormally (for example, due to power failure).

For more information, see the **RESTART DATABASE** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Run_Statistics

Description

Updates statistics about the physical characteristics of a table and the associated indexes.

TMR roles

- Super
- Senior
- Admin

Run program syntax (from a monitor)

`wecctlib ECC_Run_Statistics` *`schema name with_dist indexes index_only detailed change`* [*`index-schema index-name username password`*]

Arguments

schema
Specify the qualifier for the table name. For example, if the qualified name of a table is USERID.TABLE1, the *schema* is USERID.

name Specify the unqualified table name. For example, if the qualified name of a table is USERID.TABLE1, the *name* is TABLE1.

with_dist
Specify **YES** or **NO** to indicate whether you want distribution statistics calculated.

indexes

YES Updates statistics on the indexes.

NO Indicates that you do not want to update statistics on the indexes.

INDEX
Specify to update statistics on a specific index that is defined in the *index-schema* and *index-name* variables.

index_only
Specify **YES** or **NO** to indicate whether you want to update statistics on the indexes only.

If *indexes* is YES and *index_only* is NO, statistics on both the table and its indexes are updated.

If *indexes* is NO and *index_only* is NO, statistics on the table only are updated.

detailed
Specify **YES** or **NO** to indicate whether you want extended index statistics calculated when you are updating statistics on the indexes.

change
Specify **YES** or **NO** to indicate how other users can access the table while statistics are calculated:

YES Other users can read from and write to the table while statistics are calculated.

NO Other users can only read from the table while statistics are calculated.

index-schema
Specify the qualifier for the index name. This variable is required if you specify the **INDEX** parameter for the *indexes* variable.

index-name
Specify the unqualified index name. This variable is required if you specify the **INDEX** parameter for the *indexes* variable.

username
Specify the name of the user to attach to a remote partition. If you run this task with DB2 Enterprise-Extended Edition on a partition group endpoint and at least one of the partitions in the partition group is on a remote host, you must specify a user name. The user name is used for all nodes included in the partition group whether they are on a local or remote host.

password
Specify the password used to authenticate with the *username* variable.

Example

```
wecctlib ECC_Run_Statistics joe table1 NO INDEX NO YES NO joe index2
```

DB2 version
This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes
When this task is run on DB2 Enterprise-Extended Edition, it affects all nodes in the node group.

Run this task when a table has had many updates, or after reorganizing a table. The statistics updated by this task include number of records, number of pages, and average record length. The optimizer uses these statistics when determining access paths to the data.

After statistics are updated, you can create new access paths to the table by rebinding the packages.

Collect statistics that apply only to the tables before you create indexes. By running the table statistics first, you ensure that the indexes statistics are not overlaid by table statistics.

For more information, see the **RUN STATISTICS** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Set_Admin_Server_ID

Description
Establishes or modifies the user account associated with the DB2 Administration Server instance on a Windows NT host.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Set_Admin_Server_ID user password
```

Arguments

user Specify the user account to be used.

password
Specify the password to be used.

Example

```
wecctlib ECC_Set_Admin_Server_ID joe joepw
```

DB2 version

This task runs on DB2 UDB and DB2 Enterprise-Extended Edition.

For more information, see the **db2admin** system command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Start_Admin_Server

Description

Starts the DB2 Administration Server on a Windows NT host.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Start_Admin_Server
```

DB2 version

This task runs on DB2 UDB and DB2 Enterprise-Extended Edition.

For more information, see the **db2admin** system command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Start_DB2

Description

Starts the DB2 instance and allocates resources.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Start_DB2 [profile [nodenum [startopt [hostname port
[netname [temp]]]]]]
```

Arguments

The following arguments can be run only on DB2 Enterprise-Extended Edition.

profile Specify the name of the profile file (which defines the DB2 environment) to execute at each node or specify **NONE**. This variable is optional.

nodenum
Specify the number of the node that you want to start. Valid node numbers are from 0 to 999. If you do not specify the node number, a normal startup is performed on the node where the start command was issued.

startopt
Specify one of the following start options. This variable is optional.

ADDNODE
If you specify this parameter, the following listed variables are specified according to the following descriptions:

hostname
Specify the host name to add to the db2nodes.cfg file. This variable is required.

port Specify the logical port that you want to add to the db2nodes.cfg file. This variable is required. Valid logical port numbers are from 0 to 999.

netname
Specify the network name to be added to the db2nodes.cfg file or specify **NONE**. If you do not specify this parameter, or if it is set to **NONE**, the network name is set to the value specified in *hostname*.

temp Specify the node number to specify that the containers for the temporary table spaces will be the same as the containers on the specified node number for each database in the instance. Specify **WITHOUT** to specify that containers for the temporary table spaces are not created for any of the databases. This parameter is optional.

STANDALONE
If you specify this parameter, all further arguments are ignored.

RESTART
If you specify this parameter, the following listed variables are specified according to the following descriptions:

hostname
Specify the host name that will override the host name in the node configuration file. This variable is optional.

port Specify the logical port that will override the port in the node configuration file. This variable is optional.

netname
Specify the network name that will override the name in the node configuration file. This variable is optional.

Example

```
wecctlib ECC_Start_DB2 joeprof1
wecctlib ECC_Start_DB2 NONE 6 STANDALONE
```

DB2 version
This task runs on DB2 for common servers and DB2 UDB.

Usage notes
When this task is run on DB2 Enterprise-Extended Edition in a multinode environment, it affects all nodes that are listed in the $HOME/sqllib/db2notes.cfg file, unless you specify the *nodenum* variable.

After this task starts, the database manager instance runs until you stop it, even if all programs that were using it have ended.

For more information when you are running UNIX, see the **db2start** system command in the *DB2 Command Reference* for the version of DB2 that you are using. For more information when you are running DB2 on Windows NT, see the:

- *IBM DATABASE 2 for Windows 95 & Windows NT Installation and Operation Guide* if you are using DB2 for common servers
- *Quick Beginnings for Windows NT* if you are using DB2 UDB or DB2 Enterprise-Extended Edition

ECC_Start_DB2_NT_Security

Description
Starts the DB2 security service on a Windows NT system.

TMR roles
- Super
- Senior

Run program syntax (from a monitor)
```
wecctlib ECC_Start_DB2_NT_Security
```

DB2 version
This task runs on DB2 for common servers and DB2 Enterprise-Extended Edition.

For more information, see the:

- *IBM DATABASE 2 for Windows 95 & Windows NT Installation and Operation Guide* if you are using DB2 for common servers
- *Quick Beginnings for Windows NT* if you are using DB2 UDB or DB2 Enterprise-Extended Edition

ECC_Stop_Admin_Server

Description
Stops the DB2 Administration Server on a Windows NT host.

TMR roles
- Super
- Senior

Run program syntax (from a monitor)
```
wecctlib ECC_Stop_DB2_Admin_Server
```

DB2 version
This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

For more information, see the **db2admin** system command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Stop_DB2

Description

Stops the DB2 instance and releases resources.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Stop_DB2 [profile [type [nodenum]]]
```

Arguments

The following arguments can be run only on DB2 Enterprise-Extended Edition.

profile Specify the name of the profile that was executed during startup, which defined the DB2 environment for the nodes that were started. This variable is optional.

type Specify one of the following stop options:

NODE If you specify this parameter, the following listed variable is specified according to the following description:

nodenum
Specify the number of the node to be stopped. Valid node numbers are from 0 to 999. This variable is required.

DROP If you specify this parameter, the following listed variable is specified according to the following description:

nodenum
Specify the number of the node to be dropped. Valid node numbers are from 0 to 999. When this argument is specified, all nodes are stopped. This variable is required.

FORCE
If you specify this parameter, all further arguments are ignored. This parameter specifies that ″ALL″ is used when stopping the database manager at each node.

Example

```
wecctlib ECC_Stop_DB2 joeprof1
wecctlib ECC_Stop_DB2 NONE FORCE
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

When this task is run on DB2 Enterprise-Extended Edition in a multinode environment, it affects all nodes that are listed in the $HOME/sqllib/db2notes.cfg file, unless you specified the **NODE** parameter for the *type* variable.

This task does not stop DB2 if any applications are connected to databases. If there are no database connections, but there are instance attachments, ECC_Stop_DB2 forces the instance attachments and stops DB2.

ECC_Stop_DB2 can be run only on a DB2 server. After DB2 stops, a successful completion message is sent to the standard output device. If an error occurs, processing stops, and an error message is sent to the standard output device.

For more information when you are running UNIX, see the **db2stop** system command in the *DB2 Command Reference* for the version of DB2 that you are using. For more information when you are running on Windows NT, see the:

- *IBM DATABASE 2 for Windows 95 & Windows NT Installation and Operation Guide* if you are using DB2 for common servers
- *Quick Beginnings for Windows NT* if you are using DB2 UDB or DB2 Enterprise-Extended Edition

ECC_Stop_DB2_NT_Security

Description

Stops the DB2 security service on a Windows NT system.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Stop_DB2_NT_Security
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

For more information, see the:

- *IBM DATABASE 2 for Windows 95 & Windows NT Installation and Operation Guide* if you are using DB2 for common servers
- *Quick Beginnings for Windows NT* if you are using DB2 UDB or DB2 Enterprise-Extended Edition

ECC_Update_Admin_Configuration

Description

Modifies individual entries in the database manager configuration file that are relevant to the DB2 Administration Server.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Update_Admin_Configuration key01 val01 [key02 val02 ...
key05 val05]
```

Arguments

key Select or specify the name of the database manager configuration parameter that you want to update. You can select or specify from one to five keyword-value pairs.

val Specify the new value for the configuration parameter specified in the corresponding *key*.

Example

```
wecctlib ECC_Update_Admin_Configuraiton DIAGLEVEL 3
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

For more information, see the **UPDATE ADMIN CONFIGURATION** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Update_Database_Configuration

Description

Modifies individual entries in a specific database configuration file.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

`wecctlib ECC_Update_Database_Configuration` *`key01 val01`* [*`key02 val02 ... key05 val05`* [*`username password`*]]

Arguments

key Specify the name of the database configuration parameter you want to update. You can specify from one to five keyword-value pairs.

val Specify the new value for the configuration parameter specified in the corresponding *key*.

username
Specify the name of the user to attach to a remote partition. If you run this task with DB2 Enterprise-Extended Edition on a partition group endpoint and at least one of the partitions in the partition group is on a remote host, you must specify a user name. The user name is used for all nodes included in the partition group whether they are on a local or remote host.

password
Specify the password used to authenticate with the *username* variable.

Example

```
wecctlib ECC_Update_Database_Configuration LOCKLIST 120
```

To change the values of the APPLHEAPSZ, DBHEAP, PCKCACHESZ, and CATALOGCACHE_SZ configuration parameters, you can run:

```
wecctlib ECC_Update_Database_Configuration APPLHEAPSZ 256 DBHEAP 1500
PCKCACHESZ 128 CATALOGCACHE_SZ 256
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

When this task is run on DB2 Enterprise-Extended Edition, it affects only the node on which it is executed.

Changes to the database configuration file become effective only after they are loaded into memory. All applications must disconnect from the database before changes can be loaded. If an error occurs, the database configuration file does not change.

The database configuration file cannot be updated if the checksum is invalid. If this problem occurs, the database must be restored to reset the database configuration file.

For more information, see the **UPDATE DATABASE CONFIGURATION** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

ECC_Update_Database_Manager_Configuration

Description

Modifies individual entries in the database manager configuration file.

TMR roles

- Super
- Senior

Run program syntax (from a monitor)

```
wecctlib ECC_Update_Database_Manager_Configuration key01 val01
[key02 val02 ... key05 val05]
```

Arguments

key Specify the name of the database manager configuration parameter you want to update. You can specify from one to five keyword-value pairs.

val Specify the new value for the configuration parameter specified in the corresponding *key*.

Example

The following example changes the values of the MAXAGENTS and MAXCAGENTS configuration parameters to update the parameters to the specified values.

```
wecctlib ECC_Update_Database_Manager_Configuration MAXAGENTS 300 MAXCAGENTS 250
```

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

Changes to the database manager configuration file become effective only after they are loaded into memory. For a server configuration, parameter changes are loaded into memory when ECC_Start_DB2 is run. For a client configuration parameter, this occurs when the application is restarted. If an error occurs, the database manager configuration file does not change.

The database manager configuration file cannot be updated if the checksum is invalid. If this error occurs, the database manager must be reinstalled to reset the database manager configuration file.

For more information, see the **UPDATE DATABASE MANAGER CONFIGURATION** CLP command in the *DB2 Command Reference* for the version of DB2 that you are using.

Data Replication Tasks

Library name:
DB2_ECC-ReplicationTasks

This section provides information about the function and uses of the tasks in the DB2_ECC-ReplicationTasks, which are used with DB2 Replication. The tasks are listed alphabetically.

If you are using DB2 for common servers, you must have DataPropagator Relational installed to run the data replication tasks.

The wecctlib command, which is the program that runs the DB2 ECC monitors, consists of two files:

wecctlib
Contains all tasks except for the Data Replication tasks for Windows NT hosts.

wecctlib2
Contains all Data Replication tasks for Windows NT hosts.

The Run program syntax (from a monitor) section of each task denotes when you should use the wecctlib command and when you should use the wecctlib2 command. For example, if you want to run ECC_Apply_Start on AIX, you might enter the following command:

```
wecctlib ECC_Apply_Start joe joepw "D:\applydir" NONE NONE NO R 1
```

If you want to run ECC_Apply_Start on Windows NT, you might enter the following command:

```
wecctlib2 ECC_Apply_Start joe joepw "D:\applydir" NONE NONE NO R 1
```

For more information on data replication tasks, see the:

- *DataPropagator Relational Guide* if you are using DB2 for common servers
- *Replication Guide and Reference* if you are using DB2 UDB or DB2 Enterprise-Extended Edition

Table 52 on page 245 lists the tasks in this library and the endpoints they run on.

Table 52. Data Replication Task Library and Task Endpoints

Task	Description	Task Endpoints*				
		DB	DB Part.	DB Part. Grp.	Inst.	Man. Node
ECC_Apply_Start	Start the data replication Apply program	✓				
ECC_Apply_Stop	Stop the data replication Apply program					✓
ECC_Capture_Get_Log_Seq	Retrieve the current log sequence number and timestamp	✓	✓	✓	✓	
ECC_Capture_Prune	Prune the data replication Capture tables	✓				
ECC_Capture_Reinit	Reinitialize the data replication Capture program	✓				
ECC_Capture_Resume	Resume the data replication Capture program	✓				
ECC_Capture_Start	Start the data replication Capture program	✓				
ECC_Capture_Stop	Stop the data replication Capture program	✓				
ECC_Capture_Suspend	Suspend the data replication Capture program	✓				

*Task Endpoints DB= database, DB Part. = DB2 partition, DB Part. Grp. = DB2 partition group, Inst. = instance (partitioned or nonpartitioned), Man. Node = managed node

ECC_Apply_Start

Description

A program that refreshes or updates a target table.

TMR roles

- Super
- Senior
- Admin

Run program syntax (from a monitor) on AIX and Solaris

```
wecctlib ECC_Apply_Start userid password "path" qualifier server
asnload priority drep_vers drep_inst
```

Run program syntax (from a monitor) for Windows NT

```
wecctlib2 ECC_Apply_Start userid password "path" qualifier server
asnload priority drep_vers drep_inst
```

Arguments

userid Specify the user ID to be used for starting the Apply program.

password

Specify the password for *username*:

NONE For AIX only: Specifies that no password is supplied.

password
For NT only: Specify the password used to authenticate the user ID entered for *userid*.

path Specify the full path to which you want the Apply program to write files. You must specify the quotation marks around the name.

An example of a full path on AIX is:

```
"/u/myname/applydir"
```

On NT:

```
"D:\applydir"
```

qualifier

If this task is being run on DB2 Enterprise-Extended Edition, specify the Apply qualifier that the Apply instance uses to identify the subscriptions to be served.

If this task is being run on DB2 for common servers, specify **NONE**.

server

If this task is being run on DB2 Enterprise-Extended Edition, specify the name of the server where the replication control tables will reside, or specify **NONE**.

If this task is being run on DB2 for common servers, specify **NONE**.

asnload
Specify **YES** or **NO** to indicate whether to call the ASNLOAD user-exit routine. ASNLOAD calls an IBM or vendor utility to initialize a point-in-time table.

priority For DB2 for common servers only: Specify the processing order of subscriptions:

R Specifies round-robin processing of subscriptions by LASTRUN, PRIORITY for this instance of the Apply program.

L Specifies processing of subscriptions by PRIORITY, LASTRUN for this instance of the Apply program.

OTHER
Specifies that the Apply program selects the next subscription in the order of PRIORITY, COPY_OWNER, and COPY_TABLE.

drep_vers
Specify the version of DPROPR or IBM Replication to use with the task. If you are running the task on Windows NT, be sure that the directory for the path variable matches the data replication version you select:

0 Specify to use the default. If this task is run on a DB2 UDB database endpoint, the IBM Replication executable is the default. If this task is run on a DB2 for common servers database endpoint, the DPROPR V1 executable is the default.

1 Specify to run this task using the DPROPR V1 executable.

5 Specify to run this task using the IBM Replication

executable. If you specify 5 but are running this task on a DB2 for common servers endpoint on UNIX, you must also specify the *drep_inst* variable.

drep_inst
UNIX only: Specify this variable only if you specified 5 for the *drep_vers* variable and this task will run on a DB2 for common servers endpoint. Specify the name of an instance running DB2 Enterprise-Extended Edition on the same system as the DB2 for common servers database endpoint. This variable is required so that the IBM Replication executable can be used.

Example

```
wecctlib ECC_Apply_Start joe joepw "D:\applydir" NONE NONE NO R 5 v5inst
wecctlib ECC_Apply_Start joe joepw "D:\applydir" NONE NONE NO R 1
```

The first example forces the task to use the IBM Replication executable from a DB2 for common servers database endpoint. The second example forces the task to run the DPROPR V1 executable from a DB2 UDB database endpoint.

DB2 version
This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes
After you run ECC_Apply_Start, the Apply program runs continuously until:

- You stop the Apply program (see “ECC_Apply_Stop”).
- An unexpected error or failure occurs.
- The ENABLE field of the global control record in any refresh control table indicates that it should stop.

The Apply program returns a message only if it fails within 60 seconds.

You must start the Apply program with an Apply ID. ECC_Apply_Start uses the root ID, then prompts you for an Apply ID. On Windows NT, you’re then prompted for the password.

ECC_Apply_Stop

Description
Stops the Apply program on AIX.

TMR roles

- Super
- Senior
- Admin

Run program syntax (from a monitor) on AIX

```
wecctlib ECC_Apply_Stop userid "path" [qualifier] drep_vers
drep_inst
```

Argument

userid Specify the user ID to be used for stopping the Apply program.

path Specify the full path to which you want the Apply program to write files. You must specify the quotation marks around the name.

An example of a full path on AIX is:

```
"/u/myname/applydir"
```

On NT:

```
"D:\applydir"
```

qualifier
If this task is being run on , specify the Apply qualifier that is used to identify the subscriptions that are served.

drep_vers
Specify the version of DPROPR or IBM Replication to use with the task. If you are running the task on Windows NT, be sure that the directory for the path variable matches the data replication version you select:

0 Specify to use the default. If this task is run on a DB2 UDB database endpoint, the IBM Replication executable is the default. If this task is run on a DB2 for common servers database endpoint, the DPROPR V1 executable is the default.

1 Specify to run this task using the DPROPR V1 executable.

5 Specify to run this task using the IBM Replication executable. If you specify 5 but are running this task on a DB2 for common servers endpoint on UNIX, you must also specify the *drep_inst* variable.

drep_inst
UNIX only: Specify this variable only if you specified 5 for the *drep_vers* variable and this task will run on a DB2 for common servers endpoint. Specify the name of an instance running DB2 Enterprise-Extended Edition on the same system as the DB2 for common servers database endpoint. This variable is required so that the IBM Replication executable can be used.

Example

```
wecctlib ECC_Apply_Stop joe "D:\applydir" 5 v5inst
wecctlib ECC_Apply_Stop joe "D:\applydir" 1
```

The first example forces the task to use the IBM Replication executable from a DB2 for common servers database endpoint. The second example forces the task to run the DPROPR V1 executable from a DB2 UDB database endpoint.

DB2 version
This task runs on DB2 UDB, DB2 Enterprise-Extended Edition, and it runs on DB2 for common servers with limited function.

Usage notes
When this task is run with DB2 Enterprise-Extended Edition, it can run on the AIX and Windows NT platforms. When this task is run with DB2 Enterprise-Extended Edition, it affects all nodes in the node group.

When this task is run with DB2 for common servers, it runs only on AIX.

ECC_Capture_Get_Log_Seq

Description

Retrieves the current log sequence number and timestamp.

TMR roles

- Super
- Senior
- Admin

Run program syntax (from a monitor) on AIX and Solaris

`wecctlib ECC_Capture_Get_Log_Seq "path" drep_inst`

Run program syntax (from a monitor) on Windows NT

`wecctlib2 ECC_Capture_Get_Log_Seq "path" drep_inst`

Arguments

path Specify the full path to which you want the Apply program to write files. You must specify the quotation marks around the name.

An example of a full path on AIX is:

```
"/u/myname/applydir"
```

On NT:

```
"D:\applydir"
```

drep_inst

UNIX only: If you run this task on a DB2 for common servers database endpoint, specify the name of an instance running DB2 Enterprise-Extended Edition on the same system as the DB2 for common servers database endpoint. This variable is required so that the IBM Replication executable can be used.

Example

```
wecctlib ECC_Capture_Get_Log_Seq "D:\applydir" v5inst1
```

DB2 version

This task runs on DB2 UDB and DB2 Enterprise-Extended Edition.

Usage notes

The information retrieved by this task helps determine how far the Capture program has read the DB2 log.

ECC_Capture_Prune

Description

Prunes the change data table and unit-of-work table if you specified the **NOPRUNE** argument when you started the Capture program.

TMR roles

- Super
- Senior
- Admin

Run program syntax (from a monitor) on AIX and Solaris

`wecctlib ECC_Capture_Prune "path" drep_vers drep_inst`

Run program syntax (from a monitor) on Windows NT

```
wecctlib2 ECC_Capture_Prune "path" drep_vers drep_inst
```

Arguments

path Specify the full path to which you want the Apply program to write files. You must specify the quotation marks around the name.

An example of a full path on AIX is:

```
"/u/myname/applydir"
```

On NT:

```
"D:\applydir"
```

drep_vers
Specify the version of DPROPR or IBM Replication to use with the task. If you are running the task on Windows NT, be sure that the directory for the path variable matches the data replication version you select:

0 Specify to use the default. If this task is run on a DB2 UDB database endpoint, the IBM Replication executable is the default. If this task is run on a DB2 for common servers database endpoint, the DPROPR V1 executable is the default.

1 Specify to run this task using the DPROPR V1 executable.

5 Specify to run this task using the IBM Replication executable. If you specify 5 but are running this task on a DB2 for common servers endpoint on UNIX, you must also specify the *drep_inst* variable.

drep_inst
UNIX only: Specify this variable only if you specified 5 for the *drep_vers* variable and this task will run on a DB2 for common servers endpoint. Specify the name of an instance running DB2 Enterprise-Extended Edition on the same system as the DB2 for common servers database endpoint. This variable is required so that the IBM Replication executable can be used.

Example

```
wecctlib ECC_Capture_Prune "D:\applydir" 5 v5inst1
wecctlib ECC_Capture_Prune "D:\applydir" 1
```

The first example forces the task to use the IBM Replication executable from a DB2 for common servers database endpoint. The second example forces the task to run the DPROPR V1 executable from a DB2 UDB database endpoint.

DB2 version
This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes
If pruning was disabled by using the **NOPRUNE** argument while starting the Capture program, this command prunes the change data and unit-of-work tables one time.

ECC_Capture_Reinit

Description

Causes the Capture program to reread tables while it is running so that it recognizes new registrations (for DPROPR) and new source table definitions (for DB2 UDB and DB2 Enterprise-Extended Edition)

TMR roles

- Super
- Senior
- Admin

Run program syntax (from a monitor) on AIX and Solaris

```
wecctlib ECC_Capture_Reinit "path" drep_vers drep_inst
```

Run program syntax (from a monitor) on Windows NT

```
wecctlib2 ECC_Capture_Reinit "path" drep_vers drep_inst
```

Arguments

path
Specify the full path to which you want the Apply program to write files. You must specify the quotation marks around the name.

An example of a full path on AIX is:

```
"/u/myname/applydir"
```

On NT:

```
"D:\applydir"
```

drep_vers
Specify the version of DPROPR or IBM Replication to use with the task. If you are running the task on Windows NT, be sure that the directory for the path variable matches the data replication version you select:

0 Specify to use the default. If this task is run on a DB2 UDB database endpoint, the IBM Replication executable is the default. If this task is run on a DB2 for common servers database endpoint, the DPROPR V1 executable is the default.

1 Specify to run this task using the DPROPR V1 executable.

5 Specify to run this task using the IBM Replication executable. If you specify 5 but are running this task on a DB2 for common servers endpoint on UNIX, you must also specify the *drep_inst* variable.

drep_inst
UNIX only: Specify this variable only if you specified 5 for the *drep_vers* variable and this task will run on a DB2 for common servers endpoint. Specify the name of an instance running DB2 Enterprise-Extended Edition on the same system as the DB2 for common servers database endpoint. This variable is required so that the IBM Replication executable can be used.

Example

```
wecctlib ECC_Capture_Reinit "D:\applydir" 5 v5inst1
wecctlib ECC_Capture_Reinit "D:\applydir" 1
```

The first example forces the task to use the IBM Replication executable from a DB2 for common servers database endpoint. The second example forces the task to run the DPROPR V1 executable from a DB2 UDB database endpoint.

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

This command also rereads the tuning parameters table for any changes made to the tuning parameters.

Attention: For DPROPR, do not use ECC_Capture_Reinit to reinitialize the Capture program after canceling a registration or dropping a registered table while Capture is running. For DB2 UDB and DB2 Enterprise-Extended Edition, do not use ECC_Capture_Reinit to reinitialize the Capture program after canceling a replication source or dropping a replication source table while Capture is running. In either situation, use the ECC_Capture_Stop task to stop the Capture program and then use the ECC_Capture_Start task with the **WARM** parameter to restart the Capture program.

ECC_Capture_Resume

Description

Restarts a suspended Capture program.

TMR roles

- Super
- Senior
- Admin

Run program syntax (from a monitor) on AIX and Solaris

```
wecctlib ECC_Capture_Resume "path" drep_vers drep_inst
```

Run program syntax (from a monitor) on Windows NT

```
wecctlib2 ECC_Capture_Resume "path" drep_vers drep_inst
```

Arguments

path

Specify the full path to which you want the Apply program to write files. You must specify the quotation marks around the name.

An example of a full path on AIX is:

```
"/u/myname/applydir"
```

On NT:

```
"D:\applydir"
```

drep_vers

Specify the version of DPROPR or IBM Replication to use with the task. If you are running the task on Windows NT, be sure that the directory for the path variable matches the data replication version you select:

0 Specify to use the default. If this task is run on a DB2 UDB database endpoint, the IBM Replication executable is the default. If this task is run on a DB2 for common servers database endpoint, the DPROPR V1 executable is the default.

1 Specify to run this task using the DPROPR V1 executable.

5 Specify to run this task using the IBM Replication executable. If you specify 5 but are running this task on a DB2 for common servers endpoint on UNIX, you must also specify the *drep_inst* variable.

drep_inst
UNIX only: Specify this variable only if you specified 5 for the *drep_vers* variable and this task will run on a DB2 for common servers endpoint. Specify the name of an instance running DB2 Enterprise-Extended Edition on the same system as the DB2 for common servers database endpoint. This variable is required so that the IBM Replication executable can be used.

Example

```
wecctlib ECC_Capture_Resume "D:\applydir" 5 v5inst1
wecctlib ECC_Capture_Resume "D:\applydir" 1
```

The first example forces the task to use the IBM Replication executable from a DB2 for common servers database endpoint. The second example forces the task to run the DPROPR V1 executable from a DB2 UDB database endpoint.

DB2 version
This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes
If you are pruning a file with ECC_Capture_Prune and you run ECC_Capture_Stop or ECC_Capture_Suspend, pruning does not resume after you run ECC_Capture_Resume. You must run ECC_Capture_Prune again.

ECC_Capture_Start

Description
Starts the Capture program, which reads database log or journal records to capture data about changes made to source tables.

TMR roles

- Super
- Senior
- Admin

Run program syntax (from a monitor) on AIX and Solaris

```
wecctlib ECC_Capture_Start "path" type prune drep_vers drep_inst
```

The **TRACE** argument is not supported. This task runs with **NOTRACE** as the default.

Run program syntax (from a monitor) on Windows NT

```
wecctlib2 ECC_Capture_Start "path" type prune drep_vers drep_inst
```

The **TRACE** argument is not supported. This task runs with **NOTRACE** as the default.

Arguments

path Specify the full path to which you want the Capture program to write files. You must specify the quotation marks around the name.

An example of a full path on AIX is:

```
"/u/myname/captdir"
```

On NT:

```
"D:\captdir"
```

type Specify where the Capture program should start processing:

WARM
Specifies that the Capture program starts processing where it ended in its previous run if warm start information is available. Otherwise, the Capture program switches to a cold start.

WARMNS
Specifies that the Capture program starts processing where it ended in its previous run if warm start information is available. Otherwise, the Capture program issues a message and ends.

COLD Specifies that the Capture program starts processing by deleting all rows in its change data table, unit-of-work table, pruning control table, and trace table during initialization.

prune Specify **YES** or **NO** to indicate whether the Capture program should automatically prune the change data and unit-of-work tables.

drep_vers
Specify the version of DPROPR or IBM Replication to use with the task. If you are running the task on Windows NT, be sure that the directory for the path variable matches the data replication version you select:

0 Specify to use the default. If this task is run on a DB2 UDB database endpoint, the IBM Replication executable is the default. If this task is run on a DB2 for common servers database endpoint, the DPROPR V1 executable is the default.

1 Specify to run this task using the DPROPR V1 executable.

5 Specify to run this task using the IBM Replication executable. If you specify 5 but are running this task on a DB2 for common servers endpoint on UNIX, you must also specify the *drep_inst* variable.

drep_inst
UNIX only: Specify this variable only if you specified 5 for the *drep_vers* variable and this task will run on a DB2 for common servers endpoint. Specify the name of an instance running DB2 Enterprise-Extended Edition on the same system as the DB2 for common servers database endpoint. This variable is required so that the IBM Replication executable can be used.

Example

```
wecctlib ECC_Capture_Start "/w/joe/captdir" WARM NO 5 v5inst1
wecctlib ECC_Capture_Start "/w/joe/captdir" WARM NO 1
```

The first example forces the task to use the IBM Replication executable from a DB2 for common servers database endpoint. The second example forces the task to run the DPROPR V1 executable from a DB2 UDB database endpoint.

DB2 version

This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes

The Capture program runs continuously until you stop it or it detects an error.

ECC_Capture_Stop

Description

Stops the data replication Capture program in an orderly way and commits the log records that it processed up to that point.

TMR roles

- Super
- Senior
- Admin

Run program syntax (from a monitor) on AIX and Solaris

`wecctlib ECC_Capture_Stop "`*path*`" `*drep_vers drep_inst*

Run program syntax (from a monitor) on Windows NT

`wecctlib2 ECC_Capture_Stop "`*path*`" `*drep_vers drep_inst*

Arguments

path

Specify the full path to which you want the Apply program to write files. You must specify the quotation marks around the name.

An example of a full path on AIX is:

```
"/u/myname/applydir"
```

On NT:

```
"D:\applydir"
```

drep_vers

Specify the version of DPROPR or IBM Replication to use with the task. If you are running the task on Windows NT, be sure that the directory for the path variable matches the data replication version you select:

0 Specify to use the default. If this task is run on a DB2 UDB database endpoint, the IBM Replication executable is the default. If this task is run on a DB2 for common servers database endpoint, the DPROPR V1 executable is the default.

1 Specify to run this task using the DPROPR V1 executable.

5 Specify to run this task using the IBM Replication executable. If you specify 5 but are running this task on a DB2 for common servers endpoint on UNIX, you must also specify the *drep_inst* variable.

drep_inst
UNIX only: Specify this variable only if you specified 5 for the *drep_vers* variable and this task will run on a DB2 for common servers endpoint. Specify the name of an instance running DB2 Enterprise-Extended Edition on the same system as the DB2 for common servers database endpoint. This variable is required so that the IBM Replication executable can be used.

Example

```
wecctlib ECC_Capture_Stop "D:\applydir" 5 v5inst1
wecctlib ECC_Capture_Stop "D:\applydir" 1
```

The first example forces the task to use the IBM Replication executable from a DB2 for common servers database endpoint. The second example forces the task to run the DPROPR V1 executable from a DB2 UDB database endpoint.

DB2 version
This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes
If you are pruning a file with ECC_Capture_Prune and you run ECC_Capture_Stop or ECC_Capture_Suspend, pruning does not resume after you run ECC_Capture_Resume. You must run ECC_Capture_Prune again.

ECC_Capture_Suspend

Description
Relinquishes resources to operational transactions during peak periods without damaging the Capture environment.

TMR roles

- Super
- Senior
- Admin

Run program syntax (from a monitor) on AIX and Solaris

```
wecctlib ECC_Capture_Suspend "path" drep_vers drep_inst
```

Run program syntax (from a monitor) on Windows NT

```
wecctlib2 ECC_Capture_Suspend "path" drep_vers drep_inst
```

Arguments

path
Specify the full path to which you want the Apply program to write files. You must specify the quotation marks around the name.

An example of a full path on AIX is:

```
"/u/myname/applydir"
```

On NT:

```
"D:\applydir"
```

drep_vers
Specify the version of DPROPR or IBM Replication to use with the

task. If you are running the task on Windows NT, be sure that the directory for the path variable matches the data replication version you select:

0 Specify to use the default. If this task is run on a DB2 UDB database endpoint, the IBM Replication executable is the default. If this task is run on a DB2 for common servers database endpoint, the DPROPR V1 executable is the default.

1 Specify to run this task using the DPROPR V1 executable.

5 Specify to run this task using the IBM Replication executable. If you specify 5 but are running this task on a DB2 for common servers endpoint on UNIX, you must also specify the *drep_inst* variable.

drep_inst
UNIX only: Specify this variable only if you specified 5 for the *drep_vers* variable and this task will run on a DB2 for common servers endpoint. Specify the name of an instance running DB2 Enterprise-Extended Edition on the same system as the DB2 for common servers database endpoint. This variable is required so that the IBM Replication executable can be used.

Example

```
wecctlib ECC_Capture_Suspend "D:\applydir" 5 v5inst1
wecctlib ECC_Capture_Suspend "D:\applydir" 1
```

The first example forces the task to use the IBM Replication executable from a DB2 for common servers database endpoint. The second example forces the task to run the DPROPR V1 executable from a DB2 UDB database endpoint.

DB2 version
This task runs on DB2 for common servers, DB2 UDB, and DB2 Enterprise-Extended Edition.

Usage notes
This task suspends the Capture program until you run ECC_Capture_Resume.

If you are pruning a file with ECC_Capture_Prune and you run ECC_Capture_Stop or ECC_Capture_Suspend, pruning does not resume after you run ECC_Capture_Resume. You must run ECC_Capture_Prune again.

Appendix. DB2 Snapshot Application ID Formats

The DB2 snapshot application ID is an identifier that is generated when the application connects to the database at the database manager. The application ID is unique across the network.

The format used for the application ID depends on the communication protocol between the client and the server machine on which the database manager is running. Each of the formats consists of three parts separated by periods. The formats for each of the following communication protocols are:

APPC

Format
Network.LU_Name.Application_inst

Example
`CAIBMTOR.OSFDBX0.930131194520`

Details
The application ID generated for an application using the APPC protocol is made up by concatenating the network name, the logical unit (LU) name, and the logical unit-of-work ID (LUWID) instance number. In Systems Network Architecture (SNA), a logical unit is a port through which an end user access the SNA network to communicate with another end user.

Network and *LU_name* can each be a maximum of 8 characters. *Application_inst* corresponds to the 12–decimal-character LUWID instance number.

IPX/SPX

Format
Netid.nodeid.Identifier

Example
`C11A8E5C.400011528250.1116183045`

Details
The application ID generated for an application using the IPX/SPX protocol is made up by concatenating a character network ID (8 hexadecimal characters), a node ID (12 hexadecimal characters), and a unique identifier for this particular application (*Identifier*). *Identifier* corresponds to a 10-decimal-character timestamp of the form *mmddhhmmss*.

Local applications

Format
*LOCAL.*DB2_instance.Identifer*

Example
`*LOCAL.DBINST1.930131214645`

Details
The application ID generated for a local application is made up by concatenating the string *LOCAL, the name of the DB2 instance, and a unique identifier for this particular application.

NetBIOS

Format

*NETBIOS.*nname.Identifier*

Example

`*NETBIOS.DBERE.930131214645`

Details

The application ID generated for an application using the NetBIOS protocol is made up by concatenating the string *NETBIOS, the NetBIOS node name (nname) defined in the client's database configuration file, and a unique identifier for this particular application.

TCP/IP

Format

*TCPIP.*IPAddr.Identifier*

Example

`*TCPIP.A12CF9E8.930131214645`

Details

The application ID generated for an application using the TCP/IP protocol is made up by concatenating the string *TCPIP, the IP address in hexadecimal characters (maximum of 8), and a unique identifier for this particular application.

You can use "ECC_List_Applications" on page 222 to list the application IDs for all applications currently active for a particular database.

Terms and Abbreviations

A

action. (1) An operation on a managed object, the semantics of which are defined as part of the managed object class definition. (2) In the UNIX operating system, a defined task that an application performs. An action modifies the properties of an object or manipulates the object in some way.

administrator. See *TME 10 administrator*.

agent. (1) In DB2, a separate process or thread that carries out all DB2 requests that are made by a particular client application. Each agent operates with its own private memory and shares the database manager and database global resources such as the buffer pool with other agents. (2) In DB2 ECC, an entity that represents one or more managed objects by emitting notifications regarding the objects and handling requests from managers for management operations to modify or query the objects. See also *daemon*.

alert. A message sent to a management services focal point in a network to identify a problem or an impending problem.

application ID. In DB2, a string that uniquely identifies an application across networks. An ID is generated at the time that the application connects to the database. This ID is known on both the client and the server and can be used to correlate the two parts of the application.

Apply program. A replication program that is used to refresh or update a target table, depending on the applicable source-to-target rules. Contrast with *Capture program*.

authorization role. In the TME 10 environment, a role assigned to TME 10 administrators to enable them to perform their assigned systems management tasks. A role can be granted over the entire Tivoli Management Region (TMR) or over a specific set of resources, such as those contained in a policy region. Examples of authorization roles include: super, senior, admin, and user. See also *resource role* and *TMR role*.

B

BAROC. See *Basic Recorder of Objects in C*.

Basic Recorder of Objects in C (BAROC). In the event server of the TME 10 Enterprise Console, the internal representation of the defined event classes.

C

Capture program. A replication program that reads database log or journal records to capture data about changes made to source tables. Contrast with *Apply program*.

CLI. Command line interface.

D

daemon. A program that runs unattended to perform a standard service. Some daemons are triggered automatically to perform their tasks; others operate periodically. See also *agent (2)*.

database partition. In DB2, a part of a database that consists of its own user data, indexes, configuration files, and transaction logs. Typically, a single database partition exists on each physical node and the processors on each system are used by the database manager at each database partition to manage its part of the database's total data. Because data is divided across database partitions, the power of multiple processors on multiple physical nodes is used to satisfy requests for information.

DB2 database. In DB2 ECC, a managed resource that represents a particular database in a DB2 instance.

DB2 ECC administrator. A TME 10 administrator who has been granted the privileges to perform DB2 database management tasks.

DB2 instance. (1) In DB2, a logical DB2 database manager environment similar to an image of the actual database manager environment. You can have several instances of the DB2 database manager product on the same workstation. (2) In DB2 ECC, a managed object that represents a particular DB2 instance, or a partitioned database server.

DB2 node. In DB2 ECC, a node that is listed in the db2nodes.cfg node configuration file. This file contains configuration information for all database partitions in an instance, and is shared by all database partitions for that instance. The db2nodes.cfg file is located in the sqllib subdirectory of the home directory for the instance.

DB2 partition. In DB2 ECC, a managed resource that represents the combination of two items: a DB2 node, and a DB2 database that exists in the partitioned database server that includes that DB2 node. For example, if two logical nodes (0 and 1) exist on one system, and the partitioned database server that includes these two logical nodes has a database named

SAMPLE, you can create two DB2 partitions: one for SAMPLE at node 0, and one for SAMPLE at node 1.

DB2 partition group. In DB2 ECC, a managed resource that represents a named collection of DB2 partitions. The partitions must belong to the same DB2 instance.

distribute. In the TME 10 environment, to send a profile to a subscriber or endpoint. The profiles that are sent are copies of the original profile; the original profile remains in the profile manager.

E

endpoint. (1) In the TME 10 environment, a TME 10 client that is the ultimate recipient for any type of TME 10 operation. Examples are managed nodes and, in DB2 ECC, DB2 instances, DB2 databases, DB2 partitions, and DB2 partition groups. See also *proxy endpoint*. (2) In the TME 10 environment, a TME 10 service that runs on multiple operating systems and performs TME 10 operations on those systems, thereby enabling the TME 10 Framework to manage the systems as TME 10 clients.

event. (1) An occurrence of significance to a task; for example, an SNMP trap or the completion of an asynchronous operation. (2) In the TME 10 environment, any significant change in the state of a system resource, network resource, or network application. An event can be generated for a problem, for the resolution of a problem, or for the successful completion of a task. Examples of events are: the normal starting and stopping of a process, the abnormal termination of a process, and the malfunctioning of a server. (3) In TME 10 Enterprise Console (TEC), a message sent to the TME 10 Enterprise Console.

F

fast communication manager (FCM). A group of functions that provide internodal communication support.

FCM. See *fast communication manager*.

I

indicator. In Sentry, an icon on the TME 10 desktop that graphically displays the status of a monitor that has been associated with it. The icon resembles a thermometer, which the TME 10 administrator can read to determine the status of the monitor.

indicator collection. In the TME 10 environment, a single location from which a TME 10 administrator can determine the status of monitors in different profiles, as well as clear and reset alarmed states.

indoubt transaction. A global transaction whose state is in doubt when the transaction is prepared, but not yet committed or rolled back; or when one phase of it completes successfully but the system fails before the second phase can complete, leaving the database in an inconsistent state.

J

job. In the TME 10 environment, a resource consisting of a task and its preconfigured execution parameters. The execution parameters can specify the set of managed resources on which the job is to execute.

M

managed node. In the TME 10 environment, any managed system (for example, a UNIX host or Windows NT system) on which the TME 10 Framework is installed. Each managed node is provided with the capability of interpreting and processing profiles that are distributed to it.

managed resource. In the TME 10 environment, any hardware or software entity (machine, service, system, or facility) that is represented by a database object in the TME 10 object database and an icon on the TME 10 desktop. The managed resources in a particular policy region must be of a resource type that is supported in the region, and they are subject to the rules that govern the region. Managed resources include, but are not limited to, profiles, profile managers, managed nodes, and, in DB2 ECC, DB2 instances, DB2 databases, DB2 partitions, and DB2 partition groups.

monitor. Software that observes, supervises, controls, or verifies operations of a system. Monitors typically monitor information such as available disk space or application errors and compare the information to defined thresholds. When thresholds are exceeded, either system or network administrators can be notified, or an automated response can be performed.

monitoring collection. In Sentry, a collection of monitoring sources or predefined monitors. Several monitoring collections are packaged with Sentry, but TME 10 administrators can use custom-developed and third-party monitoring collections as well, such as DB2 ECC monitoring collections.

monitoring source. In DB2 ECC, a monitor that has been predefined to observe and record a particular activity. For example, the “DB2 instance status” monitoring source determines whether the DB2 instance is up or down. DB2 ECC monitoring sources are contained in the various DB2 ECC monitoring collections.

N

notice. In the TME 10 environment, a message generated by a systems management operation that

contains information about an event or the status of an application. Notices are stored in *notice groups*.

notice group. In the TME 10 environment, an application- or operation-specific container that stores and displays notices pertaining to specific TME 10 functions.

O

oserv. The name of the object request broker used by the TME 10 environment. This object request broker runs on the TMR server and each TMR client. In object-oriented programming, an object request broker is software that serves as an intermediary by transparently enabling objects to exchange requests and responses.

P

partitioned database. A database with two or more database partitions. Data in user tables can be located in one or more database partitions. When a table is on multiple partitions, some of its rows are stored in one partition and others are stored in other partitions. See *database partition*. Contrast with *single-partition database*.

partitioned database server. In DB2 ECC, a DB2 instance that has one or more partitioned databases.

policy. In the TME 10 environment, a set of rules that are applied to managed resources. A specific rule in a policy is referred to as a policy method.

policy region. In the TME 10 environment, a group of managed resources that share one or more common policies. TME 10 administrators use policy regions to model the management and organizational structure of a network computing environment. The administrators can group similar resources, define access to and control the resources, and associate rules for governing the resources. The policy region contains resource types and the list of resources to be managed. A policy region is represented on the TME 10 desktop by an icon that resembles a capitol building (dome icon). When a Tivoli Management Region (TMR) is created, a policy region for the host where Tivoli is installed is also created. In most cases, a TME 10 administrator creates other policy regions and subregions to represent the organization of resources. A TMR addresses the physical connectivity of resources, whereas a policy region addresses the logical organization of resources.

policy subregion. In the TME 10 environment, a policy region created or residing in another policy region.

profile. In the TME 10 environment, a container for application-specific information about a particular type of resource. A TME 10 application specifies the template for its profiles; the template includes information about the resources that can be managed by that TME 10 application. For example, Sentry specifies the templates for its profiles, which contain definitions of monitors. A profile is created in the context of a profile manager; the profile manager links a profile to the TME 10 resource (for example, a managed node) that uses the information contained in the profile. A profile does not have any direct subscribers.

profile manager. In the TME 10 environment, a container for profiles that links the profiles to a set of resources, called *subscribers.* Subscribers can include managed nodes, other profile managers, and, in DB2 ECC, DB2 instances, DB2 databases, DB2 partitions, and DB2 partition groups. A profile manager can contain profiles of multiple types or multiple profiles of the same type. TME 10 administrators use profile managers to organize and distribute profiles. A profile manager is created in the context of a policy region and is a managed resource in a policy region.

proxy endpoint. A representation of an entity (such as a network device or a host) that functions as a subscriber for Sentry profiles. A TME 10 administrator associates each proxy endpoint with a managed node; several proxy endpoints can be associated with a single managed node.

R

registration. In DPROPR, the act of identifying a source table to DPROPR to make the table available for subscription. In DB2 UDB and DB2 Enterprise-Extended Edition, the term source table is used instead.

resource role. In DB2 ECC, an authorization role that allows an administrator to perform an operation, or set of operations, against a specific managed resource.

response. A response action.

response action. In Sentry, a defined operation or task that is performed when the state of a monitor reaches a specified threshold. Sending a Tivoli notice to a TME 10 administrator is a response action.

response level. In Sentry, the state of a monitor when a specified threshold has been reached. An administrator can set thresholds for each response level and have Sentry trigger a different response action for each level. There can also be several response actions for each response level. The five response levels are: critical, severe, warning, normal, and always.

role. See *authorization role*.

root. In the UNIX operating system, the user name for the system user with the most authority.

S

scheduler. A computer program designed to perform functions such as scheduling, initiating, and terminating jobs.

Sentry. A TME 10 product that monitors system resources, initiates any necessary corrective actions, and informs system administrators of potential problems. Sentry consists of a group of monitors that are installed on each managed node that is to be monitored. It resolves some events on its own and can send others to the TME 10 Enterprise Console. See also TME 10 Distributed Monitoring.

Simple Network Management Protocol (SNMP). In the Internet suite of protocols, a network management protocol that is used to monitor routers and attached networks. SNMP is an application layer protocol. Information on devices managed is defined and stored in the application's Management Information Base (MIB).

single-partition database. A database that has only one database partition.

SNMP. Simple Network Management Protocol.

source table. In DB2 UDB and DB2 Enterprise-Extended Edition, a table that contains the data that is to be copied to a target table.

subscribe. In the TME 10 environment, to become eligible to receive profiles that reside in a profile manager.

subscriber. In the TME 10 environment, a TME 10 client, a profile manager, and any endpoint type (for example, a PC managed node or a proxy endpoint) that is subscribed to a profile manager. In DB2 ECC, a DB2 instance, DB2 database, DB2 partition, and DB2 partition group can be subscribers. Although profiles are distributed to a subscriber, the subscriber may or may not be the final destination of the profile distribution. See also *endpoint*.

subscription. In the TME 10 environment, the process of identifying the endpoints or profile managers to which profiles will be distributed. Endpoints, profile managers, and profiles are associated with each other in profile managers.

subscription list. In the TME 10 environment, a list that identifies the subscribers to a profile manager. Including a profile manager on a subscription list (in effect, a list within a list) is a way of subscribing several resources simultaneously rather than adding each one individually.

T

task. In the TME 10 environment, the definition of an action that must be routinely performed on various managed nodes throughout the network. A task defines the executables to be run when the task is executed, the authorization role required to execute the task, and the user and group name under which the task will execute. DB2 ECC comes with custom-developed tasks which can be run against DB2 instances, DB2 databases, DB2 partitions, DB2 partition groups, and managed nodes.

task library. In the TME 10 environment, a container in which a TME 10 administrator can create and store tasks and jobs. DB2 ECC comes with custom-developed task libraries.

TEC. See *TME 10 Enterprise Console*.

threshold. A value that defines a limit for a monitored condition.

Tivoli Management Region (TMR). In the TME 10 environment, a network of connected nodes consisting of a TME 10 server and its clients. A TME 10 server is the server that holds or references the complete set of TME 10 software. An organization can have more than one TMR. A TMR addresses the physical layout of resources, whereas a policy region addresses the logical organization of resources.

TME 10. The suite of Tivoli applications that enable system administrators to manage their network computing enterprise.

TME 10 administrator. In the TME 10 environment, a system administrator who has been authorized to perform systems management tasks and manage policy regions in one or more networks. Each TME 10 administrator is represented by an icon on the TME 10 desktop.

TME 10 client. A client of a TME 10 server.

TME 10 desktop. In the TME 10 environment, the graphical user interface that system administrators use to manage their network computing environment.

TME 10 Distributed Monitoring. In this book, the term Sentry is used to refer to TME 10 Distributed Monitoring. See *Sentry*.

TME 10 Enterprise Console (TEC). A TME 10 product that collects, processes, and automatically initiates corrective actions for system, application, network, and database events; it is the central control point for events from all supported sources. The TME 10 Enterprise Console provides a centralized, global view of the network computing environment; it uses distributed event monitors to collect information, distributed event consoles to present the information to

system administrators, and a central event server to correlate and interpret the information.

TME 10 environment. The TME 10 applications, based on the TME 10 Framework, that are installed at a specific customer location and that address network computing management issues across many platforms. In a TME 10 environment, a system administrator can distribute software, manage user configurations, change access privileges, automate operations, monitor resources, and schedule jobs.

TME 10 Framework. The base software that is required to run the applications in the TME 10 product suite. This software infrastructure enables the integration of systems management applications from Tivoli and the Tivoli Partners. The Framework includes:

- Object request broker (oserv)
- Distributed object database
- Platform services and functions, such as notice groups and the graphical user interface

In a TME 10 environment, the TME 10 Framework is installed on every client and every server with these exceptions:

- The TME 10 Framework is never installed on a client PC; rather, the PC agent is installed on the PC.
- In a particular TMR, the TME 10 server is the only server that contains the full object database.

TME 10 server. A server that holds or references the complete set of TME 10 software, including the full object database.

TMR. See *Tivoli Management Region (TMR)*.

TMR client. In a TME 10 environment, any computer—except the TMR server—on which the TME 10 Framework is installed. See also *TMR server*.

TMR role. In DB2 ECC, an authorization role that allows an administrator to perform an operation, or set of operations, that involves an entire TMR, or that can be run against any managed node within a TMR.

TMR server. The server that holds or references the complete set of TME 10 software, including the full object database, for a specific Tivoli Management Region (TMR).

V

victim. A page or buffer that has been selected for an action to be performed on it. For example, a page that will be written to disk.

Bibliography

This bibliography includes all publications cited in this book and publications that provide additional information for DB2 ECC users.

The DB2 Enterprise Control Center for TME 10 V1.2 Library

The following books comprise the DB2 Enterprise Control Center for TME 10 V1.2 library:

- *DB2 Enterprise Control Center for TME 10 User's Guide*, SC26-9084

  This book describes how to install and use the DB2 ECC product.
- *DB2 Enterprise Control Center for TME 10 Monitoring Collection and Task Library Reference*, SC26-9085

  This book contains information on the monitoring collections and task libraries provided by DB2 ECC.

Related IBM Publications

This section lists books associated with IBM products.

DB2 for common servers Publications

- *DB2 Administration Guide*, S20H-4580
- *DB2 API Reference for common servers*, S20H-4984
- *DB2 Command Reference*, S20H-4645
- *DB2 Database System Monitor Guide and Reference*, S20H-4871
- *IBM DATABASE 2 for Windows 95 & Windows NT Installation and Operation Guide*, S33H-0312
- *DB2 SQL Reference*, S20H-4665

DB2 Universal Database Publications

- *DB2 Universal Database Administration Getting Started*, S10J-8154
- *DB2 Universal Database Administration Guide*, S10J-8157
- *DB2 Universal Database API Reference*, S10J-8167
- *DB2 Universal Database Command Reference*, S10J-8166
- *DB2 Universal Database System Monitor Guide and Reference*, S10J-8164
- *DB2 Universal Database for Windows NT Quick Beginnings*, S10J-8149
- *DB2 Universal Database SQL Reference*, S10J-8165

DDCS and DB2 Connect Publications

- *Distributed Database Connection Services User's Guide for common servers*, S20H-4793
- *DB2 Connect User's Guide*, S10J-8163

DRDA Publications

- *Distributed Relational Database Architecture Connectivity Guide*, SC26-4783

IBM Replication Publications

- *DataPropagator Relational Guide*, SC26-3399
- *Replication Guide and Reference*, S95H-0999

ITSO (Redbook) Publications

- *TME 10 Cookbook for AIX Systems Management and Networking*, SG24-4867
- *Understanding Tivoli's TME 3.0 and TME 10*, SG24-4948
- *Wow! DRDA Supports TCP/IP: DB2 Server for OS/390 and DB2 Universal Database*, SG24-2212

Tivoli Publications

This section lists books produced by Tivoli.

Tivoli/Sentry and Distributed Monitoring Publications

- *Tivoli/Sentry User's Guide*, GC31-8382
- *TME 10 Distributed Monitoring User's Guide*, GC31-8382

TME 10 Versions 3.1 and 3.2 Publications

- *TME 10 Framework Planning and Installation Guide*, SC31-8432
- *TME 10 Framework Reference Manual*, SC31-8434
- *TME 10 Framework User's Guide*, GC31-8433

Index

A

adding monitoring sources
  description 2
  with default values 3
  with the CLI 3
  with the GUI 2
applications, formats for IDs 259
arguments, for tasks 202

C

conn: # of direct reads (4 KB pages) 52
conn: # of direct writes (4 KB pages) 52
conn: # of times agents are stolen 5
conn: Application section inserts 177
conn: Application section lookups 178
conn: Application sorts 169
conn: Application work load 6
conn: Average # of sectors read per direct read 53
conn: Average # of sectors written per direct write 54
conn: Binds/precompiles attempted 179
conn: Buffer pool time waited for prefetch (ms) 54
conn: Commits 179
conn: Data page writes (4 KB pages) 54
conn: DDL SQL statements 180
conn: Deadlocks found during last interval 144
conn: Dynamic SQL statements 180
conn: Explicit commits 181
conn: Explicit rollbacks 182
conn: Failed SQL statements 182
conn: Index page reads (4 KB pages) 55
conn: Index page writes (4 KB pages) 55
conn: Lock escalations during last interval 145
conn: Lock wait time (s) 146
conn: Lock waits 146
conn: Locks held 147
conn: Logical data reads (4 KB pages) 56
conn: Maximum associated agents 7
conn: Open block cursors 183
conn: Open cursors 183
conn: Percent buffer pool hit ratio, data+index 57
conn: Percent buffer pool hit ratio, index 57
conn: Percent DDL SQL 184
conn: Percent locklist space used by application 17
conn: Percent package cache hit ratio 58
conn: Percent UID SQL 184
conn: Pool read time (s) 59
conn: Pool write time (s) 59
conn: Rollbacks 185
conn: Rows deleted 186
conn: Rows inserted 186
conn: Rows selected 187
conn: Rows updated 187
conn: Select SQL statements 188
conn: Sort time (s) 170
conn: SQL requests since last commit 188
conn: Statement sorts 170
conn: Static SQL statements 189
conn: Total pool I/O time (s) 60
conn: Total system CPU time used by agents 7
conn: Total user CPU time used by agents 8
conn: UOW lock wait time (s) 147
conn: Update/Insert/Delete SQL 189

D

Data replication Apply status 139
data replication task endpoints 244
data replication tasks
  ECC_Apply_Start 245
  ECC_Apply_Stop 247
  ECC_Capture_Prune 249
  ECC_Capture_Reinit 251
  ECC_Capture_Resume 252
  ECC_Capture_Start 253
  ECC_Capture_Stop 255
  ECC_Capture_Suspend 256
DataJoiner status 138
db: # of applications connected currently 18
db: # of applications executing 18
db: # of Apply lagging subscriptions 140
db: # of binds/precomps attempted 19
db: # of catalog cache heap full 19
db: # of catalog cache inserts 19
db: # of catalog cache lookups 20
db: # of catalog cache overflows 20
db: # of cleaners due to dirty threshold 60
db: # of commit statements attempted 21
db: # of connects since 1st db connect 21
db: # of database files closed 61
db: # of database table spaces 154
db: # of deadlocks detected 22
db: # of direct read requests 61
db: # of direct reads from database 62
db: # of direct write requests 62
db: # of direct writes to database 62
db: # of event monitors 154
db: # of internal automatic rebinds 22
db: # of internal commits 22
db: # of internal rollbacks 23
db: # of internal rows deleted 23
db: # of internal rows inserted 23
db: # of internal rows updated 24
db: # of invalid triggers 155
db: # of lock timeouts 24
db: # of log pages read 24
db: # of log pages written 25
db: # of log space cleaners invoked 63
db: # of package cache inserts 25
db: # of package cache lookups 25
db: # of page cleans in an interval 63
db: # of pages for each prefetch request 64
db: # of pool reads 64
db: # of pool writes 64
db: # of prefetch requests in an interval 65
db: # of rollback statements attempted 26

db: # of rows deleted 26
db: # of rows inserted 26
db: # of rows selected 26
db: # of rows updated 27
db: # of secondary logs allocated currently 27
db: # of synchronous I/O 65
db: # of synchronous index reads (4 KB pages) 65
db: # of synchronous index writes (4 KB pages) 66
db: # of synchronous reads 66
db: # of synchronous writes 66
db: # of system table spaces 155
db: # of table spaces 156
db: # of table spaces with LONG data 156
db: # of tables 156
db: # of triggers 156
db: # of user indexes 157
db: # of victim page cleaners invoked 67
db: # of views 157
db: Active sorts 171
db: app_ctl_heap_sz 116
db: applheapsz 117
db: Applications in lock wait 148
db: Apply failed subscriptions 140
db: Apply requires full refresh 140
db: Apply subscription status 142
db: Apply subscriptions lag time 141
db: Average # of async reads per pool read 67
db: Average # of async writes per pool write 68
db: Average # of pool writes per pool read 68
db: Average # of sectors read per direct read 69
db: Average # of sectors written per direct write 69
db: Average direct read time (ms) 69
db: Average direct write time (ms) 70
db: Average lock wait time (s) 148
db: Average locks held per application 148
db: Average pool I/O time (ms) 70
db: Average pool read time (ms) 70
db: Average pool write time (ms) 71
db: Average sort time (ms) 171
db: Average synchronous data read time (ms) 71
db: Average synchronous data write time (ms) 71
db: Average synchronous I/O (ms) 72
db: Avg # of pages for each cleaner 72
db: avg_appls 117
db: Avg lock escalations per connection 27
db: Buffer pool async data reads (4 KB pages) 72
db: Buffer pool async data writes 73
db: Buffer pool async index reads (4 KB pages) 73
db: Buffer pool async index writes 73
db: Buffer pool async read time (ms) 74
db: Buffer pool async write time (ms) 74
db: Buffer pool data logical reads 74
db: Buffer pool data pages copied from extended storage 75
db: Buffer pool data pages copied to extended storage 75
db: Buffer pool data physical reads 75
db: Buffer pool data writes 76
db: Buffer pool I/Os per second 78
db: Buffer pool index logical reads 76
db: Buffer pool index pages copied from extended storage (4 KB pages) 76
db: Buffer pool index pages copied to extended storage (4 KB pages) 77
db: Buffer pool index physical reads 77
db: Buffer pool index writes 77
db: Buffer pool time waited for prefetch (ms) 78
db: buffpage 117
db: Capture errors 142
db: Capture lag 143
db: Capture pruning 143
db: catalogcache_sz 118
db: chngpgs_thresh 118
db: Committed statements per second 190
db: dbheap 119
db: DDL SQL statements 190
db: Deadlocks found during last interval 149
db: Direct read time (ms) 78
db: Direct write time (ms) 79
db: Dynamic SQL statements 190
db: Extended storage read/write ratio 79
db: Failed SQL statements 190
db: Internal rollbk due to deadlock 28
db: Last backup timestamp 28
db: Lock escalations during last interval 149
db: Lock list in use (bytes) 150
db: Lock timeouts during last interval 150
db: Lock waits 150
db: Lock waits in an interval 151
db: locklist 119
db: Locks held 151
db: Log I/O 29
db: logbufsz 120
db: logprimary 121
db: Max # of concurrent connections 29
db: Max database heap allocated 29
db: Max primary log space used 30
db: Max secondary logs space used 30
db: Max total log space used 30
db: maxappls 121
db: Maximum agents associated with applications 8
db: Maximum coordinating agents 9
db: maxlocks 122
db: mincommit 123
db: newlogpath 123
db: num_iocleaners 124
db: num_ioservers 125
db: Number of all invalid packages 194
db: Number of invalid SYSTEM packages 195
db: Number of invalid user packages 196
db: pckcachesz 126
db: Percent application in lock wait 151
db: Percent buffer pool hit ratio, data+index 80
db: Percent buffer pool hit ratio, index 80
db: Percent catalog cache hit ratio 81
db: Percent connections used 31
db: Percent DDL SQL 191
db: Percent deadlock rollbacks 152
db: Percent failed SQL 191
db: Percent internal rollbacks due to internal deadlocks 31

db: Percent locklist space used by database 32
db: Percent package cache hit ratio 32
db: Percent select SQL 192
db: Percent sort overflowed 172
db: Percent space used in database heap 126
db: Percent UID SQL 192
db: Percent used in primary log 33
db: Percent used in secondary log 34
db: Primary log space allocated 35
db: restore_pending 127
db: Rollbacks per second 192
db: Select SQL statements 193
db: seqdetect 127
db: Sort heap allocated (4 KB pages) 172
db: Sort overflows 173
db: sortheap 128
db: SQL statements per second 193
db: Static SQL statements 193
db: Synchronous read time (ms) 82
db: Synchronous write time (ms) 82
db: Time waited on locks 152
db: Total direct I/O time (ms) 82
db: Total exclusive lock escalations 153
db: Total pool physical I/O (ms) 83
db: Total pool physical read (ms) 83
db: Total pool physical write (ms) 83
db: Total sort time (ms) 173
db: Total sorts 173
db: Total synchronous I/O time (ms) 84
db: Update/Insert/Delete SQL 193
db: User-defined SQL (numeric result) 157
db: User-defined SQL (string result) 158
DB2_Agent_Monitors 4
DB2 agents and applications monitoring collection
conn: # of times agents are stolen 5
conn: Application work load 6
conn: Maximum associated agents 7
conn: Total system CPU time used by agents 7
conn: Total user CPU time used by agents 8
db: Maximum agents associated with applications 8
db: Maximum coordinating agents 9
dbms: # of idle agents 9
dbms: # of times agents are stolen 9
dbms: Agent creation ratio 9
dbms: Agents registered 10
dbms: Agents waiting for a token 10
dbms: Maximum agents registered 11
dbms: Maximum agents waiting 11
dbms: Percent agents waiting 11
dbms: Percent total connections executing 12
description 4
DB2 basic monitoring collection
conn: Percent locklist space used by application 17
db: # of applications connected currently 18
db: # of applications executing 18
db: # of binds/precomps attempted 19
db: # of catalog cache heap full 19
db: # of catalog cache inserts 19
db: # of catalog cache lookups 20
db: # of catalog cache overflows 20
db: # of commit statements attempted 21
DB2 basic monitoring collection *(continued)*
db: # of connects since 1st db connect 21
db: # of deadlocks detected 22
db: # of internal automatic rebinds 22
db: # of internal commits 22
db: # of internal rollbacks 23
db: # of internal rows deleted 23
db: # of internal rows inserted 23
db: # of internal rows updated 24
db: # of lock timeouts 24
db: # of log pages read 24
db: # of log pages written 25
db: # of package cache inserts 25
db: # of package cache lookups 25
db: # of rollback statements attempted 26
db: # of rows deleted 26
db: # of rows inserted 26
db: # of rows selected 26
db: # of rows updated 27
db: # of secondary logs allocated currently 27
db: Avg lock escalations per connection 27
db: Internal rollbk due to deadlock 28
db: Last backup timestamp 28
db: Log I/O 29
db: Max # of concurrent connections 29
db: Max database heap allocated 29
db: Max primary log space used 30
db: Max secondary logs space used 30
db: Max total log space used 30
db: Percent connections used 31
db: Percent internal rollbacks due to internal deadlocks 31
db: Percent locklist space used by database 32
db: Percent package cache hit ratio 32
db: Percent used in primary log 33
db: Percent used in secondary log 34
db: Primary log space allocated 35
DB2 ECC monitoring communication agent status 15
DB2 instance status 16
DB2 SNMP agent status 16
dbms: # of idle agents 35
dbms: # of local connections 35
dbms: # of local connections executing 36
dbms: # of local databases with current connections 36
dbms: # of piped sorts accepted 37
dbms: # of piped sorts requested 37
dbms: # of remote connections 37
dbms: # of remote connections executing 38
dbms: Committed private memory 38
dbms: Last reset timestamp 38
dbms: Percent piped sort hit ratio 39
dbms: Start database manager timestamp 40
dbpg: Data redistribution status 40
dbpg: Data redistribution status for a nodegroup 41
dbpg: DB2 local nodes status 41
dbpg: FCM daemons status 42
dbpg: Partitioned database data redistribution status 42

DB2 basic monitoring collection *(continued)*
dbpg: Percentage differential in row distribution of a table 42
description 13
tbsp: Percent space used in DMS table space 43
tbsp: Space used in SMS table space 44
tbsp: Table space status 45
User-defined OS command (numeric result) 17
User-defined OS command (string result) 17
DB2_Basic_Monitors 13
DB2 buffer pool I/O monitoring collection
conn: # of direct reads (4 KB pages) 52
conn: # of direct writes (4 KB pages) 52
conn: Average # of sectors read per direct read 53
conn: Average # of sectors written per direct write 54
conn: Buffer pool time waited for prefetch (ms) 54
conn: Data page writes (4 KB pages) 54
conn: Index page reads (4 KB pages) 55
conn: Index page writes (4 KB pages) 55
conn: Logical data reads (4 KB pages) 56
conn: Percent buffer pool hit ratio, data+index 57
conn: Percent buffer pool hit ratio, index 57
conn: Percent package cache hit ratio 58
conn: Pool read time (s) 59
conn: Pool write time (s) 59
conn: Total pool I/O time (s) 60
db: # of cleaners due to dirty threshold 60
db: # of database files closed 61
db: # of direct read requests 61
db: # of direct reads from database 62
db: # of direct write requests 62
db: # of direct writes to database 62
db: # of log space cleaners invoked 63
db: # of page cleans in an interval 63
db: # of pages for each prefetch request 64
db: # of pool reads 64
db: # of pool writes 64
db: # of prefetch requests in an interval 65
db: # of synchronous I/O 65
db: # of synchronous index reads (4 KB pages) 65
db: # of synchronous index writes (4 KB pages) 66
db: # of synchronous reads 66
db: # of synchronous writes 66
db: # of victim page cleaners invoked 67
db: Average # of async reads per pool read 67
db: Average # of async writes per pool write 68
db: Average # of pool writes per pool read 68
db: Average # of sectors read per direct read 69
db: Average # of sectors written per direct write 69
db: Average direct read time (ms) 69
db: Average direct write time (ms) 70
db: Average pool I/O time (ms) 70
db: Average pool read time (ms) 70
db: Average pool write time (ms) 71
db: Average synchronous data read time (ms) 71
db: Average synchronous data write time (ms) 71
db: Average synchronous I/O (ms) 72
db: Avg # of pages for each cleaner 72
db: Buffer pool async data reads (4 KB pages) 72
db: Buffer pool async data writes 73
DB2 buffer pool I/O monitoring collection *(continued)*
db: Buffer pool async index reads (4 KB pages) 73
db: Buffer pool async index writes 73
db: Buffer pool async read time (ms) 74
db: Buffer pool async write time (ms) 74
db: Buffer pool data logical reads 74
db: Buffer pool data pages copied from extended storage 75
db: Buffer pool data pages copied to extended storage 75
db: Buffer pool data physical reads 75
db: Buffer pool data writes 76
db: Buffer pool I/Os per second 78
db: Buffer pool index logical reads 76
db: Buffer pool index pages copied from extended storage (4 KB pages) 76
db: Buffer pool index pages copied to extended storage (4 KB pages) 77
db: Buffer pool index physical reads 77
db: Buffer pool index writes 77
db: Buffer pool time waited for prefetch (ms) 78
db: Direct read time (ms) 78
db: Direct write time (ms) 79
db: Extended storage read/write ratio 79
db: Percent buffer pool hit ratio, data+index 80
db: Percent buffer pool hit ratio, index 80
db: Percent catalog cache hit ratio 81
db: Synchronous read time (ms) 82
db: Synchronous write time (ms) 82
db: Total direct I/O time (ms) 82
db: Total pool physical I/O (ms) 83
db: Total pool physical read (ms) 83
db: Total pool physical write (ms) 83
db: Total synchronous I/O time (ms) 84
dbp: # of FCM nodes 84
dbp: Connection Status 84
dbp: FCM buffers currently free 85
dbp: FCM connection entries currently free 85
dbp: FCM request blocks currently free 85
dbp: Minimum FCM buffers free 86
dbp: Minimum FCM connection entries free 86
dbp: Minimum FCM message anchors free 86
dbp: Minimum FCM request blocks free 87
dbp: Percent FCM buffers currently used 87
dbp: Percent FCM connection entries currently used 88
dbp: Percent FCM request blocks currently used 88
dbp: Percent maximum FCM buffers used 89
dbp: Percent maximum FCM connection entries used 89
dbp: Percent maximum FCM message anchors used 90
dbp: Percent maximum FCM request blocks used 91
dbp: Total buffers received 91
dbp: Total buffers sent 92
description 47
tbsp: # of direct read requests 92
tbsp: # of direct reads from database 93
tbsp: # of direct write requests 93
tbsp: # of direct writes to database 94

DB2 buffer pool I/O monitoring collection *(continued)*
tbsp: # of pool reads 94
tbsp: # of pool writes 94
tbsp: # of synchronous index reads (4 KB pages) 95
tbsp: # of synchronous index writes (4 KB pages) 95
tbsp: # of synchronous reads 96
tbsp: # of synchronous writes 96
tbsp: Average # of sectors read per direct read 97
tbsp: Average # of sectors written per direct write 97
tbsp: Average direct read time (ms) 97
tbsp: Average direct write time (ms) 98
tbsp: Average pool I/O time (ms) 98
tbsp: Average pool read time (ms) 98
tbsp: Average pool write time (ms) 99
tbsp: Average synchronous data read time (ms) 99
tbsp: Average synchronous data write time (ms) 100
tbsp: Average synchronous I/O time (ms) 100
tbsp: Buffer pool async data reads 100
tbsp: Buffer pool async data writes 101
tbsp: Buffer pool async index writes 102
tbsp: Buffer pool async read time (ms) 102
tbsp: Buffer pool async write time (ms) 103
tbsp: Buffer pool asynchronous index reads 101
tbsp: Buffer pool data logical reads 103
tbsp: Buffer pool data pages copied from extended storage 104
tbsp: Buffer pool data pages copied to extended storage 104
tbsp: Buffer pool data physical reads 104
tbsp: Buffer pool data writes 105
tbsp: Buffer pool I/Os per second 105
tbsp: Buffer pool index logical reads 106
tbsp: Buffer pool index pages copied from extended storage (4 KB pages) 106
tbsp: Buffer pool index pages copied to extended storage< 106
tbsp: Buffer pool index physical reads 107
tbsp: Buffer pool index writes 107
tbsp: Direct read time 108
tbsp: Direct write time (ms) 108
tbsp: Extended storage read/write ratio 108
tbsp: File closed 109
tbsp: Percent buffer pool hit ratio, data+index 109
tbsp: Percent buffer pool hit ratio, index 110
tbsp: Percent prefetch satisfied 110
tbsp: Percent total I/O 111
tbsp: Prefetch requests in an interval 111
tbsp: Synchronous read time (ms) 112
tbsp: Synchronous write time (ms) 112
tbsp: Total direct I/O time (ms) 112
tbsp: Total pool physical I/O time (ms) 113
tbsp: Total pool physical read time (ms) 113
tbsp: Total pool physical write time (ms) 113
tbsp: Total synchronous I/O 114
tbsp: Total synchronous I/O time (ms) 114
DB2_Bufio_Monitors 47
DB2_Config_Monitors 115
DB2 configuration monitoring collection
db: app_ctl_heap_sz 116
db: applheapsz 117
db: avg_appls 117
db: buffpage 117
db: catalogcache_sz 118
db: chngpgs_thresh 118
db: dbheap 119
db: locklist 119
db: logbufsz 120
db: logprimary 121
db: maxappls 121
db: maxlocks 122
db: mincommit 123
db: newlogpath 123
db: num_iocleaners 124
db: num_ioservers 125
db: pckcachesz 126
db: Percent space used in database heap 126
db: restore_pending 127
db: seqdetect 127
db: sortheap 128
dbms: agentpri 128
dbms: aslheapsz 129
dbms: fcm_num_anchors 129
dbms: fcm_num_buffers 129
dbms: fcm_num_connect 130
dbms: fcm_num_rqb 130
dbms: max_agents 131
dbms: max_coordagents 131
dbms: maxcagents 132
dbms: Maximum idle agents in the agent pool 133
dbms: Minimum committed private memory 133
dbms: mon_heap_sz 133
dbms: num_poolagents 134
dbms: Percent private memory used 134
dbms: query_heap_sz 135
dbms: rqrioblk 135
dbms: sheapthres 136
dbms: sqlstmtsz 137
description 115
DB2 data replication relational monitoring collection
Data replication Apply status 139
DataJoiner status 138
db: # of Apply lagging subscriptions 140
db: Apply failed subscriptions 140
db: Apply requires full refresh 140
db: Apply subscription status 142
db: Apply subscriptions lag time 141
db: Capture errors 142
db: Capture lag 143
db: Capture pruning 143
description 137
SNA link status 139
TCP/IP communication link status 139
DB2 database state 161
DB2 database task endpoints 208
DB2 database tasks
ECC_Activate_Database 210
ECC_Alter_Bufferpool 211
ECC_Backup_Database 212

DB2 database tasks *(continued)*
ECC_Create_Admin_Server 214
ECC_Create_Explain_Tables 214
ECC_Create_Sample_Database 215
ECC_Deactivate_Database 216
ECC_Drop_Admin_Server 217
ECC_Drop_Sample_Database 217
ECC_Force_All_Applications 217
ECC_Force_Applications 218
ECC_Get_Admin_Configuration 219
ECC_Get_Admin_Server_Instance 219
ECC_Get_Database_Configuration 220
ECC_Get_Database_Manager_Configuration 221
ECC_Invoke_Stored_Procedure 221
ECC_List_Applications 222
ECC_List_Backup_Recovery_File 223
ECC_List_DCS_Applications 224
ECC_List_Node_Directory 225
ECC_Prune_Recovery_History_File 226
ECC_Quiesce_Tablespaces 227
ECC_Rebind_All_Packages 228
ECC_Rebind_Package 229
ECC_Reorganize_Table 230
ECC_Reorgchk 231
ECC_Reset_Admin_Configuration 232
ECC_Reset_Database_Configuration 232
ECC_Reset_Database_Manager_Configuration 233
ECC_Restart_Database 234
ECC_Run_Statistics 235
ECC_Set_Admin_Server_ID 236
ECC_Start_Admin_Server 237
ECC_Start_DB2 237
ECC_Start_DB2_NT_Security 239
ECC_Stop_Admin_Server 239
ECC_Stop_DB2 240
ECC_Stop_DB2_NT_Security 241
ECC_Update_Admin_Configuration 241
ECC_Update_Database_Configuration 242
ECC_Update_Database_Manager_Configuration 243
DB2_DataReplication_Monitors 137
DB2_ECC-AdminTasks 203
DB2_ECC-DatabaseTasks 208
DB2 ECC monitoring communication agent status 15
DB2_ECC-ReplicationTasks 244
DB2 ECC task libraries 201
DB2 instance status 16
DB2_Lock_Monitors 143
DB2 locks and deadlocks monitoring collection
conn: Deadlocks found during last interval 144
conn: Lock escalations during last interval 145
conn: Lock wait time (s) 146
conn: Lock waits 146
conn: Locks held 147
conn: UOW lock wait time (s) 147
db: Applications in lock wait 148
db: Average lock wait time (s) 148
db: Average locks held per application 148
db: Deadlocks found during last interval 149
db: Lock escalations during last interval 149
db: Lock list in use (bytes) 150
DB2 locks and deadlocks monitoring collection *(continued)*
db: Lock timeouts during last interval 150
db: Lock waits 150
db: Lock waits in an interval 151
db: Locks held 151
db: Percent application in lock wait 151
db: Percent deadlock rollbacks 152
db: Time waited on locks 152
db: Total exclusive lock escalations 153
description 143
DB2 query monitoring collection
db: # of database table spaces 154
db: # of event monitors 154
db: # of invalid triggers 155
db: # of system table spaces 155
db: # of table spaces 156
db: # of table spaces with LONG data 156
db: # of tables 156
db: # of triggers 156
db: # of user indexes 157
db: # of views 157
db: User-defined SQL (numeric result) 157
db: User-defined SQL (string result) 158
description 153
table: # of rows 158
tbsp: # of tables 158
user: # of tables 159
DB2_Query_Monitors 153
DB2 server accumulated connections 161
DB2 server current connections 162
DB2 server finished transactions 162
DB2 server status 163
DB2 server version 163
DB2 SNMP agent status 16
DB2 SNMP monitoring collection
DB2 database state 161
DB2 server accumulated connections 161
DB2 server current connections 162
DB2 server finished transactions 162
DB2 server status 163
DB2 server version 163
description 159
Maximum number of agents allowed to register at the same time 164
Maximum number of agents registered at the same time 164
Total logical bytes read by DB2 server for all active databases 165
Total logical pages read by DB2 server for all active databases 165
Total physical bytes read by DB2 server 166
Total physical bytes written by DB2 server for all active databases 166
Total physical pages written by DB2 server for all active databases 166
Total rows selected returned by DB2 server for all active databases 167
Total SQL statements handled by DB2 server for all active databases 167

DB2 SNMP monitoring collection *(continued)*
Total SQL statements received by DB2 server for all active databases 168
DB2_SNMP_Monitors 159
DB2_Sort_Monitors 168
DB2 sort work monitoring collection
conn: Application sorts 169
conn: Sort time (s) 170
conn: Statement sorts 170
db: Active sorts 171
db: Average sort time (ms) 171
db: Percent sort overflowed 172
db: Sort heap allocated (4 KB pages) 172
db: Sort overflows 173
db: Total sort time (ms) 173
db: Total sorts 173
dbms: Percent piped sorts rejected 174
dbms: Percent sort heap allocated 174
dbms: Post threshold sorts 175
dbms: Sort heap allocated (4 KB pages) 175
dbms: Total piped sorts rejected 175
description 168
DB2 SQL statement activity monitoring collection
conn: Application section inserts 177
conn: Application section lookups 178
conn: Binds/precompiles attempted 179
conn: Commits 179
conn: DDL SQL statements 180
conn: Dynamic SQL statements 180
conn: Explicit commits 181
conn: Explicit rollbacks 182
conn: Failed SQL statements 182
conn: Open block cursors 183
conn: Open cursors 183
conn: Percent DDL SQL 184
conn: Percent UID SQL 184
conn: Rollbacks 185
conn: Rows deleted 186
conn: Rows inserted 186
conn: Rows selected 187
conn: Rows updated 187
conn: Select SQL statements 188
conn: SQL requests since last commit 188
conn: Static SQL statements 189
conn: Update/Insert/Delete SQL 189
db: Committed statements per second 190
db: DDL SQL statements 190
db: Dynamic SQL statements 190
db: Failed SQL statements 190
db: Percent DDL SQL 191
db: Percent failed SQL 191
db: Percent select SQL 192
db: Percent UID SQL 192
db: Rollbacks per second 192
db: Select SQL statements 193
db: SQL statements per second 193
db: Static SQL statements 193
db: Update/Insert/Delete SQL 193
description 176
DB2_Stmt_Monitors 176
DB2 table monitoring collection
db: Number of all invalid packages 194
db: Number of invalid SYSTEM packages 195
db: Number of invalid user packages 196
description 194
table: Overflow accesses 196
table: Reorg needed 197
table: Rows read per second 199
table: Rows written per second 199
DB2_Table_Monitors 194
dbms: # of idle agents 9, 35
dbms: # of local connections 35
dbms: # of local connections executing 36
dbms: # of local databases with current connections 36
dbms: # of piped sorts accepted 37
dbms: # of piped sorts requested 37
dbms: # of remote connections 37
dbms: # of remote connections executing 38
dbms: # of times agents are stolen 9
dbms: Agent creation ratio 9
dbms: agentpri 128
dbms: Agents registered 10
dbms: Agents waiting for a token 10
dbms: aslheapsz 129
dbms: Committed private memory 38
dbms: fcm_num_anchors 129
dbms: fcm_num_buffers 129
dbms: fcm_num_connect 130
dbms: fcm_num_rqb 130
dbms: Last reset timestamp 38
dbms: max_agents 131
dbms: max_coordagents 131
dbms: maxcagents 132
dbms: Maximum agents registered 11
dbms: Maximum agents waiting 11
dbms: Maximum idle agents in the agent pool 133
dbms: Minimum committed private memory 133
dbms: mon_heap_sz 133
dbms: num_poolagents 134
dbms: Percent agents waiting 11
dbms: Percent piped sort hit ratio 39
dbms: Percent piped sorts rejected 174
dbms: Percent private memory used 134
dbms: Percent sort heap allocated 174
dbms: Percent total connections executing 12
dbms: Post threshold sorts 175
dbms: query_heap_sz 135
dbms: rqrioblk 135
dbms: sheapthres 136
dbms: Sort heap allocated (4 KB pages) 175
dbms: sqlstmtsz 137
dbms: Start database manager timestamp 40
dbms: Total piped sorts rejected 175
dbp: # of FCM nodes 84
dbp: Connection Status 84
dbp: FCM buffers currently free 85
dbp: FCM connection entries currently free 85
dbp: FCM request blocks currently free 85
dbp: Minimum FCM buffers free 86
dbp: Minimum FCM connection entries free 86

dbp: Minimum FCM message anchors free 86
dbp: Minimum FCM request blocks free 87
dbp: Percent FCM buffers currently used 87
dbp: Percent FCM connection entries currently used 88
dbp: Percent FCM request blocks currently used 88
dbp: Percent maximum FCM buffers used 89
dbp: Percent maximum FCM connection entries used 89
dbp: Percent maximum FCM message anchors used 90
dbp: Percent maximum FCM request blocks used 91
dbp: Total buffers received 91
dbp: Total buffers sent 92
dbpg: Data redistribution status 40
dbpg: Data redistribution status for a nodegroup 41
dbpg: DB2 local nodes status 41
dbpg: FCM daemons status 42
dbpg: Partitioned database data redistribution status 42
dbpg: Percentage differential in row distribution of a table 42
default values, adding monitoring sources 3

E

ECC_Activate_Database 210
ECC_Alter_Bufferpool 211
ECC_Apply_Start 245
ECC_Apply_Stop 247
ECC_Backup_Database 212
ECC_Broadcast_Message 203
ECC_Capture_Prune 249
ECC_Capture_Reinit 251
ECC_Capture_Resume 252
ECC_Capture_Start 253
ECC_Capture_Stop 255
ECC_Capture_Suspend 256
ECC_Create_Admin_Server 214
ECC_Create_Explain_Tables 214
ECC_Create_Sample_Database 215
ECC_Deactivate_Database 216
ECC_Drop_Admin_Server 217
ECC_Drop_Sample_Database 217
ECC_Force_All_Applications 217
ECC_Force_Applications 218
ECC_Get_Admin_Configuration 219
ECC_Get_Admin_Server_Instance 219
ECC_Get_Database_Configuration 220
ECC_Get_Database_Manager_Configuration 221
ECC_Invoke_Stored_Procedure 221
ECC_List_Applications 222
ECC_List_Backup_Recovery_File 223
ECC_List_DCS_Applications 224
ECC_List_Node_Directory 225
ECC_Prune_Recovery_History_File 226
ECC_Quiesce_Tablespaces 227
ECC_Rebind_All_Packages 228
ECC_Rebind_Package 229
ECC_Reorganize_Table 230
ECC_Reorgchk 231
ECC_Reset_Admin_Configuration 232
ECC_Reset_Counters 204
ECC_Reset_Database_Configuration 232
ECC_Reset_Database_Manager_Configuration 233
ECC_Restart_Database 234
ECC_Run_Statistics 235
ECC_Send_Notice 204
ECC_Set_Admin_Server_ID 236
ECC_Set_Up_Monitoring_Agent 205
ECC_Start_Admin_Server 237
ECC_Start_DB2 237
ECC_Start_DB2_NT_Security 239
ECC_Start_DB2SNMP_Agent 206
ECC_Start_Monitoring_Agent 207
ECC_Stop_Admin_Server 239
ECC_Stop_DB2 240
ECC_Stop_DB2_NT_Security 241
ECC_Stop_DB2SNMP_Agent 207
ECC_Stop_Monitoring_Agent 207
ECC_Update_Admin_Configuration 241
ECC_Update_Database_Configuration 242
ECC_Update_Database_Manager_Configuration 243
endpoints, prefixes for 1
endpoints for
  data replication tasks 244
  DB2 database tasks 208
  monitor and administration tasks 203

F

formats, application IDs 259

M

Maximum number of agents allowed to register at the same time 164
Maximum number of agents registered at the same time 164
monitor and administration task endpoints 203
monitor and administration tasks
  ECC_Broadcast_Message 203
  ECC_Reset_Counters 204
  ECC_Send_Notice 204
  ECC_Set_Up_Monitoring_Agent 205
  ECC_Start_DB2SNMP_Agent 206
  ECC_Start_Monitoring_Agent 207
  ECC_Stop_DB2SNMP_Agent 207
  ECC_Stop_Monitoring_Agent 207
monitoring collections
  DB2_Agent_Monitors 4
  DB2_Basic_Monitors 13
  DB2_Bufio_Monitors 47
  DB2_Config_Monitors 115
  DB2_DataReplication_Monitors 137
  DB2_Lock_Monitors 143
  DB2_Query_Monitors 153
  DB2_SNMP_Monitors 159
  DB2_Sort_Monitors 168
  DB2_Stmt_Monitors 176
  DB2_Table_Monitors 194
  description 1
  list of 1

monitoring sources
adding
with defaults values 3
with the CLI 3
with the GUI 2
prefixes and endpoints 1
that return a numeric value, using 4
using 2

P

prefixes, description 1

S

SNA link status 139
snapshot application IDs 259
stolen agents, defined 6

T

table: # of rows 158
table: Overflow accesses 196
table: Reorg needed 197
table: Rows read per second 199
table: Rows written per second 199
task libraries
data replication tasks 244
DB2 database 208
monitor and administration 203
task libraries, DB2 ECC 201
tasks
data replication, list of 244
DB2 database, list of 208
monitoring and administration, list of 203
tbsp: # of direct read requests 92
tbsp: # of direct reads from database 93
tbsp: # of direct write requests 93
tbsp: # of direct writes to database 94
tbsp: # of pool reads 94
tbsp: # of pool writes 94
tbsp: # of synchronous index reads (4 KB pages) 95
tbsp: # of synchronous index writes (4 KB pages) 95
tbsp: # of synchronous reads 96
tbsp: # of synchronous writes 96
tbsp: # of tables 158
tbsp: Average # of sectors read per direct read 97
tbsp: Average # of sectors written per direct write 97
tbsp: Average direct read time (ms) 97
tbsp: Average direct write time (ms) 98
tbsp: Average pool I/O time (ms) 98
tbsp: Average pool read time (ms) 98
tbsp: Average pool write time (ms) 99
tbsp: Average synchronous data read time (ms) 99
tbsp: Average synchronous data write time (ms) 100
tbsp: Average synchronous I/O time (ms) 100
tbsp: Buffer pool async data reads 100
tbsp: Buffer pool async data writes 101
tbsp: Buffer pool async index writes 102
tbsp: Buffer pool async read time (ms) 102
tbsp: Buffer pool async write time (ms) 103
tbsp: Buffer pool asynchronous index reads 101
tbsp: Buffer pool data logical reads 103
tbsp: Buffer pool data pages copied from extended storage 104
tbsp: Buffer pool data pages copied to extended storage 104
tbsp: Buffer pool data physical reads 104
tbsp: Buffer pool data writes 105
tbsp: Buffer pool I/Os per second 105
tbsp: Buffer pool index logical reads 106
tbsp: Buffer pool index pages copied from extended storage (4 KB pages) 106
tbsp: Buffer pool index pages copied to extended storage< 106
tbsp: Buffer pool index physical reads 107
tbsp: Buffer pool index writes 107
tbsp: Direct read time 108
tbsp: Direct write time (ms) 108
tbsp: Extended storage read/write ratio 108
tbsp: File closed 109
tbsp: Percent buffer pool hit ratio, data+index 109
tbsp: Percent buffer pool hit ratio, index 110
tbsp: Percent prefetch satisfied 110
tbsp: Percent space used in DMS table space 43
tbsp: Percent total I/O 111
tbsp: Prefetch requests in an interval 111
tbsp: Space used in SMS table space 44
tbsp: Synchronous read time (ms) 112
tbsp: Synchronous write time (ms) 112
tbsp: Table space status 45
tbsp: Total direct I/O time (ms) 112
tbsp: Total pool physical I/O time (ms) 113
tbsp: Total pool physical read time (ms) 113
tbsp: Total pool physical write time (ms) 113
tbsp: Total synchronous I/O 114
tbsp: Total synchronous I/O time (ms) 114
TCP/IP communication link status 139
Total logical bytes read by DB2 server for all active databases 165
Total logical pages read by DB2 server for all active databases 165
Total physical bytes read by DB2 server 166
Total physical bytes written by DB2 server for all active databases 166
Total physical pages written by DB2 server for all active databases 166
Total rows selected returned by DB2 server for all active databases 167
Total SQL statements handled by DB2 server for all active databases 167
Total SQL statements received by DB2 server for all active databases 168

U

user: # of tables 159
User-defined OS command (numeric result) 17
User-defined OS command (string result) 17
using monitoring sources that return a numeric value 4

V

victim page, defined 67

Readers' Comments — We'd Like to Hear from You

DB2 Enterprise Control Center for TME 10 V1.2
Monitoring Collection and Task Library Reference

SC26-9085-01

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You
SC26-9085-01

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department W92/H3
PO Box 49023
San Jose, CA
95161-9945

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC26-9085-01